Made Kamisutara - Gatut Purwantoro

BUKU AJAR

E-COMMERCE
PEMOGRAMAN *WEB*

Diterbitkan oleh:
NAROTAMA *UNIVERSITY PRESS*

SANKSI PELANGGARAN PASAL 113

Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta:

(1) Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp 100.000.000 (seratus juta rupiah).

(2) Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp 500.000.000,00 (lima ratus juta rupiah).

(3) Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp 1.000.000.000,00 (satu miliar rupiah).

(4) Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp 4.000.000.000,00 (empat miliar rupiah).

Made Kamisutara - Gatut Purwantoro

BUKU AJAR

E-COMMERCE

PEMOGRAMAN *WEB*

Diterbitkan oleh:

Narotama *University Press*

NAROTAMA University Press

e-Commerce Pemrograman Web/disusun oleh Made Kamisutara, dkk
274 hal [xviii] ; editor, Seger S.S

Ukuran Buku : A4 (21 x 29,7 cm)
Copyright © 20176 oleh Made Kamisutara, dkk

Hak Cipta Dilindungi Undang-Undang
© All Rights Reserved

"Dilarang mengutip dan memperbanyak sebagian
atau seluruhnya dalam bentuk apapun tanpa ijin
tertulis dari Penerbit"

© Cetakan Pertama: Nopember 2016

Penyusun	: Made Kamisutara, ST., M.Kom Gatut Purwantoro, S.Kom., MM
Editor	: Seger, S.S
Layout/Setting	: TIM Kreatif Narpress
Design Cover	: Narpress Design

ISBN : **978-602-6557-06-3**

© HAK CIPTA DILINDUNGI UNDANG-UNDANG

Diterbitkan oleh **Narotama *University Press***
Jl. Arief Rachman Hakim No.51 Surabaya 60117
Telp: 031-5946404, 5995578 Fax: 031-5931213
Website: www.narotama.ac.id
Email: narotamapress@narotama.ac.id

KATA PENGANTAR

Khalayak pengguna buku ini adalah ditujukan untuk pembaca dari kalangan mahasiswa, dosen, dan masyarakat luas yang berkecinpung di bidang IT maupun industri bisnis, termasuk di dalamnya sektor pelaku usaha kecil atau UKM. Struktur penyajian buku ini mencakup penjelasan *E–Commerce* dan Pemrograman *Web* serta langkah-langkah pembuatan aplikasi *E–Commerce* secara sederhana, yang nantinya juga dapat digunakan oleh pelaku usaha kecil/UKM. Pesan khusus yang ingin penulis sampaikan adalah dengan mempelajari buku ini, maka kita bisa menciptakan aplikasi *E-Commerce* atau program *web* dan membantu masyarakat yang mempunyai kegiatan usaha, terutama bisnis di bidang UKM.

Penulis juga menyadari bahwa taka da gading yang tak retak, begitu pula dengan kehadiran buku ini yang masih penulis rasakan perlu adanya *update* informasi-informasi terbaru dalam memperbaiki materi. Kritik dan saran yang bersifat membangun sangat kami nantikan sehingga akan semakin membuka wawasan keilmuan dan implementasi pembelajaran *E-Commerce* dan Pemrograman *Web* yang ideal di seluruh penjuru Nusantara. Semoga. Untuk segala saran dan masukan perbaikan, silakan kirim email ke: kamisutara@gmail.com, akhir kata semoga buku ini bermanfaat bagi semua.

Surabaya, Pebruari 2017

Penulis,

UNIVERSITAS NAROTAMA
PRO PATRIA
SURABAYA

DAFTAR ISI

KATA PENGANTAR v

DAFTAR ISI vii

DAFTAR GAMBAR xvii

BAB 1 E–COMMERCE

Tujuan Instruksional Umum 2

1.1. Pengertian *E–Commerce* 3

1.2. Jenis *E-Commerce* 5

1.2.1. *Business to Business* (B2B) 5

1.2.2. *Business to Consumer* (B2C) 7

1.2.3. *Consumen to Consumen* (C2C) 8

1.2.4. *Consumen to Business*(C2B) 9

1.2.5. Perdagangan Intrabisnis (Intraorganisasional) 9

1.2.6. Perdagangan Kolabratif *(Collaborativ E-Commerce)* 9

1.2.7. Perdagangan *Mobile* (*Mobile Commerce/M-Commerce*) 9

1.2.8. Intrabusiness *(Organizational) E-Commerce* 9

1.2.9. Pemerintah ke Masyarakat (*Goverment to Citizen*-G2C) 10

1.3. Standar Teknologi *E-Commerce* 11

1.4. Manfaat dan Resiko *E–Commerce* 12

1.5. Klasifikasi Transaksi *E–Commerce* 14

Ray Tomlinson Penemu E-mail (Elektronik Mail) 15

Rangkuman 16

Daftar Istilah 16

Latihan Soal 16

Kasus Dalam Jaringan UKM 17

Bab 2 PEMROGRAMAN WEB

Tujuan Instruksional Umum 20

2.1. Pengertian *Webmaster* 21

2.2. Pengertian dan Spesifikasi *Webmaster* 21

2.2.1. *Web Designer* 22

2.2.2. *Web Programmer* 23

2.2.3. *Web* Master 23

2.2.4. *Web Animator* 24

2.2.5. *Web Administr*ator 25
2.2.6. *Web Content Creator* 24
2.3. Tahapan Pembuatan Situs 24
2.4. Teknologi *Webmaster* 26
2.4.1. URL *(Uniform Resource Locator)* 26
2.4.2. *TCP/IP* 26
2.4.3. *IP Address* 26
2.4.4. DNS *(Domain Name System)* 26
2.4.5. Browser 27
2.5. Perencanaan dan Perancangan *Web* 28
2.6. Tujuan Pembuatan Situs *Web* 28
2.6.1. Alat Pemasaran 29
2.6.2. Nilai Tambah 29
2.6.3 Katalog 29
Ada Lovelace, Programer Wanita Pertama (1815–1852) 30
Rangkuman 31
Daftar Istilah 31
Latihan Soal 31
Permasalahan UKM Terkait dengan Webmaster 32

BAB 3 *HIPERTEXT MARKUP LANGUAGE* (HTML) DAN *CASCADING STYLE SHEET* (CSS)

Tujuan Instruksional Umum 34
3.1. *Hipertext Markup Language* (HTML) 35
3.1.1. Pengertian *Hipertext Markup Language* (HTML) 35
3.1.2. Skema Dasar HTML 36
3.1.3. Membuat Teks 37
3.1.3.1. Membuat Font 38
3.1.3.2. List 38
3.1.3.3. Menampilkan Teks yang Sudah Diformat 39
3.1.4. Membuat *Links* 40
3.1.4.1. Menyisipkan Sejumlah *Link* Pada Satu Gambar 40
3.1.5. Membuat Tabel 41

3.1.6. Membuat *Frame* 43
3.1.6.1. *Inline Frame* 44
3.1.7. Membuat *Form* 44
3.1.8. Membuat *Text Box* 44
3.1.9. Membuat *Password* 45
3.1.10. Membuat *Text Area* 45
3.1.11. Membuat Radio 46
3.1.12. Membuat *Checkbox* 46
3.1.13. Membuat *Select* 47
3.1.14. Membuat *Button, Reset, Submit,* dan *Image* 47
3.1.15. Membuat Hidden 48
3.1.16. Membuat File 48
3.1.17. Informasi Dokumen 48
3.1.18. Pengelompokan Elemen Dokumen 49
3.2. *Cascading Style Sheet* (CSS) 49
3.2.1. *Pengertian Cascading Style Sheet* (CSS) 49
3.2.2. CSS dan HTML 50
3.2.2.1. Menggunakan *tag* <*link*> 50
3.2.2.2. Menggunakan *tag* <*style*> 50
3.2.2.3. Perintah *@import* 51
3.2.2.4. *Inline Style* 51
3.2.3. CSS *Rules* 51
3.2.4. *Selector* 52
3.2.4.1. *tag* HTML 52
3.2.4.2. Selektor *class* 52
3.2.4.3. Selektor *id* 52
3.2.4.4. *Contextual Selector* 52
3.2.4.5. *Pseudo-Class* 53
3.2.4.6. Pseudo-Element 53
3.2.5. *Grouping* 53

3.2.6. *Specificity* 53
3.2.7. *Inheritance* 54
3.2.8. *Font* 54
3.2.9. Color 55
3.2.10. *Background* 55
3.2.11. *Text* 56
3.2.12. *Element Box* 56
Tim Berners-Lee, Penemu E-mail (Elektronik Mail) 58
Rangkuman 60
Daftar Istilah 60
Latihan Soal 60
Kasus Pembobolan ATM Pada Kegiatan UKM 61

Bab 4 DASAR PHP

Tujuan Instruksional Umum 64
4.1. *Server-Side Scripting* 65
4.2. Menyisipkan kode PHP dalam HTML atau sebaliknya 66
4.3. Komentar dalam PHP 68
4.4. Pengenal *(Identifier)* 68
4.5. Tipe Data 69
4.6. Variabel 69
4.7. Operator 70
4.7.1. Operator Aritmetika 71
4.7.2. Operator *Assignment* (Pemberian Nilai) 71
4.7.3. Operator String 71
4.7.4. Operator *Increment/Decrement* 71
4.7.5. Operator Logika 71
4.7.6. Operator Perbandingan 72
4.7.7. Operator *Tertiary* 72
4.7.8. Operator Kontrol *Error* 72

4.7.9. *Statement* Kontrol 72
4.7.9.1. Pernyataan *if* 72
4.7.9.2. Pernyataan *Switch* 73
4.7.9.3. Pernyataan *While* 74
4.7.9.4. Pernyataan *Do While* 74
4.7.9.5. Pernyataan *For* 74
4.7.9.6. Pernyataan *Break* 74
4.7.9.7. Pernyataan *Continue* 75
4.7.10. Fungsi dalam PHP 77
4.7.10.1. Operasi *Array* 79
4.7.10.1.1. *Array* Multi Dimensi 80
4.7.11. Operasi *String* 81
4.7.12. Operasi *Date* 83
4.7.12.1. Fungsi *date()* 83
4.7.12.2. Fungsi *getdate()* 84
4.8. Mengikutkan Skrip dari File Lain 86
4.8.1. Pernyataan *Include* 86
4.8.2. Pernyataan *Require* 86
4.9. Menangani Berkas 86
4.9.1. Fungsi *fopen* 87
4.9.2. Fungsi *fclose* 87
4.9.3. Fungsi *fgets* 88
4.9.4. Fungsi *fputs* 88
4.9.5. Fungsi *feof* 88
4.10. Menangani Form 89
4.10.1. Metode *Get* dan *Post* 89
4.10.2. Verifikasi *form* 91
4.11. *Appendix* 96
Rasmus Lerdorf Penemu PHP *(Perl Hypertext Preprocessor)* 93
Rangkuman 95

Daftar Istilah 95
Latihan Soal 95

Bab 5 *MySQL*

Tujuan Instruksional Umum 98
5.1. Pengertian *SQL* 99
5.2. Menjalankan *SQL* Pertama Kali 99
5.3. Hak Akses 100
5.4. Pembuatan *Password* untuk *Root* 106
5.5. *Localhost* atau %? 109
5.6. Perintah-perintah Dasar *MySQL* 110
5.7. Menjalankan *MySQL* 110
5.8. Keluar dari *MySQL* 112
5.9. Memilih *Database* 112
5.10. Melihat Isi *Database* 113
5.11. Melihat Struktur Tabel 113
5.12. Membuat *Database* Baru 114
5.13. Menghapus *Database* 114
5.14. Membuat Tabel Baru 114
5.15. Melihat Struktur Tabel 116
5.16. Menghapus Tabel 116
5.17. Perintah-perintah Dasar *MySQL* 117
5.18. Mengisi Data Baru 117
5.19. Menampilkan Data 118
5.20. Mengisi Data Baru (Bentuk Lainnya) 119
5.21. Mengubah Struktur Tabel 120
5.22. Mengubah Data 122
5.23. Menampilkan Data dengan Kriteria Tertentu 124
5.24. Menampilkan Data Kolom Tertentu Saja 125
5.25. Menampilkan Data Kolom Tertentu dengan Urutan 126
5.26. Menampilkan Data Baris dan Kolom Tertentu 127
5.27. Menampilkan Data Baris dan Kolom Tertentu dengan Urutan 128

5.28. Operator Pembanding dan Operator Logika 128
5.29. Menampilkan Terurut 132
Dr Fujio Masuoka Penemu *Flashdisk* 132
Rangkuman 134
Daftar Istilah 134
Latihan Soal 134
Studi Kasus yang Berkaitan dengan UKM 135

Bab 6 APLIKASI PHP *DATABASE*

Tujuan Instruksional Umum 138
6.1. Menggunakan *MySQL* 139
6.2. Aplikasi *Database* menggunakan *MySQL* 142
6.2.1. Membuat *Database MySQL* dengan *PhpMyAdmin* 142
6.3. *View* 146
6.4. *Insert* 147
6.5. *Edit* 150
6.6. *Searching* 153
6.7. Menggunakan ODBC 156
6.7.1. Apa itu ODBC 156
6.7.2. Membuat DSN 156
6.8. Mengakases *Database* Melalui PHP 160
Reynold B. Johnson (Penemu Hardisk Komputer) 167
Rangkuman 169
Daftar Istilah 169
Latihan Soal 170
Kasus Pada UKM: Membangun Jaringan Usaha 170

Bab 7 APLIKASI PHP *DATABASE*

Tujuan Instruksional Umum 172
7.1. Melakukan *Upload* 173
7.2. Membatasi *Ukuran File* 176
7.3. Mengenal *Variabel Global* Berakhiran *_name, _size, _type* 176

7.4. Menangani Penyalinan Secara Permanen 178
Marissa Ann Mayer (CEO Yahoo) 179
Rangkuman 180
Daftar Istilah 181
Latihan Soal 181
Konsep Jaringan Usaha: Sebuah Studi Kasus 181

Bab 8 APLIKASI PHP *DATABASE*

Tujuan Instruksional Umum 184
8.1. Menangani *Cookies* dan *Session* 185
8.1.1. Penggunaan *Cookie* 185
8.1.2. Penanganan *Session* 187
8.2. Enkripsi Data 189
8.2.1. Enkripsi dengan *Message–Digest* 190
8.2.2. Enkripsi dengan *Crypt*. 190
8.3. Aplikasi Manajemen *User* 192
Andrew E. Rubin (Penemu Android) 205
Rangkuman 207
Daftar Istilah 207
Latihan Soal 207
Kasus : Keamanan Internet Bagi Kegiatan Online UKM 207

Bab 9 MANAJEMEN GRAFIK DAN *PAGING*

Tujuan Instruksional Umum 210
9.1. Pendahuluan 211
9.2. Manipulasi Grafik 211
9.3. Paging Halaman 213
9.3.1. Paging Halaman Secara Statis 213
9.3.1.1. Paging Statis Cara Pertama 214
9.3.1.2. Paging Statis Cara Kedua 216
9.3.2. Paging Halaman secara Dinamis 218

Dennis MacAlistair Ritchie (Penulis Bahasa Pemrograman C) 222
Rangkuman 224
Daftar Istilah 224
Latihan Soal 224

Bab 10 OOP DALAM PHP

Tujuan Instruksional Umum 226
10.1. Apa Itu OOP? 227
10.2. Kelas dan Objek 228
10.3. Variabel Anggota 228
10.4. Konstruktor dan Destruktor 230
10.5. Pembuatan Objek 230
10.6. *Inheritance* dengan *extends* 231
10.7. Polimorfisme 233
10.8. Kelas *MySQLDb* dan Postgre*SQLDb* 234
10.9. Konstruktor *MySQLDb* 234
10.10. Metode *Select()* 235
10.11. Kelas *PostgreSQLDb* 2**36**
Charles Babbage (Perancang mesin komputer) 237
Rangkuman 239
Daftar Istilah 239
Latihan Soal 239

Bab 11 PEMBUATAN APLIKASI

Tujuan Instruksional Umum 242
11.1. Konsep MVC 243
10.2. Struktur Direktori Aplikasi 244
11.3. Membuat *Layout* 244
11.4. Membuat *Controller* 251
11.5. Membuat *View* 252
Grace Murray Hopper (Penyusun Konsep COBOL) 256

Rangkuman 258
Daftar Istilah 258
Latihan Soal 258
Kasus : Jaringan Internet Membantu Hidupnya UKM 259
DAFTAR PUSTAKA 261
TENTANG PENULIS 264
GLOSARIUM 265

DAFTAR GAMBAR

Gambar 1.1	Cara Kerja *E-Commerce*	4
Gambar 1.2	Model B2C dan B2B	8
Gambar 1.3	Unsur – Unsur Pada *E-Goverment*	11
Gambar 1.4	Macam interaksi antar pelaku dalam *E-Government*	12
Gambar 2.1	Cara Kerja *Browser*	26
Gambar 6.1	*Database* Baru	148
Gambar 6.2	Tabel Pengunjung	149
Gambar 6.3	Membuat *field* pada *Database*	149
Gambar 6.5	Tabel Pengunjung	151
Gambar 6.6	Isi Tabel	151
Gambar 6.7	*Browse Tab*le	152
Gambar 6.9	*Input.php*	156
Gambar 6.10	Tambahan *Record*	157
Gambar 6.11	*View.php*	159
Gambar 6.12	*Edit.php?id=1*	161
Gambar 6.13	Cari.htm	164
Gambar 11.1	Konsep Model *View Controller*	250
Gambar 11.2	Struktur Direktori Aplikasi	251
Gambar 11.3	*Layout*	252
Gambar 11.4	*Page Not Found*	255
Gambar 11.5	Halaman Awal	257
Gambar 11.6	Tampilan Halaman Awal	260

UNIVERSITAS NAROTAMA
PRO PATRIA
SURABAYA

BAB 1

Pemahaman dan Pengenalan Mengenai *E-Commerce*

Tujuan Instruksional Umum

Setelah mempelajari bab ini, Mahasiswa dipastikan dapat:

1. Menjelaskan Pengertian *E-Commerce*
2. Menjalaskan Jenis-Jenis *E-Commerce*
3. Menjalaskan Manfaat Dan Resiko *E-Commerce*
4. Menjelaskan Cara Pembayaran Menggunakan *E-commerce*

BAB

1

PEMAHAMAN DAN PENGENALAN MENGENAI *E-COMMERCE*

1.1. Pengertian *E – Commerce*

Sebelum lebih jauh mempelajari tentang ***E–Commerce*** atau ***Electronic Commerce*** (Perdagangan Elektronik) ada baiknya kita memahami definisi maupun pemgertian dari istilah tersebut, menurut sumber yang telah dihimpun pengertian *E–Commerce* memiliki pemahaman yang beragam.

Antara lain adalah sebagai berikut:

- KAMUS ELEKTRONIK: memaknai *E-Commerce* sebagai suatu penyebaran, pembelian, penjualan, pemasaran barang dan jasa melalui sistem elektronik seperti internet atautelevisi, www, atau jaringan komputer lainnya. *E-Commerce* dapat melibatkan transfer dana elektronik, pertukaran data elektronik,sistem manajemen inventori otomatis dan sistem pengumpulan data otomatis.
- UNCITRAL (salah satu komisi dibawah PBB yang khusus membahas hukum perdagangan internasional) mendefinisikan *E-Commerce* sebagai penggunaan semua jenis informasi dalam bentuk data untuk keperluan komersial.
- **WTO** memaknai *E-Commerce* sebagai suatu proses meliputi produksi, distribusi, pemasaran, penjualan dan pengiriman barang serta jasa melalui elektronik.
- Sementara **Robert E. Johnson**, III *(http://www.cimcor.com),* berpendapat bahwa *E-Commerce* merupakan suatu tindakan melakukan transaksi bisnis secara elektronik dengan menggunakan internet sebagai media komunikasi yang paling utama.

Gary Coulter dan **John Buddiemeir** *E-Commerce* berhubungan dengan penjualan, periklanan, pemesanan produk yang semuanya dikerjakan melalui internet.

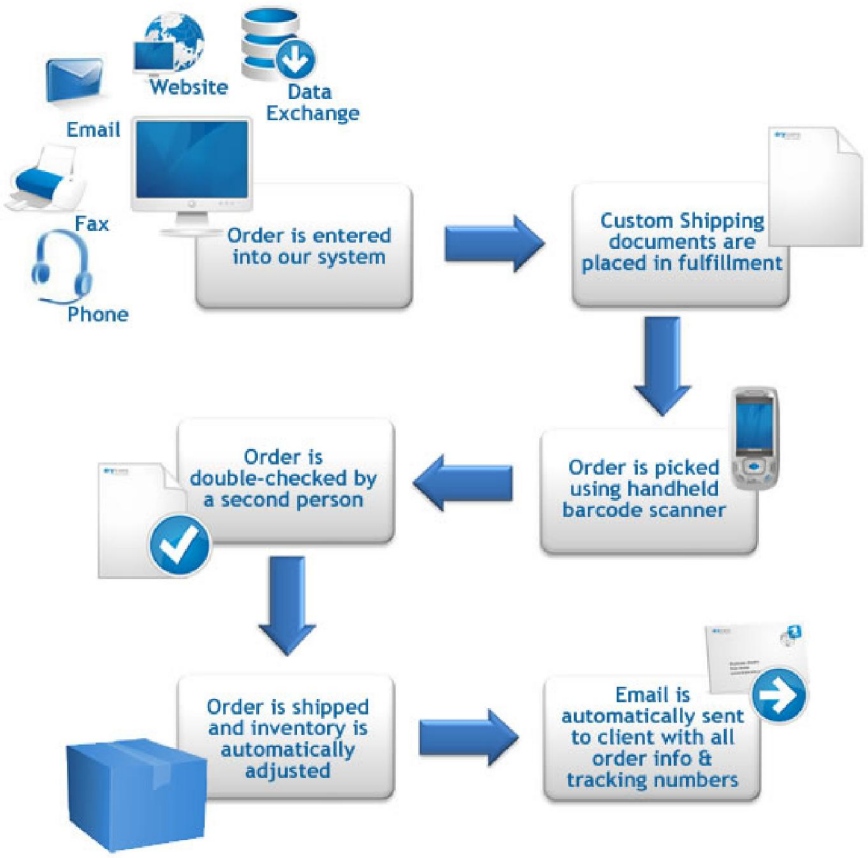
- **Gary Coulter** dan **John Buddiemeir** (seorang praktisi di *E-Commerce outline) menyimpulkan bahwa E-Commerce* berhubungan dengan penjualan, periklanan, pemesanan produk yang semuanya dikerjakan melalui internet. Beberapa perusahan memilih untuk menggunakan kegiatan bisnis ini sebagai tambahan metode bisnis tradisional, sementara yang lainnya menggunakan internet secara ekslusif dalam upayanya mendapatkan para pelangan yang berpotensi.
- Menurut Ding *E-Commerce* adalah transaksi komersial antara penjual dan pembeli atau pihak-pihak lainnya dalam hubungan kontrak yang menggunakan media elektronik atau digital yang dalam prosesnya tidak diperlukan temu muka dan transaksi dilakukan secara lintasbatas
- Sementara menurut Kalakota dan Whinston *E-Commerce* adalah suatu:
  - Aktivitas pengiriman komunikasi dan informasi, produk-produk atau jasa, atau pembayaran yang dilakukan melalui telepon, jaringan-jaringan komputer atau sarana-sarana elektronik lainnya.
  - Proses bisnis dengan mengaplikasikan teknologi untuk melakukan transaksi-transaksi bisnis atau alur kerja.
  - Sarana yang memungkinkan perusahaan-perusahaan, konsumen-konsumen dan manajemen perusahaan untuk menurunkan biaya-biaya pelayanan.
  - Sarana yang memungkinkan dilakukannya penjualan dan pembelian produk dan informasi melalui internet dan layanan-layanan *online* lainnya.

Dari berbagi uraian dan pemahaman yang telah diuraikan di atas, dapat disimpulkan bahwa *E-Commerce* adalah suatu transaksi komersial melalui jaringan komunikasi yang dapat berupa fax, email, telegram, telek, EDI *(Electronic Data Interchange),* dan sarana elektronik lainnya meliputi kegiatan tukar-menukar informasi, iklan, pemasaran, kontrak dan kegiatan perbankan melalui internet.

E-Commerce adalah suatu transaksi komersial melalui jaringan komunikasi dan sarana elektronik lainnya meliputi kegiatan tukar-menukar informasi, iklan, pemasaran, kontrak dan kegiatan perbankan melalui internet.

E–Commerce merupakan bagian dari *E-Business,* dimana cakupan *E-Business* menjangkau aspek yang lebih luas, tidak hanya sekedar dari sektor perniagaan tetapi mencakup juga pengkolaborasian mitra bisnis, pelayanan nasabah, lowongan pekerjaan dan lain-lain. Selain melalui teknologi jaringan www, *E–Commerce* juga menggunakan teknologi berbasisdata *(databases),* surat elektronik *(e-mail),* dan berbagai bentuk teknologi non komputer yang lain seperti halnya sistem pengiriman barang, dan alat pembayaran melalui jaringan *E–Commerce* tersebut.

Cikal bakal *E–Commerce* pertama kali dipublikasikan pada khalayak di tahun 1994 pada saat pertama kali *banner*-elektronik dipakai untuk tujuan promosi dan periklanan di halaman-web *(website).*



Gambar 1.1 Cara Kerja *E-Commerce*

Sumber: E-Commerce. Wordpress.com

1.2. Jenis *E – Commerce*

... seiring perkembangan zaman, kini bisnis *E-commerce* sendiri sudah sangat banyak jenisnya

Bisnis *E-commerce* berkembang dengan cepat dimasyarakat. Pada awalnya bisnis ini hanya berkembang antara pemilik bisnis dengan customer atau biasa disebut B2C *(Business to Customer)*. Namun seiring perkembangan zaman, kini bisnis *E-commerce* sendiri sudah sangat banyak jenisnya dan dapat diklasifikasikan menjadi beberapa jenis yang memiliki karakteristik berbeda-beda. Berikut ini adalah ragam E-Commerce menurut jenisnya seperti:

B2B : *Business to Business*
B2C : *Business to Customer*
C2C : *Customer to Customer*
C2B : *Customer to Business*

1.2.1. *Business to Business* (B2B)

Kebanyakan *E-Commerce* yang diterapkan saat ini merupakan tipe B2B. *E-Commerce* tipe ini meliputi transaksi IOS yang digambarkan tadi serta transaksi antar organisasi yang dilakukan di *electronic market*. Contohnya Wal-Mart dengan Warner-Lambert. *Business to Business E-Commerce* umumnya menggunakan mekanisme *Electronic Data Interchange* (EDI). Sayangnya banyak standar EDI yang digunakan sehingga menyulitkan interkomunikasi antar pelaku bisnis.

Standar yang ada saat ini antara lain: EDIFACT, ANSI X.12, SPEC 2000, CARGO-IMP, TRADACOMS, IEF, GENCOD, EANCOM, ODETTE, CII. Selain standar yang disebutkan di atas, masih ada format-format lain yang sifatnya *proprietary*. Jika anda memiliki beberapa partner bisnis yang sudah menggunakan standar yang berbeda, maka anda harus memiliki sistem untuk melakukan konversi dari satu format ke format lain. Saat ini sudah tersedia produk yang dapat melakukan konversi seperti ini.

Pendekatan lain yang sekarang cukup populer dalam standarisasi pengiriman data adalah dengan menggunakan *Extensible Markup Language* (XML) yang dikembangkan oleh *World Wide Web Consortium* (W3C). XML menyimpan struktur dan jenis elemen data di dalam dokumennya dalam bentuk *tags* seperti *HTML tags* sehingga sangat efektif digunakan untuk sistem yang berbeda. Kelompok yang mengambil jalan ini antara lain adalah XML/EDI group.

Pada mulanya EDI menggunakan jaringan tersendiri yang sering disebut *VAN (Value Added Network)*. Populernya jaringan komputer Internet memacu inisiatif EDI melalui jaringan Internet, atau dikenal dengan nama *EDI over Internet*. Topik yang juga mungkin termasuk di dalam *business-to-business E–Commerce* adalah *electronic/Internet procurement* dan *Enterprise Resource Planning* (ERP). Hal inilah yang menjadi implementasi penggunaan teknologi informasi pada perusahaan dan pada manufakturing. Sebagai contoh, perusahaan Cisco maju pesat dikarenakan menggunakan teknologi informasi sehingga dapat menjalankan *just-in-time manufacturing* untuk produksi produknya.

Adapun karakteristik B2B yaitu :

- *Trading partners* yang sudah diketahui dan umumnya memiliki hubungan (*relationship*) yang cukup lama. Informasi hanya dipertukarkan dengan partner tersebut. Dikarenakan sudah mengenal lawan komunikasi, maka jenis informasi yang dikirimkan dapat disusun sesuai dengan kebutuhan dan kepercayaan *(trust)*.
- Pertukaran data (*data exchange*) berlangsung berulang-ulang dan secara berkala, misalnya setiap hari, dengan format data yang sudah disepakati bersama. Dengan kata lain, servis yang digunakan sudah tertentu. Hal ini memudahkan pertukaran data untuk dua entiti yang menggunakan standar yang sama.
- Salah satu pelaku dapat melakukan inisiatif untuk mengirimkan data, tidak harus menunggu parternya.
- Model yang umum digunakan adalah *peer-to-peer*, at dengan ,enggunakan B2B yaitu :
- Pencapaian kesempatan berkompetisi secara nyata:
  - Produktivitas kerja yang besar dan postensial.
  - Penghematan waktu dalam melakukan transaksi.
  - Berkurangnya biaya yang harus dikeluarkan (Proses yang cepat, transparan, dan harga yang lebih murah)
- Pengurangan biaya atau pengeluaran :

E-COMMERCE atau PERDAGANGAN ELEKTRONIK

Perdagangan elektronik (Bahasa Inggris: *Electronic Commerce* atau *e-commerce*) adalah penyebaran, pembelian, penjualan, pemasaran barang dan jasa melalui sistem elektronik seperti internet atau televisi, www, atau jaringan komputer lainnya. *E-commerce* dapat melibatkan transfer dana elektronik, pertukaran data elektronik, sistem manajemen inventori otomatis, dan sistem pengumpulan data otomatis.

Industri teknologi informasi melihat kegiatan *e-commerce* ini sebagai aplikasi dan penerapan dari e-bisnis *(e-business)* yang berkaitan dengan transaksi komersial, seperti: transfer dana secara elektronik, SCM *(supply chain management),* pemasaran elektronik *(e-marketing)*, atau pemasaran *online (online marketing),* pemrosesan transaksi *online (online transaction processing),* pertukaran data elektronik *(electronic data interchange /EDI),* dan lain-lain.

E-commerce merupakan bagian dari *e-business,* di mana cakupan *e-business* lebih luas, tidak hanya sekadar perniagaan tetapi mencakup juga pengkolaborasian mitra bisnis, pelayanan nasabah, lowongan pekerjaan dll. Selain teknologi jaringan *www, e-commerce* juga memerlukan teknologi basisdata atau pangkalan data *(databases),* surat elektronik (e-mail), dan bentuk teknologi non komputer yang lain seperti halnya sistem pengiriman barang, dan alat pembayaran untuk e-dagang ini.

Dari Wikipedia bahasa Indonesia ensiklopedia bebas (Dialihkan dari E-commerce)

Tantangan B2B *(Business to Business)* yaitu:

- Data *cost* OSS yang kuat berbasis *accounting*
- Transformasi paralel dari semua orang atau budaya dalam hal proses dan teknologi
- Pembaruan adalah hal yang sangat penting
- Perundangan pemerintah
- Proses perjanjian kerja sama
- Harga dan pembayaran

1.2.2. *Business to Consumer* (B2C)

Business to Consumer E–Commerce memiliki permasalahan yang berbeda. Mekanisme untuk mendekati pelanggan pada saat ini menggunakan bermacam-macam pendekatan seperti misalnya dengan menggunakan "*electronic shopping mall*" atau menggunakan konsep "*portal*".

Electronic shopping mall menggunakan *Web sites* untuk menjajakan produk dan servis. Para penjual produk dan servis membuat sebuah *storefront* yang menyediakan katalog produk dan servis yang diberikannya. Calon pembeli dapat melihat-lihat produk dan servis yang tersedia seperti halnya dalam kehidupan sehari-hari dengan melakukan *window shopping*. Bedanya, (calon) pembeli dapat melakukan *shopping* ini kapan saja dan darimana saja dia berada tanpa dibatasi oleh jam buka toko. Contoh penggunaan *Web site* untuk menjajakan produk dan servis antara lain:

Para penjual produk dan servis membuat sebuah *storefront* yang menyediakan katalog produk dan servis yang diberikannya. Calon pembeli dapat melihat-lihat produk dan servis yang tersedia seperti halnya dalam kehidupan sehari-hari dengan melakukan *WINDOW SHOPPING*

- Amazon http://www.amazon.com
  Amazon merupakan toko buku *virtual* yang menjual buku melalui *Web* sitenya. Kesuksesan Amazon yang luar biasa menyebabkan toko buku lain harus melakukan hal yang sama.
- e-Bay http://www.ebay.com, merupakan tempat lelang *on-line.*
- NetMarket *http://www.netmarket.com,*

NetMarket yang merupakan *direct marketing* dari *Cendant* (hasil merge dari HFC, CUC International, Forbes projects). NetMarket akan mampu menjual 95% dari kebutuhan rumah tangga sehari-hari.

Konsep *portal* agak sedikit berbeda dengan *electronic shopping mal*l, dimana pengelola portal menyediakan semua servis di portalnya (yang biasanya berbasis *Web*). Sebagai contoh, portal menyediakan *e-mail* gratis yang berbasis *Web* bagi para pelanggannya sehingga diharapkan sang pelanggan selalu kembali ke portal tersebut. Contoh portal antara lain: Netscape Home (*http://home.netscape.com*), *My Yahoo, dan lain-lain.* Adapun *Business to Consumer E–Commerce* memiliki karakteristik sebagai berikut:

- Terbuka untuk umum, dimana informasi disebarkan ke umum.
- Servis yang diberikan bersifat umum *(generic)* dengan mekanisme yang dapat digunakan oleh khalayak ramai. Sebagai contoh, karena sistem *Web* sudah umum digunakan maka servis diberikan dengan menggunakan basis *Web*.
- Servis diberikan berdasarkan permohonan (*on demand*). Konsumer melakukan inisiatif dan produser harus siap memberikan respon sesuai dengan permohonan.

- Pendekatan *client/server* sering digunakan dimana diambil asumsi *client (consumer)* menggunakan sistem yang minimal (berbasis *Web*) dan *processing* (*business procedure*) diletakkan di sisi server.

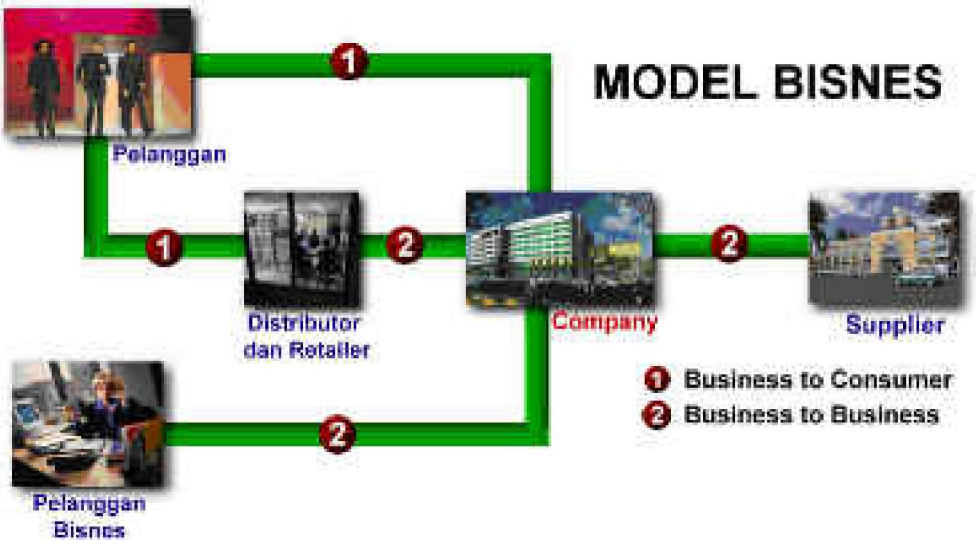
Selain itu keuntungan B2C yaitu:

- Keuntungan bagi badan bisnis:
  - Akses ke pasar global secara langsung
  - Penghematan waktu dan tempat
  - Pengurangan biaya yang sangat berarti
  - Kesediaan penuh : 24 jam perhari dan 7 hari perminggu
- Keuntungan bagi konsumen :
  - Berbelanja secara on-line tidak sesulit dari apa yang biasa didapat di pasar tradisional
  - Mudah dalam penggunaannya, tidak memerlukan kepandaian khusus
  - Banyak pilihan yang didapat dengan mudah ditambah dengan kerahasiaan yang dijamin
  - Product-on-demand (apa yang anda perlukan akan anda dapatkan)

Tantangan B2C yaitu :

- Transformasi Budaya dari tradisional ke *on-line*
- Memerlukan kepercayaan yang sangat tinggi
- Keterbatasan pembayaran (transaksi maksimum, keamanan dan lain-lain)
- Sistem pengiriman

Konsep *PORTAL* agak sedikit berbeda dengan *Electronic Shopping Mall*, dimana pengelola menyediakan semua servis di portalnya (biasanya berbasis *Web*).



Gambar 1.2 Model B2C dan B2B

Sumber: Model Bisnis. Wordpress.com

1.2.3. *Consumen To Con*sumen (C2C)

Dalam C2C seseorang menjual produk atau jasa ke orang lain. Dapat juga disebut sebagai pelanggan ke palanggan yaitu orang yang menjual produk dan jasa ke satu sama lain.

Contohnya di *priceline.com,* dimana pelanggan menyebutkan produk dan harga yang diinginkan, dan *priceline* mencoba menemukan pemasok yang memenuhi kebutuhan tersebut.

LELANG C2C. Dalam lusinan negara, penjualan dan pembelian C2C dalam situs lelang sangat banyak. Kebanyakan lelang dilakukan oleh perantara, seperti *eBay.com*, *auctionanything.com*; para pelanggan juga dapat menggunakan situs khusus seperti *buyit.com* atau *bid2bid.com*. Selain itu banyak pelanggan yang melakukan lelangnya sendiri seperti greatshop.com menyediakan piranti lunak untuk menciptakan komunitas lelang terbalik C2C *online*.

IKLAN KECIK. Orang mejual ke orang lainnya setiap hari melalui iklan kecik *(classified ad)* di koran dan majalah. Iklan kecik berbasis internet memiliki satu keunggulan besar daripada berbagai jenis iklan kecik yang lebih tradisional: iklan ini menawarkan pembaca nasional bukan hanya lokal. Iklan kecik tersedia melalui penyedia layanan internet seperti AOL, MSN, dan lain-lain.

LAYANAN PERSONAL. Banyak layanan personal (pengacara, tukang, pembuat laporan pajak, penasehat investasi, layanan kencan) tersedia di internet. Beberapa diantaranya tersedia dalam iklan kecik, tetapi lainnya dicantumkan dalam situs *Web* serta direktori khusus. Beberapa gratis dan ada juga yang berbayar

1.2.4. *Consumen To Business* (C2B).

Pada jenis C2B konsumen memberitahukan kebutuhan atas suatu produk atau jasa tertentu, dan para pemasok bersaing untuk menyediakan produk atau jasa tersebut ke konsumen. Contohnya di *priceline.com,* dimana pelanggan menyebutkan produk dan harga yang diinginkan, dan *priceline* mencoba menemukan pemasok yang memenuhi kebutuhan tersebut.

1.2.5. Perdagangan Intrabisnis (Intraorganisasional)

Dalam situasi ini perusahaan menggunakan E–Commerce secara internal untuk memperbaiki operasinya. Kondisi khusus dalam hal ini disebut sebagai e-commerce B2E(business to its employees) yang digambarkan dalam studi kasus terbuka.

1.2.6. Perdagangan Kolabratif *(Collaborativ E-Commerce)*

Dalam *e-commerce,* para mitra bisnis berkolaborasi (alih-alih membeli atau menjual) secara elektronik. Kolaborasi semacam ini seringkali terjadi antara dan dalam mitra bisnis disepanjang rantai pasokan.

1.2.7. Perdagangan *Mobile (Mobile Commerce/M-Commerce)*

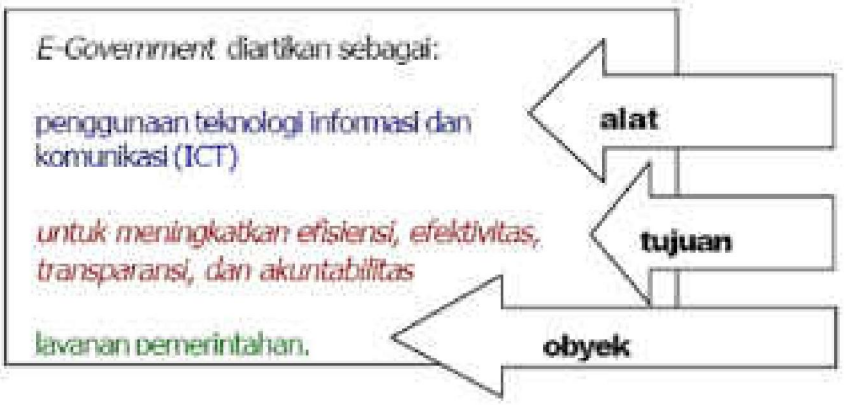
Ketika e-commerce dilakukan dalam lingkungan nirkabel, seperti dengan menggunakan telepon selluler untuk mengakses internet dan berbelanja, maka hal ini disebut *m-commerce.*

1.2.8. Intrabusiness *(Organizational) E-Commerce*

Yang termasuk dalam kategori ini adalah semua aktivitas intern organisasi, biasnya dijalankan di Internet, yang melibatkan pertukaran barang, jasa atau informasi

1.2.9. Pemerintah ke Masyarakat *(Goverment to Citizen—G2C)*

Dalam kondisi ini sebuah entitas (unit) pemerintah menyediakan layanan ke para warganya melalui teknologi *E-commerce.* Unit-unit pemerintah dapat melakukan bisnis dengan berbagai unit pemerintah lainnya serta dengan berbagai perusahaan(G2B). *E-goverment* yaitu penggunaan teknologi internet secara umum dan e-commerce secara khusus untuk mengirimkan informasi dan layanan publik ke warga, mitra bisnis, dan pemasok entitas pemerintah, serta mereka yang bekerja di sektor publik.



Gambar 1.3 Unsur-unsur pada *E-Goverment*

Sumber: Model Bisnis. Wordpress.com

E-GOVERMENT yaitu penggunaan teknologi internet secara umum dan *e-commerce* secara khusus untuk mengirimkan informasi dan layanan publik ke warga, mitra bisnis, dan pemasok entitas pemerintah, serta mereka yang bekerja di sektor publik.

E-goverment menawarkan sejumlah manfaat potensial: *E-Govermant* meningkatkan efisiensi dan efektivitas fungsi pemerintah, termasuk pemberian layanan publik. *E-goverment* memungkinkan pemerintah menjadi lebih transparan pada masyarakat dan perusahaan dengan memberikan lebih banyak akses informasi pemerintah. *E-goverment* juga memberikan peluan bagi masyarakat untuk memberikan umpan balik ke berbagai lembaga pemerintah serta berpartisipasi dalam berbagai lembaga dan proses demokrasi. Adapun konsep *E-goverment* dapat dibagi menjadi 3 (tiga) kategori:

1) **Pemerintah ke Masyarakat *(Goverment to Citizen)***. Lembaga pemerintah makin banyak yang menggunakan internet untuk menyediakan layanan pada warga.
2) **Pemerintah ke Perusahaan *(Goverment to Business)***. Pemerintah menggunakan internet untuk menjual dan membeli dari perusahaan.
3) **Pemerintah ke Pemerintah *(Goverment to Government)***. Meliputi *e-commerce* intrapemerintah (transaksi antar pemerintah yang berbeda) serta berbagai layanan antar lembaga pemerintah yang berbeda.

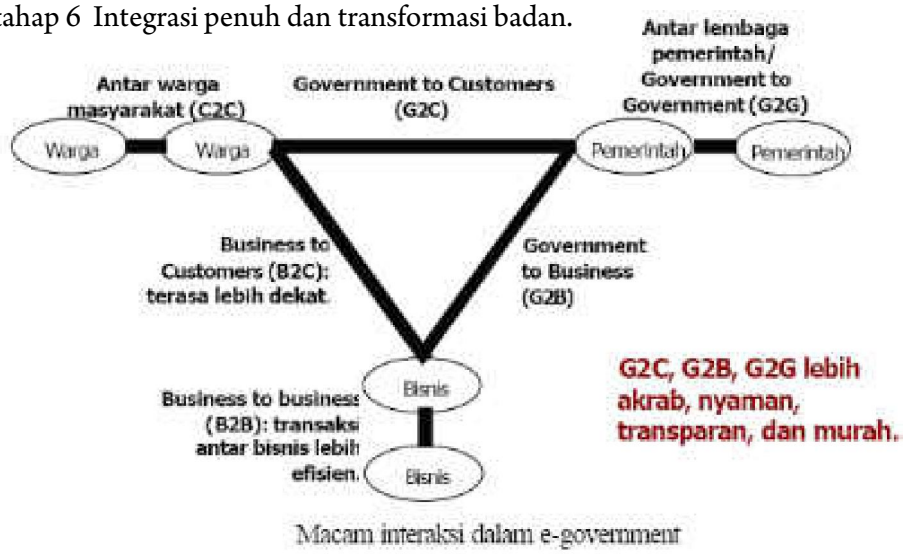
Implementasi *E-Goverment.* Transformasi dari pemberian layanan pemerintah tradisional ke implementasi penuh layanan pemerintah online dapat menjadi proses yang memakan waktu. Terdapat enam tahap dalam transformasi ke *E-Goverment*:

Tahap 1 Publikasi penyebaran informasi;

Tahap 2 Transaksi dua arah "secara resmi", dengan sebuah departemen dalam waktu yang sama;

E-GOVERMENT memungkinkan pemerintah menjadi lebih transparan pada masyarakat dan perusahaan dengan memberikan lebih banyak akses informasi pemerintah.

Tahap 3 Portal multiguna;
Tahap 4 Personalisasi portal;
Tahap 5 Pengelompokkan layanan umum;
tahap 6 Integrasi penuh dan transformasi badan.



Gambar 1.4 Macam interaksi antar pelaku dalam *E-Government*

Sumber: Model Bisnis. Wordpress.com

1.3. Standar Teknologi *E-Commerce*

Di samping berbagai standar yang digunakan di Intenet, e-commerce juga menggunakan standar yang digunakan sendiri, umumnya digunakan dalam transaksi bisnis-ke-bisnis. Beberapa diantara yang sering digunakan adalah:

Electronic Data Interchange (EDI)

EDI adalah standar struktur dokumen yang dirancang untuk memungkinkan organisasi besar untuk mengirimkan informasi melalui jaringan private.

Dibuat oleh pemerintah di awal tahun 70-an dan saat ini digunakan oleh lebih dari 1000 perusahaan Fortune di Amerika Serikat, EDI adalah sebuah standar struktur dokumen yang dirancang untuk memungkinkan organisasi besar untuk mengirimkan informasi melalui jaringan private. EDI saat ini juga digunakan dalam *corporate Web site*.

Open Buying on the Internet (OBI)

Adalah sebuah standar yang dibuat oleh *Internet Purchasing Roundtable* yang akan menjamin bahwa berbagai sistem *e-commerce* dapat berbicara satu dengan lainnya. OBI yang dikembangkan oleh konsorsium OBI *http://www.openbuy.org/* didukung oleh perusahaan-perusahaan yang memimpin di bidang teknologi seperti Actra, InteliSys, Microsoft, Open Market, dan Oracle.

Open Trading Protocol (OTP)

OTP dimaksudkan untuk melakukan standarisasi berbagai aktifitas yang berkaitan dengan proses pembayaran, seperti perjanjian pembelian, resi untuk pembelian, dan pembayaran. OTP sebetulnya merupakan standar kompetitor OBI yang dibangun oleh beberapa perusahaan, seperti AT&T, CyberCash, Hitachi, IBM, Oracle, Sun Microsystems, dan British Telecom.

Open Profiling Standard (OPS)

Sebuah standar yang didukung oleh *Microsoft* dan *Firefly http:// www.firefly. com/*. OPS memungkinkan pengguna untuk membuat sebuah profil pribadi dari kesukaan masing-masing pengguna yang dapat dia *share* (berbagi) dengan *merchant.* Ide dibalik OPS adalah untuk menolong memproteksi privasi pengguna tanpa menutup kemungkinan untuk transaksi informasi untuk proses marketing dan sebagainya.

Secure Socket Layer (SSL)

Protokol ini didisain untuk membangun sebuah saluran yang aman ke server. SSL menggunakan teknik enkripsi *public key* untuk memproteksi data yang di kirimkan melalui Internet. SSL dibuat oleh *Netscape* tapi sekarang telah dipublikasikan.

*Secure Electronic Transac*tion (SET)

SET akan mengenkodekan nomor kartu kredit yang di simpan di *server merchant.* Standar ini dibuat oleh Visa dan *MasterCard,* sehingga akan langsung didukung oleh masyarakat perbankan. Ujicoba pertama kali dari SET di *e-commerce* dilakukan di Asia.

Truste

Adalah sebuah partnership dari berbagai perusahaan yang mencoba membangun kepercayaan publik dalam *e-commerce* dengan cara memberikan cap *Good Housekeeping* yang memberikan *approve* pada situs yang tidak melanggar kerahasiaan konsumen.

1.4. Manfaat dan Resiko *E – Commerce*

Dari segi manfaat keunggulan menggunakan *E-Commerce* meliputi 3 (tiga) bagian antara lain sebagai berikut:

(1) Manfaat yang bisa diperoleh organisasi diantaranya:

- *E–Commerce* memperluas *marketplace* hingga ke pasar nasional dan Internasional
- *E–Commerce* menturunkan biaya pembuatan, peroses pendistribusian, penyimpanan dan pencarian informasi
- *E–Commerce* memungkinkan pengurangan *inventory* dan *overhead* dengan menyederhanakan *supply chain management* tipe *"pull".*
- Pemrosesan *pull-type* memungkinkan *customization* produk dan layanan yang sebetulnya mahal menjadi lebih memiliki keunggulan komparatif bagi yang menerapkan
- *E–Commerce* mengurangi waktu antara *outlay* modal dan penerimaan produk dan jasa
- *E–Commerce* mendukung upaya-upaya *business process reegineering.*
- *E–Commerce* memperkecil biaya telekomunikasi-Internet lebih murah dibanding VAN.

SSL *(SECURE SOCKET LAYER)*

Pengertian dari istilah SSL *(Secure Socket Layer)* adalah suatu protokol keamanan yang digunakan pada hampir semua transaksi aman pada internet. SSL mengubah suatu protokol transport seperti TCP menjadi sebuah saluran komunikasi aman yang cocok untuk suatu proses transaksi yang sensitif seperti *Paypal, Internet Banking,* dan lain-lain.

Dari segi keamanan protokol jenis ini dijamin mengingat dengan menggunakan kombinasi dari kiptografi kunci publik dan kriptografi kunci simetri bersamaan dengan sebuah infrastruktur sertifikat. Sebuah sertifikat adalah sebuah kumpulan data identifikasi dalam format yang telah distandardisasi. Data tersebut digunakan dalam proses verifikasi identitas dari sebuah entitas (contohnya sebuah *web server*) pada internet.

Pada protokol SSL menyediakan otentikasi (pada sisi *client,* dan opsional pada sisi server) terhadap pihak-pihak yang berkomunikasi. SSL dapat mengamankan koneksi antara dua titik, dan tidak ada pihak yang dapat melakukan hal-hal yang bersifat destruktif atau mengakses informasi yang bersifat sensitif. SSL menyediakan sebuah saluran komunikasi yang aman tanpa perlu adanya pertemuan kedua pihak yang berkomunikasi untuk melakukan proses pertukaran kunci.

Fungsi SSL pada komunikasi aman sama seperti fungsi TCP pada komunikasi normal, yaitu menyediakan sebuah infrastruktur komunikasi standar dimana sebuah aplikasi dapat menggunakannya dengan mudah dan hampir tidak dapat terlihat *(invisible).*

SSL menyediakan sebuah komponen penting pada sistem yang aman. Mekanisme otentikasi

Ide dibalik OPS adalah untuk menolong memproteksi privasi pengguna tanpa melakukan transaksi informasi dalam proses marketing dan ...

dasar seperti password Telnet dan otentikasi HTTP dasar menjadi sangat kuat ketika dieksekusi dengan SSL dibandingkan dengan TCP, dimana pada *SSL password* tidak lagi dikirim dalam bentuk plainteks.

SSL mengenkripsi koneksi, bukan data pada kedua pihak yang berkomunikasi, dan tidak mengandung mekanisme untuk otentikasi user ataupun perlindungan *password* (hanya koneksi yang diotentikasi, keamanannya akan gagal jika mesin pada kedua pihak yang berkomunikasi *compromised*).

Implementasi SSL pertama dikembangkan oleh Netscape Communications Corporation pada awal tahun 1990-an untuk mengamankan HTTP. Pada akhir tahun 1990-an, semakin terlihat dengan jelas bahwa SSL 2.0 tidaklah aman. Netscape memulai untuk membangun SSL 3.0. Dengan bantuan Netscape, *Internet Engineering Task Force* (IETF, badan yang mengatur untuk standar internet) memulai untuk menstandardisasi SSL, sebuah proyek yang kemudian dikenal dengan nama TLS *(Transport Layer Security).*

SSL 3.0 tidak dikembangkan seteliti TLS, sehingga SSL 3.0 dapat dirilis lebih dahulu dan menggantikan SSL 2.0 sebagai standar industri. TLS yang akhirnya diselesaikan pada tahun 2000, menyediakan protokol terstandardisasi yang pertama untuk SSL. Walaupun SSL 3.0 masih digunakan secara luas, untuk pengembangan terbaru termasuk teknologinya sudah tertinggal karena saat ini hampir semua browser modern mendukung TLS.

Materi disadur dari : http://ilmukomputer.org/2013/04/30/ssl-secure-socket-layer/

(2) Manfaat ***E-Commerce*** bagi **konsumen** diantaranya:

- *E–Commerce* memungkinkan pelanggan untuk berbelanja atau melakukan transaksi lain selama 24 jam sehari sepanjang tahun dari hampir setiap lokasi.
- *E–Commerce* meemberikan lebih banyak pilihan kepada pelanggan; *E–Commerce* menyediakan produk-produk dan jasa yang tidak mahal kepada pelanggan dengan cara mengunjungi banyak tempat dan melakukan perbandingan secara cepat.
- Pelanggan bisa menerima informasi relevan secara detail dalam hitungan detik, bukan lagi hari atau minggu.
- *E–Commerce* memberi tempat bagi para pelanggan untuk berinteraksi dengan pelanggan lain di *electronic community* dan bertukar pkiran serta berbagai pengalaman.

(3) Manfaat ***E-Commerce*** bagi **masyarakat** diantaranya adalah:

- E–Commerce memungkinkan orang untuk bekerja di dalam rumah dan tidak banyak keluar untuk berbelanja,
- E–Commerce memungkinkan sejumlah barang dagangan dijual dengan harga lebih rendah,
- E–Commerce memungkinkan orang di negara-negara Dunia ketiga dan wilayah pedesan untuk menikmati aneka produk dan jasa
- E–Commerce memfasilitasi layanan publik, seperti perawatan kesehatan, pendidikan, dan pemerataan layanan sosial yang dilaksanakan pemerintah dengan biaya yang lebih rendah, dan/atau dengan kualitas yang lebih baik

Dari segi resiko *yang* dapat terjadi dalam transaksi berbasis *E-Commerce* dapat disebabkan karena:

(1) penyalahgunaan dan kegagalan sistem terdiri dari:

- Kehilangan segi finansial secara langsung karena kecurangan, misal seseorang telah menghancurkan/ mengganti semua data finansial yang ada.
- Pencurian informasi rahasia yang berharga, misal pencurian terhadap kepemilikan teknologi, informasi pemasaran atau informasi yang berhubungan dengan kepentingana konsumen
- Kehilangan kesempatan bisnis karena gangguan peservice, misal gangguan yang bersifat nonteknis, seperti aliran listrik mati.
- Penggunaan akses ke sumber oleh pihak yang tidak berhak, misal seorang hacker berhasil membobol sistem perbankan dengan berhasil memindahkan sejumlah rekening orang lain ke dalam rekening peribadinya.
- Kehilangan kepercayaan dari para konsumen, misal seringnya terjadi gangguan pada jaringan yang menyebabkan akses gagal.

E–COMMERCE menyediakan produk-produk dan jasa yang tidak mahal kepada pelanggan dengan cara mengunjungi banyak tempat dan melakukan perbandingan secara cepat ...

- Kerugian-kerugian yang tak terduga, misal gangguan terhadap transaksi bisnis, akibat kesalahan faktor manusia atau kesalahan perangkat.

Infrastruktur Telekomunikasi

Infrastruktur Telekomunikasi di Indonesia masih terbatas dan harganya masih relatif lebih mahal

bisnis-online1 (www.teropongku.com)

Delivery Channel

- Kehilangan barang saat pengiriman barang
- Ketepatan waktu dalam pengiriman barang
- Jangkauan daerah pengiriman barang

Kultur dan Kepercayaan

- Orang Indonesia belum terbiasa berbelanja dengan menggunakan secara *online,* masih harus secara fisik melihat/memegang barang yang dijual
- Kepercayaan antara penjual dan pembeli masih belum terjalin
- Kepercayaan kepada pembayaran elektronik masih kurang. Penggunaan kartu kredit masih terhambat kondisi yang ada

1.5. Klasifikasi Transaksi *E – Commerce*

Terdapat 3 (tiga) klasifikasi transaksi yang seringkali dilakukan *E-Commerce* antara lain:

Online Processing Credit Card

Merupakan jenis pembayaran yang dilakukan secara *online* dan *real time* dangan melakukan proses transaksi menggunakan *credit card.*

Money Transfer

Cara ini dinilai relatif lebih aman, karena konsumen akan mentransfer langsung uang yang akan dibayarkan untuk membeli suatu produk.

Cash On Delivery

Sistem bayar ditempat, sistem ini hanya dapat dilakukan jika konsumen berada dalam satu kota dengan pihak penyedia produk. Berikut ini beberapa contoh alamat situs yang telah menerapkan *e-commerce* antara lain:

http://www.mypaysystems.com,

http://www.ccnow.com,

http://www.uni-books.comm

Inksomnia (http://inksomniastreetwear.com.comm)

Semacam forum tetapi didalamnya juga ada proses jual beli barang Kaskus *(http://kaskus.uss)*

RAY TOMLINSON | Penemu E-mail (Elektronik Mail)

Ray Tomlinson si Penemu *electronic mail*

Nama Lengkap:
Raymond Samuel Tomlinson

Tempat dan Tanggal Lahir:
Amsterdam, New York,
Amerika Serikat
23 April 1941

Meninggal:
5 Maret 2016 (umur 74)
Lincoln, Massachusetts,
Amerika Serikat

Kebangsaan:
Amerika Serikat

Almamater:
Rensselaer Polytechnic Institute
Massachusetts Institute of Technology

Pekerjaan:
Pemrogram, penemu

Hasil Penemuan:
Penemu sistem surat elektronik

(Sumber: https://id.wikipedia.org/wiki/Ray_Tomlinson)

Raymond Samuel Tomlinson atau Ray Tomlinson dilahirkan pada tahun 1941 di daerah bernama Amsterdam di wilayah New York. Ia dan keluarganya kemudian pindah ke daerah atau desa kecil bernama Vail Mills, New York. Ray Tomlinson dikenal sebagai Penemu dari Email atau Elektronik Mail. Di Saat remaja Ray Tomlinson bersekolah di Broadalbin Central School di daerah Broadalbin, New York. Setelah Lulus dari Broadalbin Central School, Ray Tomlinson kemudian melanjutkan srudinya di Rensselaer Polytechnic Institute di wilayah Troy, New York di mana Ray Tomlinson bekerja sama dengan IBM dalam program co-op. Di Rensselaer Polytechnic Institute, Ray Tomlinson menerima gelar Bachelor of Science di bidang teknik listrik pada tahun 1963.

Setelah lulus dari Rensselaer Polytechnic Institute, Ray Tomlinson kemudian melanjutkan studinya untuk gelar Master di Massachusetts Institute of Technology dalam bidang teknik elektro. Di MIT, Ray Tomlinson bekerja di Speech Communication Group dan disitu Ray Tomlinson mengembangkan teknologi analog-digital hybrid speech synthesizer yang dijadikan sebagai subyek untuk tesis Master-nya. Ray Tomlinson kemudian menerima gelar S.M. di bidang Teknik Elektro pada tahun 1965.

Pada tahun 1967 Ray Tomlinson bergabung dengan perusahaan teknologi bernama Bolt, Beranek and Newman yang sekarang dikenal dengan nama BBN Technologies, di mana Ray Tomlinson membantu mengembangkan sistem operasi TENEX termasuk ARPANET Network Control Protocol dan implementasi TELNET. Ray Tomlinson menulis sebuah program transfer file yang disebut CPYNET untuk mentransfer file melalui ARPANET. Ray Tomlinson diminta untuk mengubah sebuah program yang disebut SNDMSG, yang mengirim pesan ke pengguna lain dari komputer time-sharing, untuk dapat dijalankan pada TENEX. Dia menambahkan kode yang ia ambil dari CPYNET ke SNDMSG sehingga pesan dapat dikirim ke pengguna pada komputer lain inilah yang menjadi cikal bakal dari munculnya E-mail atau elektronik mail.

Raymond Tomlinson menemukan email atau elektronik mail berbasis internet akhir tahun 1971 di ARPAnet. Inovasi terbesar yaitu email (atau surat elektronik), kemampuan untuk mengirim pesan sederhana kepada orang lain melalui jaringan (1971). saat itu tengah mengerjakan program yang disebut SNDMSG (send message)yang memungkinan pengguna komputer yang sama bisa meninggalkan pesan satu sama lain. Atau sistem e-mail dengan jenis komputer tunggal. Ketika itu, ia mencoba sebuah program transfer (CYPNET) yang memungkinkan pengguna komputer mengirimkan file, untuk mengarahkan komputer terhubung dengan ARPANET. Tomlinson kemudian berpikir bahwa jika ia menggabungkan SNDMSG dan CYPNET berbarengan, ia mungkin bisa mengirim pesan ke kotak pos elektronik lain dalam sebuah jaringan, semudah mengirimkan file.

Salah satu pertanyaan yang kemudian mengarahkan Tomlinson pada eksperimen lain mengenai e-mail adalah bagaimana membedakan pesan-pesan yang diarahkan keluar jaringan dan pesan lain yang dialamatkan ke banyak pengguna komputer pada sebuah kantor. Ia terus mempelajari keyboard komputer yang tak langsung terkait dengan nama orang serta bukan sebuah angka. Akhirnya sampailah ia pada penggunaan simbol @ sebagai petunjuk alamat sang pengguna e-mail. Simbol ini mewakili tempat dan hanyalah sebuah preposisi pada sebuah keyboard. Penggunaan simbol @ ini membawa dampak yang global, Ray Tomlinson mengakui bahwa penemuannya dihasilkan “hanya dari berpikir tak lebih dari 30 atau 40 detik.” Ray Tomlinson memilih simbol @ untuk membedakan mana pengguna/username dengan domain/alamat server.. Simbol @ dibaca “et” yang berarti “di”.

Email pertama yang dikirim oleh Raymond Tomlinson di antara dua komputer yang benar-benar duduk di samping satu sama lain. Namun, jaringan ARPANET digunakan sebagai sambungan antara keduanya. Pesan email yang pertama adalah “QWERTYUIOP”. Pesan QWERTYUIOP dapat jadikan pesan bersejarah karena itu isi pesan pertama yang dapat dikirmkan lewat jaringan ARPANET yang berkembang sekarang menjadi jaringan internet. Penemu e-mail , Raymond Tomlinson, yang menurut banyak kalangan termasuk pencapaian penting dan revolusioner dalam teknologi informasi. Dia layak menjadi 50 tokoh yang berpengaruh dalam dunia Informatika karena dia yang memilih simbol @ sebagai penanda alamat surat elektronik, dibelakang nama si pengirim atau penerima pesan, bahkan kini pengunaan e-mail sudah menjadi sarana komunikasi global yang paling murah.

Penghargaan yang diterima Raymond Samuel Tomlinson

Pada tahun 2000 ia menerima George R. Stibitz Komputer Pioneer Award dari Museum Komputer Amerika (dengan Ilmu Komputer Departemen Montana State University).

Pada tahun 2001 ia menerima Penghargaan Webby dari International Academy of Digital Arts and Sciences untuk pencapaian seumur hidup. Juga pada tahun 2001 ia dilantik ke Rensselaer Alumni Hall of Fame.

Pada tahun 2002 Discover Magazine diberikan kepadanya Penghargaan Inovasi nya.

Pada tahun 2004, ia menerima Penghargaan Internet IEEE bersama dengan Dave Crocker.

Pada tahun 2009, ia bersama dengan Martin Cooper dianugerahi Prince of Asturias penghargaan untuk penelitian ilmiah dan teknis.

Pada tahun 2011, ia terdaftar di # 4 pada daftar MIT150 dari 150 inovator atas dan ide-ide dari MIT.

Pada 2012, Tomlinson telah dilantik ke dalam Hall of Fame internet oleh Internet Society.

Rangkuman

1. *E-Commerce* dapat melibatkan transfer dana elektronik, pertukaran data elektronik,sistem manajemen inventori otomatis dan sistem pengumpulan data otomatis.
2. *E–Commerce* memungkinkan orang di negara-negara Dunia ketiga dan wilayah pedesan untuk menikmati aneka produk dan jasa
3. *E-Commerce* berhubungan dengan penjualan, periklanan, pemesanan produk yang semuanya dikerjakan melalui internet
4. *E–Commerce* merupakan bagian dari *E-Business,* di mana cakupan *E-Business* lebih luas, tidak hanya sekedar perniagaan tetapi mencakup juga pengkolaborasian mitra bisnis, pelayanan nasabah, lowongan pekerjaan dan lain-lain
5. *E-goverment* menawarkan sejumlah manfaat potensial: *E-govermant* meningkatkan efisiensi dan efektivitas fungsi pemerintah, termasuk pemberian layanan publik
6. Konsep *portal* agak sedikit berbeda dengan *electronic shopping mal*l, dimana pengelola portal menyediakan semua servis di portalnya (yang biasanya berbasis *Web*).

Daftar Istilah

1. Electronic cash
2. Digital money
3. Disintermediation
4. Electronic checks
5. Electronic wallet
6. Extranet
7. Micropaymet

Latihan Soal

1. Jelaskan pengertian dari *E-Commerce* menurut kampus elektronik, UNCITRAL , WTO dan kesimpulan anda?
2. Sebutkan dan jelaskan pengertian dari jenis-jenis *E-Commerce* (minimal 5 jenis)!
3. Sebutkan keuntungan dan tantangan dari *Business to Business* dan *Business to Consumer!*
4. Sebutkan dan jelaskan kategori dari *E-Goverment*!
5. Sebutkan manfaat dan resiko *E-Commerce* yang anda ketahui!
6. Sebutkan klasifikasi transaksi *E-Commerce*!

KASUS DALAM JARINGAN UKM

Masalah yang sampai saat ini masih perlu diperhatikan adalah kemampuan pengusaha UKM mengakses pasar yang lebih luas. Dengan Produksi yang sudah cukup bagus bila pasar yang dijangkau terbatas maka tidak akan cukup menolong kelangsungan hidup UKM. Karena itu diperlukan langkah-langkah mengatasi masalah pemasaran produksi Usaha Kecil dan Menengah ini dari pihak pemerintah tentunya. Untuk mengatasi masalah pemasaran produksi UKM, Nining mengatakan sudah saatnya pelaku usaha mempersiapkan diri untuk memasuki era baru pemasaran produk UKM. Sebab selama ini kelemahan akses pasar menjadi persoalan bagi hampir seluruh pelaku UKM di tanah air.

Mereka masih saja kurang memiliki informasi yang lengkap dan rinci, terkait pasar mana saja yang bisa ditembus oleh produk yang dihasilkan. Di sinilah UKM, terlihat begitu lemah. Akses pasar UKM tidak terbentuk secara luas dan tidak ada jaringan usaha yang kokoh. Padahal kemampuan mengakses pasar merupakan salah satu kunci pokok untuk memenangkan persaingan. Sehingga penting pagi pengusaha kecil memperluas jaringan pemasaran produksi, tegasnya.

Beberapa kendala yang sering dihadapi oleh pengusaha UKM di bidang pasar dan pemasaran menurut Nining antara lain kesulitan mendapatkan suplai bahan baku berkualitas dan kontinyu, terbatasnya kemampuan untuk melakukan promosi dan berkompetisi di pasar. Kurang diperhatikannya mutu produk dan arti kepuasan pelanggan. Pelaku UKM juga cenderung menguasai pasar yang sempit, sebagai akibat lemahnya kemampuan untuk berkompetisi dengan perusahaan besar, dan kurangnya kesempatan yang diberikan oleh perusahaan besar untuk menjadikan UKM sebagai mitra bisnisnya. Packaging yang tidak menarik sehingga kurang diminati.

(Sumber : http://tiangazerockz.blogspot.com/2013/01/ukm-dan-masalah-pemasaran.html)

UNIVERSITAS NAROTAMA

BAB 2

Pemahaman Tentang Web Master

Tujuan Instruksional Umum

Setelah mempelajari bab ini, Mahasiswa dipastikan dapat:

1. Menjelaskan Pengertian dan Spesifikasi Dari *Webmaster*
2. Menjelaskan Langkah-Langkah Pembuatan Situs yang Baik
3. Menjelaskan Teknologi dari *Webmaster*
4. Menjelaskan Langkah-Langkah Pembuatan Web yang Baik

BAB

2

PEMAHAMAN TENTANG *WEB MASTER*

2.1. Pengertian Webmaster

Pada Dunia web, kita mengenal banyak istilah, beberapa diantaranya adalah web designer, web programer, web administrator, dan web ad ministrator. Secara umum, webmaster mencakup semua profesi tersebut. Kenapa demikian? Karena webmaster adalah sebutan untuk orang yang ahli di dunia web sehingga seorang webmaster tidak hanya bisa mendesain halaman web saja, tetapi juga harus dapat membuat sistemnya dan merawat website tersebut hingga tetap ada dan melindunginya dari kerusakan, baik dari dalam maupun bila menghadapi serangan dari luar.

tugas seorang webmaster adalah merancanakan, merancang, mengelola, mengembangkan dan memelihara situs web agar tetap dapat berjalan dengan baik

Secara garis besar dapat disimpulkan tugas seorang webmaster adalah merancanakan, merancang, mengelola, mengembangkan dan memelihara situs web agar tetap dapat berjalan dengan baik serta dapat dinikmati oleh pengunjung

Seorang webmaster juga harus terus belajar, berbagi, dan berkarya. Karena pada hakikatnya seorang layak disebut sebagai webmaster bukan karena jabatan yang disandangnya, namun karena karya dan kontribusi yang diberikannya. Jika Anda sudah memahami bahwa seorang webmaster bukan berorientasi pada jabatan, namun pada karya yang dihasilkannya, ayo kita mulai berusaha menjadi seorang webmaster.

2.2. Pengertian Spesifikasi webmaster

Kadang karena semakin kompleksnya dunia web, istilah webmaster itu digunakan sebagi julukan untuk sebuah tim, bukan seorang saja. Untuk langkah awal belajar sebagai seorang webmaster, maka sebaiknya anda menentukan spesifikasi anda. Spesifikasi yang bisa dipilih antaralain, sebagai berikut:

Perbedaan tugas dari seseorang yang berprofesi sebagai *Web Designer, Web Developer* dan *Web Master* — Banyak orang yang agak bingung apa bedanya pekerjaan web designer dan web developer serta ada istilah lain yaitu web master. Sebagian orang bahkan ada yang menganggap semua pekerjaan tersebut sama. Makanya tak ayal banyak orang yang sebenarnya hanya memiliki skill sebagai web designer , tetapipada kenyataannya mendapatkan tawaran pekerjan atau diminta untuk membuatkan website oleh seseorang.

Berikut ini adalah ulasan singkat tentang apa perbedaan antara web designer, web developer dan web master.

2.2.1. *Web Designer*

Web designer yaitu orang yang bekerja dengan unsur-unsur visual pada suatu halaman web. Web designer adalah orang yang tugasnya membuat tampilan halaman website sehinga tampak begitu cantik dan sedap dipandang mata. Sebagai seorang *Web Designer* sebaiknya Anda mempunya jiwa seni sehingga anda bisa melakukan pemilihan *design, layout,* dan warna yang akan membuat pengunjung situs Anda merasa nyaman, takjub, dan menjadi pengunjung setia *website* yang Anda rancang.

Biasanya desainer web mengintegrasikan berbagai komponen seperti file gambar, file video seperti *flash*(namun sekarang *flash* ini sudah tidak direkomendasikan), atau multimedia lain ke dalam sebuah halaman website untuk menambah pengalaman visual bagi para pengguna, atau untuk melengkapi halaman konten website.

Adapun sebuah Design merupakan elemen utama yang dilihat secara visual oleh user, bila kesan petama ini menarik, maka selanjutnya ada kemungkinan situs anda akan dikunjungi terus. Beberapa point penting bagi seorang *Web Designer* dalam merancang halaman *web* adalah sebagai berikut:

✓ Perhatikan kompabillitas resolusi monitor pengunjung

✓ Desain harus dirancang semenarik mungkin dan sesuai dengan tujuan situs

✓ Warna yang dipilih tidak terlalu terang ataupun gelap.

✓ Navigasi yang baik

Software yang dapat digunakan oleh seorang *Web Designer* untuk mendesain halaman web adalah Adobe Photoshop, Adobe Illustrator, fireworks, freehand, dan lain-lain. Pemahaman tentang *Client-side scripting* seperti HTML, *CSS, Javascript* serta mengerti cara memanipulasi gambar/foto dan animasi adalah beberapa hal yang mutlak dikuasai oleh seorang web desainer.

Web designer biasanya bekerja dalam sebuah tim untuk memastikan sebuah tampilan website bisa membuat aplikasi web bekerja dengan baik.

Hal lain yang juga harus dimengerti oleh seseorang yang berprofesi sebagai *web designer* adalah masalah *usability* atau kegunaaan, standar W3C untuk HTML dan CSS serta kompabilitas tampilan pada browser yang berbeda, bahkan yang terbaru web desainer harus mengerti masalah tampilan website yang harus bisa dibuka di setiap perangkat yang memiliki ukuran layar berbeda atau biasa diserbut desain web responsif atau responsive web design.

Tugas seorang webmaster adalah merancanakan, merancang, mengelola, mengembangkan dan memelihara situs web agar tetap dapat berjalan dengan baik serta dapat dinikmati oleh pengunjung

Seorang web designer yang handal akan terbiasa dengan bahasa pemograman web, baik itu yang ada di bagain server *(server side scripting)* maupun dari sisi klien *(Client-side scripting)*. Aspek tampilan biasanya menjadi hal yang tidak terlalu dipusingkan oleh seorang web developer, dia pasti memahami HTML dan CSS tapi desain *website* itu juga membutuhkan kreativitas dan jiwa seni, jadi walaupun web developer memahami setiap perintah yang ada di HTML dan CSS, dia tidak bisa dipekerjakan sebagai *web designer* karena tidak memiliki kemampuan dalam membuat desain, hanya untuk mengaplikasikan desain yang telah dibuat menjadi lebih dinamis pastilah dikuasai oleh web developer.

2.2.2. *Web Programer*

Web Developer adalah orang yang pekerjaannya menciptakan aplikasi berbasis website, pada prakteknya tugas dari seorang web developer adalah membuat berbagai hal bisa "terjadi" pada sebuah website. Peran web developer adalah sebagai penghubung dari semua sumber daya yang nantinya digunakan pada sebuah website, yang meliputi penggunaan database, membuat halaman yang dinamis, sampai mengatur cara pengunjung berinteraksi dengan elemen yang ada di suatu website yang dibuatnya.

...tugas dari seorang web developer adalah membuat berbagai hal bisa "terjadi" pada sebuah website.

Web Developer atau*Web Programer* sebaiknya harus bisa menguasai beberapa bahasa pemrograman web dan sejenisnya seperti HTML, DHTML, Java, PHP, .NET, atau Python, Javascript, VBScript, Applet, ASP, PHP, ColdFusion, WML, XML, dan masih banyak lagi. Secara umum tugas dari seseorang yang bergelut di bidang *Web Programer* atau *Web Developer* antara lain:

- ✓ Koding
- ✓ Membuat, merancang, dan mengelola database
- ✓ Menggabungkan antara desain dan program

Beberapa software yang dapat digunakan yaitu *Dreamweaver, frontpage, Notepad, Ultraedit,* dan lain lain.

2.2.3. *Web Master*

Web Master bisa saja orang yang memiliki dua kemampuan yang kita bahas diatas, namun kelebihannya adalah web master ini juga bertugas untuk memelihara aplikasi web yang telah dibuat, memastikan website selalu bisa diakses atau tidak down. Bahkan webmaster juga bisa saja bertanggung jawab untuk optimasi mesin pencari (SEO), mengisi konten website, hingga memasarkan website itu sendiri kepada publik.

Namun pada skup kelompok kerja yang lebih luas, *webmaster* akan lebih banyak ditugaskan untuk menjadi manager dari sebuah proyek berbasis web, mengatur pekerjaan web developer dan web designer, sampai melakukan pengujian terhadap usability atau kebergunaan aplikasi web yang terlah dibuat sebelumnya, melakukan testing, bahkan urusan keamanan website.

Bagi seorang web developer, kemampuan logis sangatlah diperlukan untuk mengatur alur aplikasi web, sedangkan pada web designer membutuhkan kemampuan artistik dan juga kreativitas, yang pasti harus memiliki jiwa seninya harus kuat atau memiliki sentuhan artistik. Sedangkan Webmaster akan lebih banyak membutuhkan kemampuan logis, artistik, serta pengalaman yang baik dalam permasalahan umum yang ada di dunia web, seperti mengatur file yang ada di *hosting,* memastikan *website* bebas *hacker, website* tidak *down,* dan hal penting lainnya.

2.2.4. *Web Animator*

Tugas web animator adalah membuat animasi yang dapat menarik perhatian dan minat pengunjung. Dalam membuat animasi web animator juga harus memperhatikan rancangan website yang ada, sehingga animasi tidak merusak atau memperburuk design yang sudah ada. Software yang dapat digunakan antara lain Macromedia flash, swish, gif animator, livemotion, dll.

2.2.5. *Web Administrator*

Web administrator bertanggung jawab atas berjalannyaq sebuah situs ketika sudah selesai dibuat. Web administrator harus mengetahui dan menguasai perbedaan web server yang ada, baik kelebihannya maupun kekurangannya. Sedangkan tugasnya antara lain:

- ✓ Menginstal web, dan mengatur konfigurasi server
- ✓ Mengatur DNS dalam sistem
- ✓ Mengatur account untuk admin dan user
- ✓ Mengatur konfigurasi keamanan dan firewall.

Melihat kompleksitas dari tugas seorang web administrator, maka setidaknya dia harus menguasai IIS, Apache server, Telnet, Windows, NT/ 2000, Linux, Solaris.

2.2.6. *Web Content Creator*

Web Content Creator bertugas menambahkan content/ mengkonsep content pada website yang sesuai dengan tujuan pembuatan website tersebut.

2.3. Tahapan Pembuatan Situs

Sama seperti membuat sebuah software, untuk membuat sebuah website yang baik juga diperlukan adanya tahapan-tahapan yang harus dilakukan. Setiap tahapan selesai dilakukan, harus dievaluasi apakah tahapan itu telah berjalan dengan baik. Secara garis besar, tahapan itu adalah sebagai berikut:

Sumber: http://img.okezone.com

PERETAS (bahasa Inggris: Hacker) adalah orang yang mempelajari, menganalisis, memodifikasi, menerobos masuk ke dalam komputer dan jaringan komputer, baik untuk keuntungan atau dimotivasi oleh tantangan.

Terminologi peretas muncul pada awal tahun 1960-an di antara para anggota organisasi mahasiswa Tech Model Railroad Club di Laboratorium Kecerdasan Artifisial Massachusetts Institute of Technology (MIT). Kelompok mahasiswa tersebut merupakan salah satu perintis perkembangan teknologi komputer dan mereka berkutat dengan sejumlah komputer mainframe.

Kata bahasa Inggris "hacker" pertama kalinya muncul dengan arti positif untuk menyebut seorang anggota yang memiliki keahlian dalam bidang komputer dan mampu membuat program komputer yang lebih baik daripada yang telah dirancang bersama.

Kemudian pada tahun 1983, istilah hacker mulai berkonotasi negatif. Pasalnya, pada tahun tersebut untuk pertama kalinya FBI menangkap kelompok kriminal komputer "The 414s" yang berbasis di Milwaukee, Amerika Serikat. 414 merupakan kode area lokal mereka. Kelompok yang kemudian disebut hacker tersebut dinyatakan bersalah atas pembobolan 60 buah komputer, dari komputer milik Pusat Kanker Memorial Sloan-Kettering hingga komputer milik Laboratorium Nasional Los Alamos. Satu dari pelaku tersebut mendapatkan kekebalan karena testimonialnya, sedangkan 5 pelaku lainnya mendapatkan hukuman masa percobaan.

Kemudian pada perkembangan selanjutnya muncul kelompok lain yang menyebut-nyebut diri sebagai peretas, padahal bukan. Mereka ini (terutama para pria dewasa) yang mendapat kepuasan lewat membobol komputer dan mengakali telepon (phreaking). Peretas sejati menyebut orang-orang ini cracker dan tidak suka bergaul dengan mereka. Peretas sejati memandang cracker sebagai orang malas, tidak bertanggung jawab, dan tidak terlalu cerdas.

Peretas sejati tidak setuju jika dikatakan bahwa dengan menerobos keamanan seseorang telah menjadi peretas.Para peretas mengadakan pertemuan tahunan, yaitu setiap pertengahan bulan Juli di Las Vegas. Ajang pertemuan peretas terbesar di dunia tersebut dinamakan Def Con. Acara Def Con tersebut lebih kepada ajang pertukaran informasi dan teknologi yang berkaitan dengan aktivitas peretasan.

Peretas memiliki konotasi negatif karena kesalahpahaman masyarakat akan perbedaan istilah tentang *hacker* dan *cracker.* Banyak orang memahami bahwa peretaslah yang mengakibatkan kerugian pihak tertentu seperti mengubah tampilan suatu situs web *(defacing),* menyisipkan kode-kode virus, dan lain-lain, padahal mereka adalah *cracker. Cracker*-lah menggunakan celah-celah keamanan yang belum diperbaiki oleh pembuat perangkat lunak (bug) untuk menyusup dan merusak suatu sistem. Atas alasan ini biasanya para peretas dipahami dibagi menjadi dua golongan: *White Hat Hackers,* yakni hacker yang sebenarnya dan *cracker* yang sering disebut dengan istilah *Black Hat Hackers.*

Peretas menurut Eric Raymond didefinisikan sebagai programmer yang pandai. Sebuah *hack* yang baik adalah solusi yang cantik untuk masalah pemrograman dan *hacking* adalah proses pembuatannya. Menurut Raymond ada lima (5) karakteristik yang menandakan seorang adalah *hacker,* yaitu:

1. Seseorang yang suka belajar detail dari bahasa pemrograman atau sistem.
2. Seseorang yang melakukan pemrograman, tidak cuma teori saja.
3. Seseorang yang bisa menghargai, menikmati hasil hacking orang lain.
4. Seseorang yang dapat secara cepat belajar pemrograman.
5. Seseorang yang ahli dalam bahasa pemrograman tertentu atau sistem tertentu, seperti UNIX hacker

(Sumber: https://id.wikipedia.org/wiki/Peretas)

1. *PLANNING* (perencanaan)

Perencanaan yang baik akan sangat membantu dan menghemat waktu dalam membuat sebuah situs. Hal-hal yang harus dilakukan pada tahap ini adalah:

- ✓ Tentukan tujuan situs yang akan dibuat
- ✓ Pengumpulan data (user gathering) kemudian menganalisisnya
- ✓ Membuat timeline atau jadwal kerja

2. *DESIGN*

Pada tahap design ini buatlah design yang menarik, pilihan warna yang sesuai(biasanya 3 warna dasar untuk sebuah website). Design ini sangat penting karena merupakan kesan pertama dengan user. Design yang ada harus membawa imej, tujuan, dan karakteristik yang ingin disampaikan.

3. *CODING* (Pemrograman)

Coding adalah tahap implementasi planning dan design yang telah Anda buat. Pada tahap ini Anda akan mengimplementasikan ide-ide dan design yang ada kedunia nyata sesuai dengan teknologi yang dibutuhkan(HTML, css, javascript, php, mysql, dll). Setelah coding selesai Anda melanjutkan ke tahap Testing.

4. *TESTING* (Uji coba)

Pengujian dilakukan untuk meyakinkan bahwa semua fungsionalitas website telah berjalan dengan baik. Tahapan ini dilakukan untuk kontrol kualitas situs yang Anda buat. Tahapan ini dilakukan baik diserver lokal, maupun di server hosting.

5. PROMOSI

Setelah melakukan tahap uji coba, saatnya Anda meluncurkan situs web Anda ke internet untuk mengatakan hallo kepada seluruh duniaJ Promosi merupakan media yang sangat penting untuk membuat situs Anda dikenal dan dilihat oleh banyak orang. Beberapa cara mempromosikan antara lain :

- ✓ Promosikan melalui brosur, poster, kartu nama, kop surat
- ✓ Melalui Search Engine
- ✓ Melalui hubungan kerjasama pemasaran dengan situs lainnya, sehingga Anda dapat menampilkan link di situs tersebut

6. *MAINTENANCE*

Setelah situs diluncurkan maka diperlukan tahap pemeliharaan di situs tersebut. Pemeliharaan itu mencakup beberapa hal, antara lain:

- ✓ Selalu mengupdate situs dengan isi-isi yang terbaru. Contohnya detik.com yang menerapkan frekuensi pengupdatean yang tinggi.
- ✓ Buanglah informasi-informasi yang sudah kadaluarsa.
- ✓ Jagalah server Anda dari serangan *hacker* baik orang maupun *software*.
- ✓ Tambahlah situs Anda dengan fitur-fitur yang beru dan menarik
- ✓ Memperbarui *design* agar *user* tidak bosan

2.4. Teknologi webmaster

Beberapa Teknologi yang umum dan harus dimengerti oleh webmaster antara lain :

2.4.1. URL *(Uniform Resource Locator)*

URL digunakan sebagai penunjuk lokasi pada internet. URL adalah alamat sebuah website, seperti http://*www.google.com,* http://*www.yahoo.com , ftp://anime.ee.itb.ac.id* . Http atau ftp adalah protokolnya, sedangkan *www.google.com* merupakan alamat server website tersebut.

Tanpa URL situs Anda tidak akan bisa dikunjungi. Saat ini sudah ada sekitar 4 milyaran situs yang ada di internet artinya ada sekitar 4 milyaran URL yang ada di internet.

2.4.2. TCP/IP

TCP/IP adlah singkatan dari Transfer Control Protokol/Internet Protokol. TCP/IP merupakan protokol yang memungkinkan seluruh komputer yang ada di internet untuk saling berkomunikasi. Kita Bisa mengibaratkan protokol TCP/IP sebagai bahasa seluruh komputer yang terhubung dalam jaringan internet.

2.4.3. IP Address

Setiap Komputer yang terhubung ke internet akan memiliki IP address yang berbeda-beda(unik). Pemberian IP Address ini diatur oleh TCP/IP. TCP/IP menggunakan pengalamatan 32 bit yang tersusun atas 4 angka yang berkisar antara 0 sampai 255 yang dipisahkan oleh tanda titik.

```
Contoh IP Address misalnya 167.205.3.2
```

2.4.4. DNS(Domain Name System)

Domain adalah alamat situs Anda di internet, sedangkan IP Address adalah sebuah susunan angka yang unik, yang memberikan suatu alamat pada komputer yang terhubung ke jaringan internet.

Setiap Komputer yang terhubung ke jaringan TCP/IP akan memiliki IP address yang unik. IP address 32 bit ini bukan merupakan hal yang mudah diingat karena merupakan sebuah deretan angka yang unik. DNS ada karena manusia akan lebih mudah mengingat nama daripada angka.

DNS sebenarnya adalah representasi dari IP address, sebagai contoh :

`http://www.google.com` dengan `http://64.233.167.99`

Kedua url itu akan menampilkan halaman yang sama, yaitu situs google.

TINGKATAN PERETAS *(HACKER)*

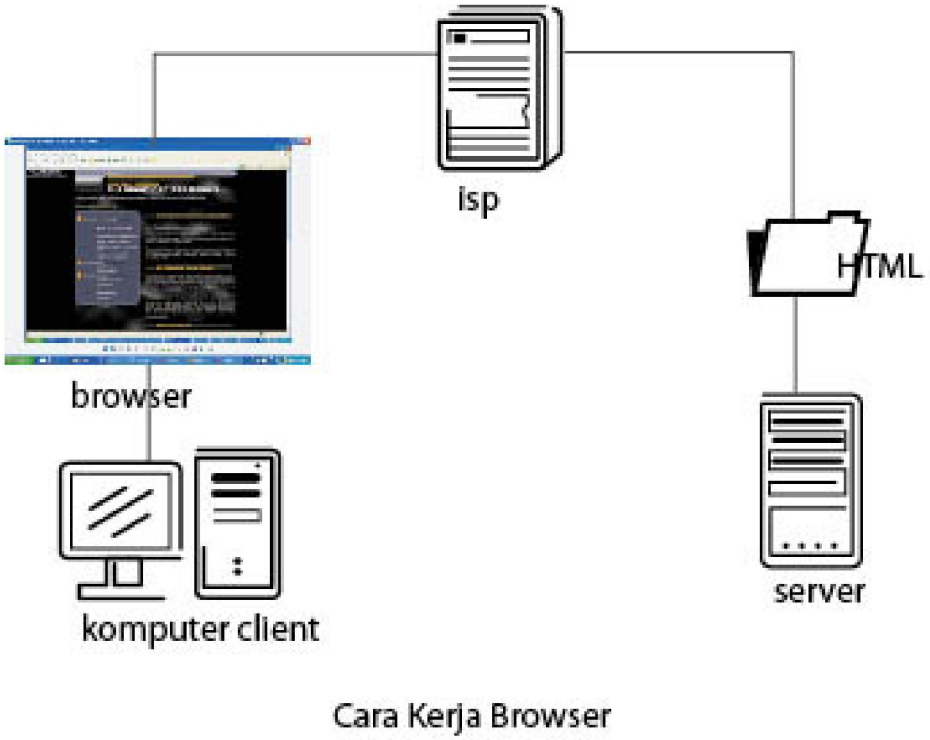
Elite
Juga dikenal sebagai 3l33t, 3l337, 31337, leet, 1337 atau kombinasi dari itu; merupakan ujung tombak industri keamanan jaringan. Mereka memahami sistem operasi sisi luar dalam, sanggup mengkonfigurasi dan menyambungkan jaringan secara global. Sanggup melakukan pemrograman setiap harinya. Sebuah anugerah yang sangat alami, mereka biasanya efisien & terampil, menggunakan pengetahuannya dengan tepat. Mereka seperti siluman yang dapat memasuki sistem tanpa terdeteksi, walaupun mereka tidak akan menghancurkan data-data yang ditemui. Karena mereka selalu mengikuti peraturan yang ada.

Semi elite
Hacker ini biasanya lebih muda daripada Elite. Mereka juga mempunyai kemampuan dan pengetahuan luas tentang komputer. Mereka mengerti tentang sistem operasi (termasuk lubangnya (vulnerability)). Biasanya dilengkapi dengan sejumlah kecil program cukup untuk mengubah program eksploit. Banyak serangan yang dipublikasi dilakukan oleh hacker tingkat ini. Sialnya oleh para Elite mereka sering kali dikategorikan Lamer.

Developed kiddie
Sebutan ini terutama karena umur kelompok ini masih muda (ABG) dan masih sekolah. Mereka membaca tentang metode hacking dan caranya di berbagai kesempatan. Mereka mencoba berbagai sistem sampai akhirnya berhasil dan memproklamasikan kemenangan ke peretas lainnya. Umumnya mereka masih menggunakan Grafic User Interface (GUI) dan baru belajar hal dasar dari UNIX, tanpa mampu menemukan lubang kelemahan baru di sistem operasi.

2.4.5. Browser

Browser adalah sebuah program yang digunakan untuk menampilkan halaman web. Browser berkomunikasi dengan web server melalui protokol http, meminta dokumen html, lalu membaca dan menerjemahkan dokumen html tersebut serta menampilkannya ke client. Gambaran cara kerja browser adalah sebagai berikut:



Gambar 2.1 Cara Kerja Browser

Sumber: browser. Wordpress.com

Kriteria Browser yang baik, antara lain:

- ✓ Mampu menerjemahkan HTML, lebih baik bila sudah mendukung XHTML
- ✓ Dapat menerjemahkan scripting web seperti javascript,Vbscript, CSS,XML, dan RSS
- ✓ Dukungan terhadap plugins seperti *flashplayer, quicktime, Java Applet,* dan sebagainya.

Plugins adalah program tambahan pada browser untuk menjalankan suatu aplikasi. Berikut ini adalah deskripsi singkat beberapa browser.

Internet Explorer

Internet Explorer(IE) adalah browser yang sepaket dalam sistem operasi microsoft windows. IE merupakan browser yang paling banyak digunakan oleh pengguna internet, jadi jangan sampai terlewat melihat tampilan situs yang anda buat dengan browser ini.

Opera

Kelebihan browser ini adlah kecepatannya dalam mendownload dari suatu halaman web.

Firefox

Browser firefox merupakan browser baru yang mengunakan mozilla sebagai mesinnya. Firefox adalah browser yang paling kecil ukurannya saat diinstall. Kelebihan yang lainnya adalah kecepatannya dalam mendownload yang tidak kalah dengan opera. Disamping itu browser ini mendukung berbagai plugins, dan dapat menampilkan halaman web dalam tab-tab. Menurut survey, banyak netter yang kemudian beralih ke browser ini.

Browser Statistik

Browser statistik digunakan untuk mengetahui jumlah user yang menggunakan browser tertentu yang mengunjungi situs anda, anda juga bisa menambahkan fitur ini, untuk mendapatkan feedback browser yang paling diminati user.

Script kiddie

Seperti developed kiddie, Script Kiddie biasanya melakukan aktivitas di atas. Seperti juga Lamers, mereka hanya mempunyai pengetahuan teknis networking yang sangat minimal. Biasanya tidak lepas dari GUI. Hacking dilakukan menggunakan trojan untuk menakuti dan menyusahkan hidup pengguna Internet.

Lamer

Mereka adalah orang tanpa pengalaman dan pengetahuan yang ingin menjadi peretas (wanna-be hacker). Mereka biasanya membaca atau mendengar tentang hacker dan ingin menjadi seperti mereka. Penggunaan komputer mereka hanyalah untuk main game, IRC, tukar-menukar perangkat lunak bajakan dan mencuri kartu kredit. Melakukan hacking menggunakan perangkat lunak trojan, nuke, dan DoS. Biasanya menyombongkan diri melalui IRC channel. Karena banyak kekurangan untuk mencapai elite, dalam perkembangannya mereka hanya akan sampai tingkat developed kiddie atau script kiddie saja.

(Sumber: https://id.wikipedia.org/wiki/Peretas)

2.5. Perencanaan dan Perancangan Web

Sekarang kita telah memahami tentang perencanaan dan perancangan web. Perencanaan adalah tahap yang harus dilakukan sebelum membuat situs web. Beberapa hal yang harus dipersiapkan saat membuat situs web antara lain:

- ✓ Tentukan terlebih dulu tujuan dan target dari situs yang akan anda buat.
- ✓ Tentukan target pengunjung anda dan apa saja yang dapat dilakukan pengunjung di situs yang akan anda buat.
- ✓ Rencanakan sistem situs anda dengan matang, intinya managelah situs anda dengan baik
- ✓ Buat sitemap dari situs yang akan anda buat.(ini akan sangat membantu)
- ✓ Buatlah time schedule pembuatan website dan disiplinlah dengan jadwal yang telah anda buat tersebut.

2.6. Tujuan Pembuatan Situs Web

Dalam membuat sebuah situs, boleh dikatakan bahwa tujuan adalah sebuah hal yang fundamental, jangan sampai Anda membuat situs yang tanpa arti dan tujuan. Bentuk design, isi, bahkan sampai pemilihan warna yang akan Anda gunakan itu pun sangat bergantung pada tujuan situs itu dibuat dan imej yang ingin Anda sampaikan. Beberapa tujuan pembuatan situs, antara lain :

2.6.1. Alat Pemasaran

Pemasaran melalui internet akan lebih cepat dan jangkauannya akan lebih luas. Situs yang digunakan sebagai alat pemasaran, didalamnya akan berisi company profile ataupun promosi produk atau jasa pelayanan yang ada pada suatu perusahaan. Beberapa hal yang harus diperhatikan pada situs dengan tujuan alat pemasaran adalah :

- ü Desain halaman web harus menarik
- ü Menampilkan latar belakang perusahaan
- ü Situs ini hanya berisi informasi dan tidak dapat melakukan transaksi online melalui situs ini
- ü Informasi tertang produk atau jasa harus jelas dan menarik

Salah satu contoh situs yang digunakan sebagai alat pemasaran adalah *http://www.darisa-im.com* yang memasarkan produk-produk kimia ke seluruh dunia, atau *http://www.oracle.com* yang berisi company profile dan informasi terbaru tentang software oracle.

2.6.2. Nilai Tambah

Sebuah situs web dibuat hanya sebagai nilai tambah. Jadi maksudnya adlah sebenarnya perusahaan tersebut sudah sangat dikenal dan tidak perlu lagi mempromosikan perusahaan mereka melalui situs, namun mereka membuat situs agar perusahaannya tetap mengikuti perkembangan jaman dan tidak dibilang kuno, contohnya adalah *http://www.anjjabar.go.id*

2.6.3. Katalog

Sebuah situs yang berfungsi sebagai katalog elektronik yang menampilkan produk-produk yang akan dijual oleh suatu perusahaan. Alasannya dibuat website ini adalah sebagai tempat mempromosikan produk-produk sebuah perusahaan, hal ini lebih murah, cepat, dan efisien bila dibandingkan dengan katalog yang dicetak (media cetak), disamping itu pula, bila ada produk yang batu maka situs tersebut dapat segera di update. Pada situs ini pembayaran terhadap produk yang dibeli tidak dapat dilakukan secara online.

CARA KERJA HACKER

Untuk melindungi komputer ketika membuka Intenet, kita perlu mengetahui cara kerja hacker mengakses suatu sistem, secara sederhana dapat digambarkan sebagai berikut: *Hacker* merupakan 'seni' tersendiri yang melibatkan proses mencari serpihan-serpihan informasi yang bertebaran di mana-mana dan seolah-olah tidak ada hubungannnya satu sama lain. Untuk memberi gambaran secara keseluruhan proses *hacking,* di bawah ini disajikan langkah-langkah logisnya, yaitu:

1. ***Footprinting***. Mencari rincian informasi terhadap sistem-sistem untuk dijadikan sasaran , mencakup pencarian informasi dengan mesin pencari, *whois,* dan DNS *one transfer.*
2. ***Scanning***. Terhadap sasaran tertentu dicari pintu masuk yang paling mungkin. Digunakan *ping sweep* dan *port scan*
3. ***Enumeration*** Telaah intensif terhadap sasaran, yang mencari *user account* absah, *network resource and share*, dan aplikasi untuk mendapatkan mana yang proteksinya lemah.
4. ***Gaining Access.*** Mendapatkan data lebih banyak lagi untuk memulai mencoba mengakses sasaran. Meliputi atau merampas kata sandi, menebak kata sandi, serta melakukan *buffer overflow*
5. ***Escalating Privilege***. Apabila baru mendapatkan *user password* ditahap sebelumnya, ditahap ini diusahakan mendapat *privilese* admin jaringan dengan *password cracking* atau eksploit sejenis *getadmin, sechole* atau *lc_messages.*
6. ***Pilfering***. Proses pengumpulan informasi dimulai lagi untuk mengidentifikasi mekanisme untuk mendapatkan akses ke *trusted system.* Mencakup evaluasi trust dan pencarian *cleartext password* di *registry, config file,* dan *user data.*
7. ***Convering Tracks***. Begitu kontrol penuh terhadap sistem diperoleh, maka menutup jejak menjadi prioritas. Meliputi membersihkan *network log* dan penggunaan *hide tool* seperti macam-macam *rootkit* dan *file streaming*
8. ***Creating Backdoors***. Pintu belakang diciptakan pada berbagai bagian dari sistem untuk memudahkan masuk kembali ke sistem ke sistem ini dengan cara membentuk *user account* palsu, menjadwalkan *batch joob* ,mengubah *startup file* ,menambahkan servis pengendali jarak jauh serta *monitoring tool* ,dan menggantikan aplikasi dengan *qtrojan.*
9. ***Denial of Service***. Apabila semua usaha di atas gagal, penyerang dapat dilumpuhkan sasaran sebagai usaha terakhir. Meliputi *SYN flood*, teknik-teknik ICMP, *supernuke, land/ latierra, teardrop, bonk, newtear, trincoo, smurf,* dan lain-lain.

ADA LOVELACE | Programer Wanita Pertama (1815–1852)

Image of Ada King, Countess of Lovelace, 1840. by Science & Society Picture Library Source=http://www.fathom.com/feature/122251/3134_adalovelace_LG.html |Date= |Author= |Permission={{PD-old}} |other_versions= }}

Nama Lengkap:
Augusta Ada Byron

Tempat dan Tanggal Lahir:
London, Inggris 10 Desember 1815

Meninggal:
27 November 1852 (umur 36)
Marylebone, London, Inggris

Kebangsaan : British
Gelar : Countess Lovelace
Pasangan:
William King-Noel, Earl pertama dari Lovelace
Anak
Byron King-Noel, Viscount Ockham dan Baron Wentworth ke-12
Anne Blunt, Baroness Wentworth ke-15
Ralph King-Milbanke, Earl ke-2 dari Lovelace
Orang tua
George Gordon Byron, Baron Byron ke-6
Anne Isabella Milbanke, Baroness Wentworth ke-11
(Sumber: https://id.wikipedia.org/AdaLovelace/bahasa_Indonesia,)

Augusta Ada King, Countess Lovelace (lahir di London, Inggris, 10 Desember 1815 – meninggal di Marylebone, London, 27 November 1852 pada umur 36 tahun), lahir dengan nama Augusta Ada Byron dan sekarang dikenal dengan nama Ada Lovelace, adalah penulis dan matematikawan Inggris yang terkenal atas karyanya berupa komputer mekanika pertama. Penemuannya atas mesin ini juga termasuk penggunaan algoritma pertama yang dimaksudkan untuk di proses oleh mesin. Atas karyanya ini, ia sering dianggap sebagai programer komputer pertama di dunia.[1][2][3]

Ada merupakan putri tunggal penyair Lord Byron (dengan Anne Isabella Byron). Ia tidak pernah berhubungan dengan ayahnya, yang bercerai dengan ibunya sebulan setelah ia lahir, dan empat bulan kemudian, ayahnya meninggalkan Inggris dan wafat di Yunani pada tahun 1823, saat Ada berusia 8 tahun. Saat remaja, Ada tertarik pada matematika, dan juga pada mesin analitikal karya Babbage. Antara tahun 1842 dan 1843, dia menerjemahkan sebuah artikel karangan matematikawan Italia Luigi Menabrea mengenai mesin, yang menjadi cikal bakal penemuannya. Kajiannya mengenai artikel ini dianggap sebagai program komputer pertama di dunia — dan juga algoritma pertama yang dikodekan untuk diproses oleh mesin. Karyanya ini berperan penting dalam perkembangan awal komputer.[4]

Lahir pada 10 Desember 1815, Ada merupakan putri dari George Gordon Byron dan Annabella. Sang ayah, Byron, merupakan penyair terkenal pada masa itu. Tapi sayangnya Ada kecil tak pernah mengenal sang ayah karena usia pernikahan yang singkat antara Byron dan Annabella. Ada kecil pun juga jarang mendapat perawatan dan kasih sayang dari ibunya. Ada justru diasuh oleh neneknya, Judith Milbanke, yang memang sangat menyayanginya. Hal itu diperburuk dengan kondisi Ada yang sering sakit-sakitan. Ia pernah lumpuh selama hampir setahun dan harus berjalan menggunakan kruk pada 1831.

Sejak kecil, Annabella memang melatih anaknya untuk lebih sering menggunakan otak kirinya. Hal itu didukung dengan kegemaran Ada terhadap bidang matematika. Saat berusia 17 tahun, kemampuan Ada dalam bidang tersebut mulai terlihat. Salah seorang guru Ada, Augustus De Morgan, sampai meminta agar ia lebih diarahkan ke bidang tersebut karena De Morgan melihat ada bakat istimewa dalam diri Ada.

Pertemuannya dengan Charles Babbage menjadi awal dari kariernya. Babbage sendiri adalah matematikawan Inggris yang pertama kali mengemukakan pendapat tentang komputer yang dapat diprogram. Singkat cerita, Ada tertarik dengan program yang saat itu tengah dijalankan Babbage, yakni Difference Engine. Sebaliknya, Babbage juga kagum dengan intelektual dan bakat menulis Ada. Babbage bahkan memberikan julukan "The Enchantress of Numbers" pada Ada. Keduanya pun lantas menjalin kerjasama.

Ada kemudian membantu Babbage menulis program komputer pertama di dunia yang diciptakan lewat mesin ciptaan Babbage yang bernama Analytical Engine. Dalam hal ini, peran besar Ada adalah menerjemahkan artikel Luigi Menabrea, seorang matematikawan Italia. Ada juga menulis penjelasan awal tentang komputer dan perangkat lunak, serta menambahkan catatan mengenai metode kalkulasi nomor Bernoulli pada mesin yang dikembangkan Babbage tersebut. Karena hal itulah Ada dikenal sebagai programmer komputer pertama di dunia.

Kanker rahim dan pendarahan yang dideritanya membuat Ada meninggal dunia pada 27 November 1852 di usia 36 tahun, yang membuat Ada gagal melanjutkan pengembangannya seputar dunia komputer. Atas kontribusinya, pada 1980 Departemen Pertahanan Amerika Serikat menamai bahasa pemrograman komputernya dengan nama "Ada" sebagai penghormatan pada Ada. Selain itu, British Computer Society pada 2008 rutin menggelar kompetisi ilmu komputer bagi siswa perempuan secara rutin tiap tahunnya dan menamai medalinya dengan nama "Ada". Setiap pertengahan Oktober, digelar acara "Ada Lovelace Day", sebuah acara yang bertujuan untuk mengingkatkan kemampuan perempuan dalam bidang ilmu pengetahuan, teknologi, teknik, dan matematika.

Gelar Penghargaan:

10 Desember 1815 – 8 Juli 1835: The Honourable Ada Augusta Byron

8 Juli 1835 – 1838: The Right Honourable The Lady King

1838 – 27 November 1852: The Right Honourable The Countess of Lovelace

Publikasi dan Referensi:

Sketch of the Analytical Engine Invented by Charles Babbage, Esq. with notes by trans. Ada Lovelace, in Scientific Memoirs, Vol 3 (1842)

1. J. Fuegi and J. Francis, "*Lovelace & Babbage and the creation of the 1843 'notes*'". IEEE Annals of the History of Computing 25 No. 4 (October–December 2003): 16–26. Digital Object Identifier
2. "*Ada Byron, Lady Lovelace*". Diarsipkan dari versi asli tanggal 21 July 2010. Diakses tanggal 11 July 2010.
3. "*Ada Lovelace honoured by Google doodle*". The Guardian. Retrieved 10 December 2012
4. Fuegi and Francis 2003 pp. 19, 25.

Rangkuman

1. Secara garis besar dapat disimpulkan tugas seorang webmaster adalah merancanakan, merancang, mengelola, mengembangkan dan memelihara situs web
2. seorang web administrator, maka setidaknya dia harus menguasai IIS, Apache server, Telnet, Windows, NT/ 2000, Linux, Solaris
3. Web designer adalah orang yang tugasnya mendesain halaman web. Sebagai seorang designer sebaiknya Anda mempunya jiwa seni
4. *Internet Explorer*(IE) adalah browser yang sepaket dalam sistem operasi microsoft windows
5. Pemasaran melalui internet akan lebih cepat dan jangkauannya akan lebih luas. Situs yang digunakan sebagai alat pemasaran, didalamnya akan berisi company profile ataupun promosi produk atau jasa pelayanan yang ada pada suatu perusahaan
6. Sebuah situs web dibuat hanya sebagai nilai tambah. Jadi maksudnya adlah sebenarnya perusahaan tersebut sudah sangat dikenal dan tidak perlu lagi mempromosikan perusahaan mereka melalui situs

Daftar Istilah

1. Publisher
2. Ad Units
3. Link Units
4. AdSense for Content
5. Alternate Ads
6. Channels
7. Page Impressions
8. Clicks
9. CTR *(Clickthrough Rate)*
10. CPC *(Cost Per Click)*
11. eCPM *(Effective CPM)*

Latihan Soal

1. Sebutkan dan jelaskan spsifikasi dari webmaster!
2. Sebutkan langkah – langkah tahapan pembuatan situs!
3. Sebutkan dan jelaskan macam – macam pengertian dari teknologi webmaster!

PERMASALAHAN UKM TERKAIT DENGAN *WEBMASTER*

Usaha Kecil dan Menengah sebagai unit kerja yang kecil akan memiliki karakteristik utama:

– Sumber daya manusia yang terbatas

– Kompleksitas jaringan komputer yang rendah

– Dukungan finansial yang terbatas

Dengan karakteristik-karakteristik yang telah disebutkan di atas, maka diperlukan suatu pendekatan yang berbeda dalam mengelola keamanan jaringan komputer jika dibandingkan dengan perusahaan yang besar. Dengan dukungan finansial yang terbatas, maka pemilihan teknologi menjadi area yang sangat penting. Diperlukan teknologi yang tidak mahal namun berkemampuan tinggi dan tidak membutuhkan biaya perawatan yang tinggi. Karena itu, penggunaan teknologi *open-source* menjadi pilihan yang tepat.

Selain memerlukan biaya yang rendah untuk implementasinya, teknologi open-source cukup aman sehingga tidak memerlukan pengelolaan yang sulit. Namun pemilihan teknologi juga terkait erat dengan kemampuan sumber daya manusia yang ada. Oleh karena itu, perlu diperhatikan juga kapabilitas yang dimiliki oleh sumber daya manusia yang tersedia, jangan sampai dipilih teknologi yang dimana tidak ada sumber daya manusia yang mampu mengelolanya.

Dengan sumber daya manusia yang terbatas, maka topologi jaringan yang dibentuk harus cukup sederhana, sehingga tidak membutuhkan banyak personel untuk melakukan pengelolaan jaringan komputer. Topologi jaringan komputer yang terlalu kompleks akan membutuhkan banyak peralatan jaringan komputer, yang selain akan membutuhkan biaya lebih tinggi, juga akan membutuhkan lebih banyak upaya untuk mengelolanya.

Hipertext Markup Language (HTML) dan Cascading Style Sheet (CSS)

Tujuan Instruksional Umum

Setelah mempelajari bab ini, Mahasiswa dipastikan dapat:

1. Menjelaskan Pengertian *Hipertext Markup Language* (HTML)
2. Menjelaskan Konsep Dasar Pembuatan HTML
3. Menjelaskan Pengertian *Cascading Style Sheet (CSS)*
4. Menjelaskan Konsep Dasar Pembuatan CSS

BAB

3

PEMAHAMAN TENTANG *HIPERTEXT MARKUP LANGUAGE*, *CASCADING STYLE SHEET*

3.1. *Hipertext Markup Language* (HTML)

3.1.1. Pengertian *Hipertext Markup Language* (HTML)

HTML (*HyperText Markup Language*) adalah format standard yang digunakan untuk membuat halaman web. HTML merupakan sub set dari bahasa yang lebih luas, yaitu SGML (*Standard Generalized Markup Language*). Selain HTML, subset lain dari SGML antara lain XML (*eXtensible Markup Language*), SMIL (*Synchronized Multimedia Integration Language*), dan MathML (*Mathematical Markup Language*). Versi terakhir HTML saat ini adalah HTML 4.01. Sedangkan versi ke-5 masih dirilis, untuk melihat spesifikasi standard dari HTML, silahkan Anda mengunjungi situs resminya di http://www.w3.org/TR/html4.

File HTML berupa file teks yang terdiri dari 2 bagian: *content* (isi), yaitu berupa teks yang akan ditampilkan oleh browser dan *markup* atau *tags* yang menjelaskan bagaimana teks tersebut diinterpretasi oleh browser.

File HTML berupa file teks yang terdiri dari 2 bagian: *content* (isi), yaitu berupa teks yang akan ditampilkan oleh browser dan *markup* atau *tags* yang menjelaskan bagaimana teks tersebut diinterpretasi oleh browser. Tag-tag yang digunakan HTML mempunyai format sebagai berikut:

```
<tag pembuka> teks </tag penutup>
```

Selain bentuk berpasangan seperti di atas, ada pula tag-tag yang berdiri sendiri. Tag-tag ini tidak *case sensitive* sehingga kita bisa menggunakan huruf kapital, huruf kecil, atau keduanya. Diantara kedua tag dapat disisipkan tag lainnya, namun penulisannya tidak boleh tumpang tindih seperti ini:

```
<tag 1><tag 2> teks </tag 1></tag 2>
```
, seharusnya

Tag-tag tertentu memiliki atribut yang ditambahkan untuk memberi detail atau memodifikasi perintah tag. Contohnya pada tag   :

```
<font color="red" face="Arial" size="3">
```

Atribut dalam tag di atas yaitu color, face, dan size. Sebuah tag dapat memiliki beberapa atribut. Penulisannya dengan dipisahkan oleh space; urutan atribut tidaklah penting. Setiap atribut mempunyai *value* tertentu.

3.1.2. Skema Dasar HTML

Sebuah file HTML umumnya terdiri dari 2 bagian utama: *head* dan *body*; keduanya dilingkupi oleh tag <html> </html>. Agar lebih jelas, perhatikan skema di bawah ini:

```
<html>
      <head>
            <title> Judul Halaman Web </title>
            <!-- berbagai header -->
            <!-- dokumen -->
      </head>
      <body>
            <!-- Isi dokumen yang ditampilkan -->
            <!-- di halaman browser -->
      </body>
</html>
```

Keterangan:

Tag	Fungsi
<html> </html>	penanda dokumen HTML
<head> </head>	berisi keterangan mengenai dokumen
<body> </body>	berisi bagian yang ditampilkan dalam halaman browser
<!-- -->	Komentar dalam file HTML

Beberapa spasi yang berdekatan, indentasi (*tab*), dan *carriage return* (*enter*), akan diinterpretasi oleh browser sebagai satu spasi. Sehingga penulisan skema diatas tidak akan mempengaruhi tampilan. Bahkan, sangat dianjurkan agar lebih mudah dibaca. Untuk membuat *multiple space*, HTML mempunyai kode tertentu untuk menampilkannya, yaitu   (satu kode ini berarti satu buah spasi). Selain untuk spasi, tersedia beberapa kode lain untuk merepresentasikan karakter tertentu.

Beberapa spasi yang berdekatan, indentasi (*tab*), dan *carriage return* (*enter*), akan diinterpretasi oleh browser sebagai satu spasi.

<lt;	<	•	•
>	>	×	×
&	&	÷	÷
"	"	→	→
®	®	←	←
©	©		

3.1.3. Membuat Teks

Sebagai penyedia informasi, suatu halaman web akan selalu melibatkan teks. HTML memberikan fasilitas untuk mengolah teks tersebut. Beberapa tag yang sering digunakan untuk memformat teks antara lain:

Tag	Fungsi
<h*n*> </h*n*>	Membuat *heading*; n = 1,2,...,6 (ukuran)
 	Memformat teks dengan warna, ukuran, dan jenis huruf tertentu
	Teks yang dilingkupi ditampilkan dengan huruf tebal (**bold**)
	Teks yang dilingkupi ditampilkan dengan huruf miring (*italic*)
<u> </u>	Teks yang dilingkupi ditampilkan bargaris bawah (underlined)
 	Tanda pindah baris
	Teks yang dilingkupi ditampilkan sebagai satu paragraf
<hr />	Membuat garis horisontal
	Membuat list berurut (*ordered list*)
	Membuat list tidak berurut (*unordered list*)
	Mendaftar item list
	Membuat *subscript*
	Membuat *superscript*
	Membuat teks rata tengah
<dl> </dl>	Membuat *definition list*
<pre> </pre>	Menampilkan teks yang diformat terlebih dahulu

Tag-tag di atas sangat berguna untuk mengatur tampilan informasi yang akan kita sajikan pada halaman web. Berikut contoh penggunaan tag-tag di atas:

Heading

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

Tampilan pada browser :

Tag heading mencerminkan fungsi utama HTML yaitu untuk mengatur struktur teks. Pemanfaatannya seperti sebuah outline dalam pembuatan tulisan.

3.1.3.1. Membuat Font

```
<font size="1" color="black" face="Comic Sans MS">Huruf
1</font>
<font size="2" color=#12fd45 face="Trebuchet">Huruf
2</font>
<font size="5" color=rgb(232,23,4) face="Arial">Huruf
3</font>
```

Tampilan pada browser:

Huruf 1 Huruf 2 Huruf 3

Value untuk atribut warna bisa dinyatakan dengan berbagai cara. Cara yang pertama menggunakan warna dasar sistem, cara yang kedua menggunakan nilai RGB dalam hexadecimal, sedangkan yang ketiga menggunakan fungsi. Untuk atribut face, jika *font* yang diberikan tidak tersedia pada sistem, maka akan digunakan *font default* browser.

Fungsi utama tag <font> adalah untuk memanipulasi properti huruf yang ditampilkan. Namun, penulisan seperti ini kurang praktis dan menyebabkan kode HTML sulit dibaca. Cara lain yang lebih baik digunakan ialah dengan memanfaatkan *style sheet* seperti CSS.

3.1.3.2. List

```
<h2><u>Referensi Materi Training</u></h2>
<ul>
<li>HTML</li>
<ol>
<li><i>Web Design in A Nutshell</i>,O'reilly, 2000</li>
<li><i>Pemrograman Web</i>, PAPSI ITS, 2005</li>
</ol>
<li>CSS</li>
    <ol>
<li><i>CSS Web Design for Dummies</i>, Wiley Publishing,
    2005</li>
<li><i>Tutorial CSS </i>,http://www.ilmukomputer.com</li>
    </ol>
</ul>
```

Tampilan pada browser:

Referensi Materi Training

- HTML
  1. *Web Design in A Nutshell*,O'reilly, 2000
  2. *Pemrograman Web*, PAPSI ITS, 2005
- CSS
  1. *CSS Web Design for Dummies*, Wiley Publishing, 2005
  2. *Tutorial CSS*, http://www.ilmukomputer.com

Membuat Definition List

```
<dl>
      <dt>HTML</dt>
      <dd>format standard untuk membuat halaman web</dd>
      <dt>CSS</dt>
      <dd>standard pemakaian <i>style</i> untuk dokumen
          Terstruktur</dd>
</dl>
```

Tampilan pada browser:

HTML
format standard untuk membuat halaman web
CSS
standard pemakaian *style* untuk dokumen Terstruktur

FONT, adapun fungsi utama dari tag <font> adalah untuk memanipulasi properti huruf-huruf yang akan ditampilkan.

3.1.3.3. Menampilkan Teks Yang Sudah Diformat

Tag <pre> berguna untuk menampilkan teks yang kita format sebelumnya. Misalnya, untuk menampilkan indentasi dan karakter-karakter khusus.

```
<pre>
      "Hello, world!"
            "I am learning HTML :)"
</pre>
```

Tampilan pada browser:

```
"Hello, world!"
      "I am learning HTML :)"
```

Terlihat teks ditampilkan *browser* sama dengan yang ditulis dalam *file source* HTML.

3.1.4. Membuat *Links*

Salah satu kemampuan file HTML yang penting ialah membuat *hyperlink* yang dapat menghubungkan suatu halaman web dengan halaman web lainnya. Biasanya dengan meng-klik suatu teks atau *image* tertentu kita bisa berpindah ke bagian lain dari suatu halaman atau ke halaman lain.

Tag yang digunakan untuk membuat link adalah tag *anchor* <a> . Contoh penggunaannya adalah sebagai berikut:

```
<a href="http://www.webdesignguide.com">E-Commerce Pemograman
    Web</a>
```

Jika kita ingin menyimpan *link* dalam gambar tambahkan tag <img>:

```
<a href="http://www.webdesignguide.com"><img src="link.gif"></a>
```

Atribut href diisi oleh URL web yang kita tuju. Tag img tidak membutuhkan penutup, disebut juga *inline style*.

Selain pindah ke halaman lain, *link* juga bisa diarahkan ke bagian lain dalam file. Caranya, *link* mengarah ke suatu bagian yang namanya telah kita definisikan.

```
<a name="awal">Bagian Awal Teks</a>
<a href="#awal">Pindah ke bagian awal</a>
```

Jika kita meng-klik kalimat "Pindah ke bagian awal" maka browser akan menampilkan awal dokumen, tepatnya ke tempat kalimat "Bagian awal teks".

3.1.4.1. Menyisipkan Sejumlah *Link* Pada Satu Gambar

Kadang seorang desainer web menginginkan suatu mekanisme sehingga jika *user* meng-klik suatu bagian dari gambar maka ia akan dibawa ke suatu halaman dan jika ia meng-klik bagian lain dari gambar, ia akan dibawa ke halaman yang lain. Salah satu alternatif pemecahannya ialah dengan memotong-motong gambar, kemudian menyusunnya kembali dalam tabel untuk ditampilkan bersama-sama pada browser. Hal ini memungkinkan beberapa *link* dalam "satu" gambar. Selain itu ada cara lain, yaitu dengan memanfaatkan fungsi bawaan HTML. Perhatikan contoh berikut:

```
<img src="link.gif" usemap="#map">
<map name map>
      <area shape="circle" coords="387,134,12"
       href="anything.html">
      <area shape="default" nohref>
</map>
```

(Sumber gambar https://www.google.com)

LINK merupakan suatu kata, sederetan kata (kalimat) atau gambar dalam suatu web yang tehubung dengan suatu halaman web, baik itu halaman web dari dalam web itu sendiri atau pun halaman web pihak lain.

Penggunaan link pada suatu blog atau situs sangat berguna untuk membuat suatu navigasi yang dibutuhkan untuk membuat suatu penjelasan terhadap suatu objek. Karakteristik setiap link selalu mempunyai sumber dan arah, sumber merupakan link itu sendiri, baik berupa halaman teks atau gambar (teks yang dijadikan link biasa disebut dengan anchor teks), sedangkan arah adalah tujuan dari link (anchor teks) baik itu halaman blog atau website dari dalam blog itu sendiri ataupun halaman blog atau website pihak lain.

(Sumber: http://chip.co.id/)

usemap diisi oleh nama map yang kita gunakan. Area shape bisa diisi oleh "default", "circle", "rect", dan "poly". Pilihan bentuk akan mempengaruhi pengisian koordinat. Pada contoh di atas, karena menggunakan bentuk "circle", maka dibutuhkan tiga data: x, y, dan r. Jika menggunakan "rect", dibutuhkan 4 data, dan jika menggunakan "poly", jumlahnya kita yang menentukan.

3.1.5. Membuat Tabel

Tabel sangat berguna dalam penyajian suatu informasi. Data-data seperti data mahasiswa, sangat baik jika ditampilkan dalam bentuk tabel. Selain itu, tabel juga bermanfaat untuk membuat *layout* halaman web. Tag-tag yang digunakan untuk menampilkan tabel antara lain:

TAG	FUNGSI
<table> </table>	Mendefinisikan tabel dalam HTML
<caption> </caption>	Memberi nama pada tabel
<tr> </tr>	Mendeskripsikan sebuah baris dalam tabel
<td> </td>	Mendeskripsikan sebuah kolom dalam tabel
<th> </th>	Mendeskripsikan judul kolom tabel (teks **bold**)

Cara membuat tabel menjadi lebih populer dengan munculnya beberapa tag tabel baru dalam HTML dan elemen lain yang tersedia di HTML. Pada kesempatan ini akan ditunjukkan cara format tabel menggunakan tag tabel dan elemen HTML. berbagai tag tabel tersebut dan bagaimana untuk format elemen-elemen tersebut menggunakan CSS *(style sheet)*.

Contoh penggunaan tabel untuk penyajian data ialah sebagai berikut:

```
<table border="1">
<caption>Tabel Data Mahasiswa</caption>
<tr>
      <th>No</th>
      <th>Nama</th>
      <th>NRP</th>
      <th>Alamat</th>
</tr>
<tr>
      <td>1</td> <td>Rudi</td>
      <td>1202100041</td>
      <td>Cisitu</td>
</tr>

<tr>
      <td>2</td>
      <td>Sita</td>
      <td>1202100022</td>
      <td>Tubagus Ismail</td>
</tr>
</table>
```

Tampilan pada browser:

Tabel Data Mahasiswa

No	Nama	NRP	Alamat
1	Rudi	1202100041	Cisitu
2	Sita	1202100022	Tubagus Ismail

Untuk membuat layout halaman:

```
<table width="600" height="400">
<tr height="40" align="center">
      <td width="20%">Header Kiri</td>
      <td width="60%">Header Tengah</td>
      <td>Header Kanan</td>
</tr>
<tr align="center">
      <td rowspan="2">Menu Kiri</td>
      <td colspan="2">Menu Isi</td>
</tr>
<tr height="40" align="center">
      <td>Footer Kiri</td>
      <td colspan="2">Footer Kanan</td>
</tr>
</table>
```

Tampilan pada browser

Header Kiri	Header Tengah	Header Kanan
Menu Kiri	Menu Isi	
	Footer Kiri	Footer Kanan

SEJARAH DICIPTAKANYA PASSWORD

Pada awal tahun 1960-an, Fernando Corbato mengubah dunia. Ia mengembangkan sistem password komputer pertama untuk sebuah proyek di Massachusetts Institute of Technology (MIT), dimana beliau mencoba untuk menciptakan akun terpisah di antara para peneliti yang ada di sana.

Dari situlah terciptanya sebuah password pada komputer. Perubahan ini tak hanya memberikan pemahaman tentang bagaimana kita menggunakan teknologi, tetapi juga memaksa kita untuk memikirkan tentang privasi ketika menggunakannya. Tapi sekarang, di era Internet, mantan profesor MIT ini percaya bahwa password telah menjadi mimpi buruk yang nyata.

(Sumber: http://iputan6.com/aribicara.com)

3.1.6. Membuat *Frame*

Frame HTML dapat digunakan untuk membuat tampilan halaman HTML yang terbagi-bagi menjadi beberapa bagian, dimana setiap bagiannya merupakan satu halaman HTML terpisah. Ini dilakukan untuk membuat tampilan halaman salah satu atau beberapa bagian berganti-ganti isinya sedangkan bagian lainnya tetap sehingga dapat menghemat *bandwidth* Internet dan mempercepat proses *download* secara keseluruhan.

Namun, ada hal penting yang perlu diperhatikan. Tidak semua browser mendukung frame, sehingga diperlukan perlakuan khusus terhadap pengguna browser tersebut. Skema dasar frame:

```
<html>
      <head>
            <title>Judul</title>
      </head>
      <frameset>
            <frame src="nama file 1">
            <frame src="nama file 2">
            .
            .
            <noframes>
                  <!-- Bagian ini ditampilkan jika browser
                  tidak mendukung frame -->
            </noframes>
      </frameset>
</html>
```

Pengaturan *layout frame* seperti tabel, ada baris dan kolom. Masing-masing baris dan kolom ditentukan besarnya. Tag <frameset> menentukan tata letak, ukuran, dan banyaknya frame yang akan ditampilkan. Dokumen di atas berfungsi untuk mendefinisikan *frameset* sehingga tidak memerlukan isi (tag <*body*> tidak diperlukan). Isi yang akan ditampilkan berasal dari halaman web yang diacu pada masing-masing frame. Contoh penggunaan frame dapat dilihat sebagai berikut:

Tag <frameset> menentukan tata letak, ukuran, dan banyaknya frame yang akan ditampilkan.

```
<html>
      <head>
            <title>Training HTML</title>
      <head>
      <frameset rows="200,*">
            <frameset cols="400,*">
                  <frame src="list.html" frameborder="1">
                  <frame src="heading.html" frameborder="1">
            </frameset>
            <frame src="font.html">
            <noframes>
                  Bagian ini ditampilkan jika browser tidak
                  mendukung frame
            </noframes>
      </frameset>
</html>
```

3.1.6.1. *Inline Frame*

Di dalam sebuah dokumen (induk) dapat ditampilkan satu atau beberapa dokumen lain (anak). Setiap dokumen anak yang disisipkan diletakkan dalam sebuah *iframe*. Skema dasar dokumen dengan *iframe*:

```
<html>
      <head>
            <title>Training HTML</title>
      <head>
      <body>
            . <!-- isi dokumen -->
            .
            <iframe src="nama file">
                  Bagian ini ditampilkan jika browser tidak
                  mendukung frame
            </iframe>
            . <!-- isi dokumen -->
            .
      </body>
</html>
```

3.1.7. Membuat *Form*

Form HTML merupakan tag yang paling penting dalam pembuatan aplikasi berbasis web. Form menyediakan *interface* untuk menerima masukan data dari *user* sehingga halaman web dapat menjadi interaktif. Contoh aplikasi form antara lain pengisian *guestbook*, mengisi polling, mengisi angket, mengisi data pribadi, dan form pembelian barang. Selain untuk menerima data user, form juga dapat berfungsi untuk navigasi atau kontrol dalam website seperti pemilihan bahasa, layout, dll. Untuk membuat form, tag yang digunakan adalah <form> yang memiliki beberapa atribut sebagai berikut:

```
<form method="POST"(/"GET") action="url"> . . </form>
```

ATRIBUT	FUNGSI
method	menentukan bagaimana informasi akan dikirim. Terdapat 2 pilihan: "POST" dan "GET". POST artinya sebundel data akan dikirim begitu saja, sementara GET akan menambahkan sesuatu pada URL di bagian *address* browser.
action	menentukan ke mana data akan dikirim. Kita dapat mengisinya dengan URL tempat pemroses data berada atau data dikirimkan ke sebuah alamat email, yaitu dengan menuliskan "mailto : *alamat email*"

3.1.8. Membuat Text Box

Untuk membuat sebuah text box digunakan tag sebagai berikut:

```
<input type="text" name="name" size="size maxlength="length"
    readonly>
```

PASSWORD atau kata sandi (inggris: password/passphrase) adalah kumpulan karakter atau string yang digunakan oleh pengguna jaringan atau sebuah sistem operasi yang mendukung banyak pengguna *(multiuser)* untuk memverifikasi identitas dirinya kepada sistem keamanan yang dimiliki oleh jaringan atau sistem tersebut.

(sumber: Imushoru - WordPress.com)

Password merupakan suatu bentuk dari data otentikasi rahasia yang digunakan untuk mengontrol akses ke dalam suatu sumber informasi. Password akan dirahasiakan dari mereka yang tidak diijinkan untuk mengakses, dan mereka yang ingin mengetahui akses tersebut akan diuji apakah layak atau tidak untuk memperolehnya. Password bukan sekedar suatu rangkaian atau bentuk kata-kata; juga bukan suatu untaian kata yang memiliki arti akan lebih sulit untuk ditebak.

Secara fungsional password sering digunakan untuk menggambarkan sesuatu yang lebih tepat disebut pass phrase. Password kadang-kadang digunakan juga dalam suatu bentuk yang hanya berisi angka (numeric); aplikasi password dapat kita jumpai pada Personal Identification Number (PIN). Umumnya memiliki karakteristik relatif pendek sehingga mudah untuk diingat.

(sumber: https://id.wikipedia.org/wiki/kata_sandi)

ATRIBUT	FUNGSI
type="text"	menentukan bahwa elemen form yang dibuat berupa text box
name	nama dari text box, biasanya digunakan untuk *event handler* bahasa script atau pemberian *style*
size	menentukan ukuran dari text box
maxlength	menentukan panjang maksimal masukan
readonly	jika disebutkan maka elemen tidak bisa diubah nilainya

Contoh:

```
<input type="text" name="nama" size="20">
```

Tampilan pada browser: Nama :

3.1.9. Membuat *Password*

Elemen form ini digunakan untuk memperoleh *password* dari *user*. Karakter yang ditampilkan saat *user* mengetikkan *password*-nya adalah *asterisk* (*) atay *bullet* (•). Sintaksnya:

```
<input type="password" name="name" size="size">
```

Contoh :

```
<input type="password" name="password" size="10">
```

Tampilan pada browser:

3.1.10. Membuat *Text area*

Text area digunakan untuk menerima masukan teks yang panjang, yang terdiri dari beberapa baris. Sintaksnya:

ATRIBUT	FUNGSI
rows	menentukan jumlah baris
cols	menentukan jumlah kolom
wrap	ada 3 pilihan,
	"off" -> tulisan akan terus berlanjut ke samping tanpa pindah baris
	"soft" -> data terpotong saat menyentuh batas kanan dan berpindah ke baris baru. Data masih utuh memanjang pada saat dikirimkan
	"hard" -> tulisan terpotong begitu saja ketika menyentuh batas kanan. Data terkirim dalam keadaan terpotong-potong

contoh : Alamat :

```
<textarea name="alamat" rows="3" cols="20" wrap="soft">
            Isikan alamat lengkap Anda </textarea>
```

Tampilan pada browser:

3.1.11. Membuat Radio

Digunakan untuk menerima pilihan kepada *user*. *User* hanya bisa memilih satu pilihan saja.

```
<input type="radio" name="name" value="value">
```

value : memberikan nilai yang isinya ditentukan oleh kita

Contoh: Jenis kelamin :

```
<input type="radio" name="sex" value="m">Female <br />
      <input type="radio" name="sex" value="f"> Male
```

Tampilan pada browser:

RADIO BUTTON adalah masukan untuk memilih salah satu opsi diantara beberapa pilihan. Input radio button sekilas nampak seperti kumpulan beberapa input *checkbox* namun pada input radio button kita hanya diijinkan untuk memilih satu diantara beberapa pilihan. Saat kita memilih satu opsi (mencentang) dalam satu input radio button otomatis opsi yang lain akan tidak dipilih {otomatis tidak tercentang}.

Untuk membuat radio button menggunakan tag <input> dengan atribut type="radio". Seperti yang iainnya, input radio button juga memiliki dua atribut yang dipakai untuk pemrograman, yaitu name dan value. Atribut name digunakan untuk mendefinisian identitas saat pengambilan data dari input radio button. Dan satu tagi adafah atribut value yang merupakan nilai dari input radio button.

Saat membuat *input radio button* kita harus membuat nama yang sama pada masing-masing *input*. Yang berbeda dari masing-masing input radio button adalah *value*-nya. Jumlah opsi minimal pada input radio button adalah dua dan jumlah maksimalnya belum saya ketahui karena sejauh ini belum pernah mengalami kekurangan opsi saat membuat input *radio button.*

Untuk mendefinisi tiap-tiap input radio button, kita bisa menambahkan keterangan label setelah input *radio button.* Dan sebaiknya salah satu kondisl input *radio button* sudah dicentang *{checked}* saat pertama kali *load* agar nantinya tidak mengembalikan nilai kosong. Agar *input radio button* dalam kondisl dicentang saat *load* bisa digunakan atribut *checked.*

3.1.12. Membuat *Checkbox*

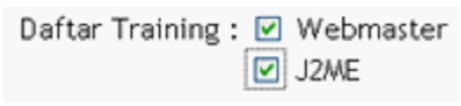
Digunakan untuk menerima pilihan dari *user*. Dapat memilih lebih dari satu.

```
<input type="checkbox" name="name" value="value">
```

Contoh:

```
      Daftar Training:

      <input type="checkbox" name="training" value="web>
         Webmaster <br />
 <input type="checkbox" name="training" value="j2me">J2ME
```

Tampilan pada browser:



CHECK BOX adalah salah satu komponen form yang difungsikan agar pengguna dapat memilih menu yang tersedia dengan cara mengklik salah satu menu yang tersedia.

Untuk checkbox dan radio, kita bisa tentukan pilihan *default* dengan menambahkan atribut checked pada tag pilihan.

3.1.13. Membuat Select

Select berbeda dengan radio maupun checkbox karena tidak menggunakan tag <input>, tapi langsung memakai tag <select>. Elemen select berupa list *drop down* terdiri dari sejumlah pilihan.

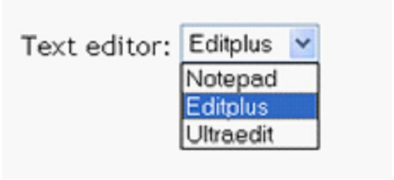
Atribut	Fungsi
multiple	berguna agar dapat menentukan pilihan lebih dari satu, yaitu dengan menekan tombol *ctrl* dan meng-klik pilihan tersebut
size	banyaknya baris pilihan ditampilkan, *defaul*: 1
selected	memberikan pilihan awal

Perintah **SELECT**, merupakan perintah untuk menampilkan data dari satu tabel atau lebih, meski demikian penggunaan perintah SELECT bisa dipotong seperlunya tanpa meninggalkan bagian yang terpenting yaitu perintah SELECT dan FORM

Contoh :

```
Text editor:
<select name="Editor" size="1">
      <option selected>Notepad<br />
      <option>Editplus<br />
      <option>Ultraedit<br />
</select>
```

Tampilan pada browser:



3.1.14. Membuat *Button, Reset, Submit,* dan *Image*

Semua komponen input ini merupakan komponen input yang berbentuk *button* yang akan bereaksi ketika terjadi *event* (misalnya penekanan tombol). Ada beberapa perbedaan dari keempat tombol ini yang terkait dengan fungsi maupun bentuk tampilan *button*.

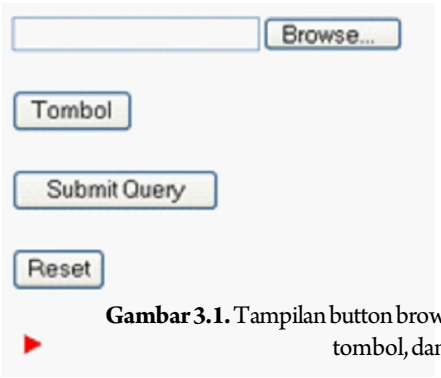
TAG BUTTON yang digunakan untuk membuat tombol di dalam HTML ini bisa digunakan di dalam form, maupun di luar form.

Atribut	Fungsi
button	akan menampilkan sebuah tombol pada browser yang tidak akan bereaksi apapun sebelum diberi kode tambahan
reset	komponen ini memiliki bentuk sama dengan tombol dan memiliki fungsi khusus yaitu menghapus semua masukan *user* pada elemen-elemen form
submit	tombol yang jika diklik, isian di form akan langsung dikirim menggunakan ke file pemroses yang disebutkan dalam atribut action
image	sama dengan submit, namun tampilannya bisa diubah dengan gambar

```
<input type="button/reset/submit/image" name="name"
        value="value" src="file gambar">
```

Contoh:

```
<form>
   <input type="file" value="Masukkan file"><br /><br />
   <input type="button" value="Tombol"><br /><br />
   <input type="submit"><br /><br />
   <input type="reset"><br /><br />
   <input type="image" src="a1.gif"><br />
</form>
```



Gambar 3.1. Tampilan button browse, submit query, tombol, dan reset

Tag button juga tidak harus berada di dalam form, dan ini... memberikan fleksibilitas yang tinggi. Anda bisa memprogramnya menggunakan javascript untuk membuat fungsi lain yang akan dijalankan ketika tombol tersebut di *klik*.

3.1.15. Membuat Hidden

Komponen input hidden memiliki fungsi untuk mengirim data melalui form yang bukan hasil isian pengunjung. Jadi pembuat situs sudah mendefinisikan data yang akan dikirim. Hal ini berguna antara lain untuk pengolahan database menggunakan CGI script.

```
<input type="hidden" name="name" value="value">
```

Fungsi tag input type *hidden* dalam pembuatan Form HTML

Tag input dengan type hidden adalah sebuah objek form khusus yang sesuai dengan nama tipenya, memang tersembunyi dan tidak dapat dilihat oleh user. Namun objek form ini menawarkan fleksibilitas yang dapat dimanfaatkan oleh programmer untuk menyisipkan suatu data untuk diproses.

Type hidden ini bisa digunakan untuk mengirim suatu data dari sebuah halaman ke halaman lain tanpa mengganggu bentuk form yang telah ada, dan biasanya digabungkan dengan pemrosesan javascript atau PHP.

3.1.16. Membuat File

Elemen ini berfungsi untuk mengirimkan file

```
<input type="file" name="name">
```

Tampilan pada browser:

Gambar 3.2. Tampilan button browse

3.1.17. Informasi Dokumen

Informasi mengenai sebuah dokumen HTML disimpan di bagian *head* dengan menggunakan tag <meta>.

Informasi	Contoh
Nama pembuat isi dokumen	<meta name="author" content=”Jmarsha”> <meta name="copyright" content="©2012 E-commerce Pemograman Web">
Deskripsi isi dokumen	<meta name="description" content="Berisi materi training HTML dan CSS” > <meta name="keywords" content="CSS, HTML,inheritance">
Tanggal dokumen	<meta name="date" content="2004-10-09 08:49:37+07">
Usia dokumen	<meta http-equiv="Expires" content="Tue, 20 Sep 2005 14:25:27 GMT"> <meta http-equiv="Refresh" content="10"> <meta http-equiv="Refresh" content="300; url=http://www.myhomepage.com/home.html">

3.1.18. Pengelompokan Elemen Dokumen

Pengertian meta tag

META TAG tidak dimaksudkan untuk menambah atau mempengaruhi isi dari konten halaman web kita, namun lebih kepada keterangan tambahan tentang halaman itu sendiri. Meta tag biasanya di tempatkan pada bagian head dari HTML, dan biasanya terdiri dari beberapa meta tag dengan atribut yang berbeda-beda. Beberapa atribut yang biasanya di tampilkan adalah charset, name, http-equiv, dan content.

Beberapa elemen dokumen (teks, paragraf, gambar, tabel, dan lain-lain) dapat dikelompokkan untuk menambahkan struktur lojik pada dokumen. Ada 2 jenis pengelompokan:

- inline (alur, *flow*), biasanya untuk sebaris teks dalam file HTML. Menggunakan tag
- block (blok), untuk sekumpulan elemen. Menggunakan tag </ div>

Di dalam tag dan biasanya menggunakan atribut class atau id untuk menentukan kelas atau identitas sekumpulan elemen tersebut. Penggunaan struktur seperti ini sangat berguna untuk mempermudah menginterpretasi isi dokumen dan memberikan style dengan CSS.

3.2. *Cascading Style Sheet* (CSS)

3.2.1. Pengertian *Cascading Style Sheet*(CSS)

CSS (*Cascading Style Sheet*) merupakan sebuah standard penggunaan dan pembuatan style pada suatu dokumen halaman web. CSS direkomendasikan oleh W3C pada tahun 1996 sebagai akibat munculnya berbagai *presentation HTML tags* (tag-tag HTML yang berfungsi mengatur tampilan informasi dalam webpage) yang ‘mencemari’ HTML sebagai bahasa struktur.

Rekomendasi awal CSS oleh W3C pada tahun 1996 kemudian disebut CSS level 1 (CSS1). Kemudian, W3C’s Cascading Style Sheet and Formatting Properties (CSS & FP) Working Group membuat spesifikasi CSS lanjut, sehingga pada tahun 1998 CSS level 2 (CSS2) telah menjadi rekomendasi penuh. CSS2 dibuat berdasarkan CSS1 dengan perluasan tanpa mengubah bagian-bagian utama dari CSS1.

Kini, CSS3 pun sudah dikembangkan namun belum secara formal. Untuk melihat rekomendasi CSS1 atau CSS2, kunjungi http://www.w3.org/TR/REC-CSS1 atau http://www.w3.org/TR/REC-CSS2.

Penggunaan CSS sangat baik untuk memisahkan style dengan format tampilan informasi dalam dokumen HTML. Contohnya, untuk mengatur style pada font, kita tidak perlu menggunakan tag <font> lagi. Cukup dengan beberapa baris perintah saja, kita dapat mendefinisikan beberapa style untuk teks.

Keuntungan keuntungan kita menggunakan CSS antara lain:

1. *rich styling,* CSS menawarkan lebih banyak alternatif *style* yang bisa ditambahkan pada webpage daripada HTML
2. *ease of use,* CSS dapat mengurangi beban seorang webmaster dalam *coding* dengan meletakkan pendefinisian style di satu tempat daripada tersebar dalam tag-tag di seluruh dokumen.
3. *using style for multiple pages,* informasi style suatu webpage dapat disimpan terpisah dari file HTML dan diaplikasikan ke banyak halaman web sehingga menjaga kesatuan semua halaman web.
4. *cascading,* CSS menangani konflik yang terjadi bila ada sebuah *style rule* yang diaplikasikan pada elemen yang sudah memiliki *style* tertentu. Dengan ini, pembuat web dapat membuat perubahan *style* dengan cepat tanpa menghapus *style* sebelumnya. Hal ini juga memungkinkan *user* membuat *style*-nya sendiri (disebut juga *reader style sheet*).
5. *compact file size,* setiap halaman web dapat berkurang ukuran filenya karena informasi mengenai *style* disimpan di tempat lain. Hal ini juga mempercepat proses download.

CSS (Cascading Style Sheets) berisi rangkaian instruksi yang menentukan bagiamana suatu text akan tertampil di halaman web. Perancangan desain text dapat dilakukan dengan mendefinisikan Elemen-elemen seperti *fonts*(huruf) , *colors*(warna), *margins*(ukuran), latar belakang *(background)*, ukuran font *(font sizes)*dan lain-lain.

Cascading Style Sheets juga bisa berarti meletakkan styles yang berbeda pada *layers* (lapisan) yang berbeda. CSS terdiri dari style sheet yang memberitahukan browser bagaimana suatu dokumen akan disajikan. Fitur-fitur baru pada halaman web lama dapat ditambahkan dengan bantuan style sheet.

Saat menggunakan CSS, Anda tidak perlu menulis font, color atau size pada setiap paragraf, atau pada setiap dokumen. Setelah Anda membuat sebuah style sheet, Anda dapat menyimpan kode tersebut sekali saja dan dapat kembali menggunakannya bila diperlukan.

3.2.2. CSS dan HTML

Ada beberapa cara yang dapat digunakan untuk melekatkan (*embedding*) CSS ke dalam file HTML.

3.2.2.1. Menggunakan *tag <link>*

```
<link rel="stylesheet" type="text/css" href="file.css">
```

Tag ini berguna untuk mengimpor *sytle* dari file eksternal ke dalam file HTML. Agar file tersebut berhasil di-*load*, tag ini sebaiknya diletakkan di antara tag <head> </head>.

3.2.2.2. Menggunakan *tag < style >*

```
<style type="text/css">
            . /* pendefinisian style rule */
            . /* tanda ini berarti komentar */
</style>
```

fungsi utama CSS adalah merancang, merubah, mendisain, membentuk halaman wesite(blog juga website). dan isi dari halaman website adalah tag-tag html, logikanya css itu dapat merubah tag-tag html(yang sederhana) sehingga menjadi lebih fungsional dan menarik.

CSS *(Cascading Style Sheet)* adalah salah satu bahasa desain web *(style sheet language)* yang mengontrol format tampilan sebuah halaman web yang ditulis dengan menggunakan penanda(markup laguage. Biasanya CSS digunakan untuk mendesain sebuah halaman HTML dan XHTML, tetapi sekarang CSS bisa diaplikasikan untuk segala dokumen XML, termasuk SVG dan XUL bahkan ANDROID.

Tag ini digunakan untuk pendefinisian *style* yang berada dalam satu file dengan file HTML. Tag ini juga diletakkan di antara tag <head>

3.2.2.3. Perintah *@import*

Fungsinya sama dengan tag <link> dan diletakkan di antara tag <style>. Tidak semua browser mendukung perintah ini. Browser browser lama mungkin saja malah akan menampilkan *style rule* kita. Oleh karena itu, salah satu penyelesaiannya ialah lingkupi *style rule* kita dengan tag komentar (<!—— >).

```
<style type="text/css">
@import url(file.css);
. /* definisi style rule */
.
</style>
```

3.2.2.4. *Inline Style*

```
<tag style="property1: value; property2: value; ..."> text </tag>
```

Penulisan *style rule* langsung pada tag tertentu. Cara ini dapat diterapkan untuk semua tag di dalam tag <body>.

3.2.3. CSS *Rules*

CARA KERJA CSS sangatlah mudah, kita hanya perlu menulis *style*-nya(*selector id* dan *class* tersesuaikan) maka secara otomatis akan bekerja pada *document* HTML.

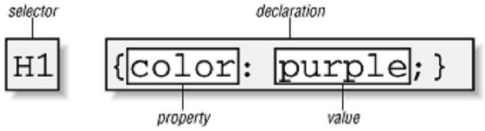
CSS didefinisikan dalam bentuk *rule*. Ada tiga bagian utama dari sebuah rule:

1. *selector,* berfungsi memberi tahu browser pada tag HTML (elemen) mana rule CSS akan diaplikasikan,
2. *property,* nama dari sifat (*quality*) dari sebuah elemen (serupa dengan atribut dalam tag HTML),
3. *value,* data aktual yang menentukan perubahan (penampilan) elemen pada halaman web.

Properti dan value dituliskan di dalam kurung kurawal, membentuk suatu kesatuan yang disebut *declaration*. Skema pendefinisian rule CSS ialah sebagai berikut:

```
selector {property: value}
```

contohnya:



Gambar 3.3. Skema pendefinisian *rule* CSS

Sama seperti dalam HTML, browser akan memperlakukan *multiple space* dan *carriage return* sebagai suatu *single space*. Sehingga penulisan rule CSS dapat dibuat berindentasi agar lebih mudah dimengerti. Untuk menyisipkan komentar, CSS menggunakan penanda /* dan */. Teks dilingkupinya tidak akan diinterpretasi.

3.2.4 Selector

Ada tiga buah selector yang sering digunakan dalam pendefinisian *style* dengan CSS, adapun ketiga selector tersebut diantaranya adalah:

3.2.4.1. Tag HTML

Semua tag HTML bisa digunakan sebagai selector. Style yang didefinisikan dengan CSS akan berasosiasi dengan elemen HTML yang bersangkutan. Contoh:

```
p {text-decoration: overline; font-color: red}
```

3.2.4.2. Selektor *class*

Untuk menggunakan selektor ini, elemen elemen di dalam *web* harus diberi kelas tertentu dengan memanfaatkan atribut *class*. Contohnya:

```
<p class="paragraf1"> teks </p>
```

Dalam CSS *rules*, selektor kelas didahului oleh titik

```
.paragraf1 {font-color: navy; font-family: Arial}

        p.paragraf1 {font-style: italic}
```

Rule yang pertama berarti setiap elemen yang termasuk kelas "paragraf1" akan menggunakan font Arial berwarna navy. Sedangkan yang kedua, hanya berlaku untuk kelas "paragraf1" yang merupakan elemen p.

3.2.4.3. Selektor *id*

Penggunaan selektor id mirip dengan class. Perbedaannya ialah, dalam sebuah dokumen HTML tidak boleh ada lebih dari 1 elemen yang memiliki id value yang sama. Jadi, sebuah id harus unik untuk satu elemen tertentu dalam suatu dokumen HTML, sedangkan class dapat digunakan untuk lebih dari 1 elemen, yaitu beberapa elemen dapat memiliki *class* yang sama. Analoginya seperti nama seseorang (id) dengan kelas kuliahnya *(class)*. Pendefinisiannya menggunakan atribut id:

```
<h1 id="title">CSS</h1>
```

Dalam CSS rule:

```
#title {font-weight: bold; text-decoration: underline}
```

3.2.4.4. *Contextual Selector*

Selektor kontekstual mirip seperti ekspresi kondisional (if...then...) dalam pemrograman. Yaitu, deklarasi akan dievaluasi jika kondisi tertentu dipenuhi. Contohnya, rule

CSS memiliki jenis *selector* yang bervariasi, bahkan sangat beragam tergantung kebutuhan kita untuk mendesain halaman web. *Selector* adalah sebuah pola *(pattern)* yang digunakan untuk 'mencari' suatu tag di dalam HTML. Analogi untuk *selector*, contoh: mencari semua tag p, atau mencari seluruh tag h1 yang memiliki atribut *class*=judul.

CSS dibuat untuk memisahkan konten utamadengan tampilan dokumen yang meliputi layout, warna da font. Pemisahan ini dapat meningkatkann daya akses konten pada web, menyediakan lebih banyak fleksibilitas dan kontrol dalam spesifikasi darisebuah karakteristik dari sebuah tampilan, memungkinkan untuk membagi halaman untuk sebuah formatting dan mengurangi kerumitan dalam penulisan kode dan struktur dari konten, contohnya teknik tableless pada desain web.

CSS juga memungkinkan sebuah halaman untuk ditampilkan dalam berbagai style dengan menggunakan metode pembawaan yang berbeda pula, seperti on-screen, in-print, by voice, dan lain-lain. Sementaraitu, pemilik konten web bisa menentukan link yang menghubungkan konten dengan file CSS.

TUJUAN UTAMA CSS DICIPTAKAN untuk membedakan konten dari dokumen dan dari tampilan dokumen, dengan itu, pembuatan ataupun pemrograman ulang web akan lebih mudah dilakukan. Hal yang termasuk dalam desain web diantaranya adalah warna, ukura dan formatting. Dengan adanya CSS, konten dan desain web akan mudah dibedakan, jadi memungkinkan untuk melakukan pengulangan pada tampilan-tampilan tertentu dalam suatu web, sehingga akan memudahkan dalam membuat halaman web yang banyak, yang pada akhirnya dapat memangkas waktu pembuatan web.

```
h1 i {font-family: "Trebuchet MS"}
```

hanya akan dievaluasi untuk elemen yang dilingkupi tag <h1> dan <i>.

3.2.4.5. *Pseudo-Class*

Pseudo-class dapat digunakan sebagai selektor meskipun *class*-nya tidak terdapat di kode HTML. *Pseudo-class* digunakan jika suatu kondisi eksternal dikenakan pada elemen HTML (misalnya sebuah link di-klik). Sintaksnya:

```
selector:pseudo-class {property: value}
```

Pseudo-class	Keterangan
link	Mengacu pada sebuah link yang belum dikunjungi
visited	Mengacu pada sebuah link yang sudah dikunjungi
hover	Mengacu pada sebuah link saat pointer berada di atasnya (*rollover*)
active	Mengacu pada sebuah link yang sedang aktif (di-klik)

3.2.4.6. Pseudo-Element

Digunakan untuk menyatakan style yang diaplikasikan pada suatu kondisi tipografi, bukan struktur. Misalnya huruf pertama, baris pertama, dsb. Contoh:

```
h1:first-letter {font-size: 150%; color: blue}
p:first-line {font-weight: bold}
```

Di CSS1, pseudo-class dan pseudo-element tidak bisa digabungkan dengan selektor lain, seperti id atau class. Namun, hal ini dimungkinkan oleh CSS2.

ID SELECTOR digunakan untuk menentukan style bagian unik (baca: nggak ada duanya) di HTML, misalkan bagian header, bagian artikel, bagian sidebar, atau bagian footer. Id Selector hanya bisa digunakan sekali dalam sebuah dokumen HTML, jika lebih dari 1 maka yang akan di proses hanya yang pertama saja.

3.2.5. Grouping

CSS memudahkan pendefinisian rule dengan konsep grouping. Ada tiga macam grouping:

Pengelompokan	Contoh
Selector	h1, h2, h3 {color: blue}
Property	p {font-family: Verdana; font-size: 12pt}
Value	i {font: bold 12pt "Times New Roman"}

3.2.6. Specificity

Specificity menentukan style mana yang akan digunakan jika terdefinisi dua atau lebih style pada suatu elemen. Misalnya

```
.paragraf1 {color: blue}
p {color: red}
```

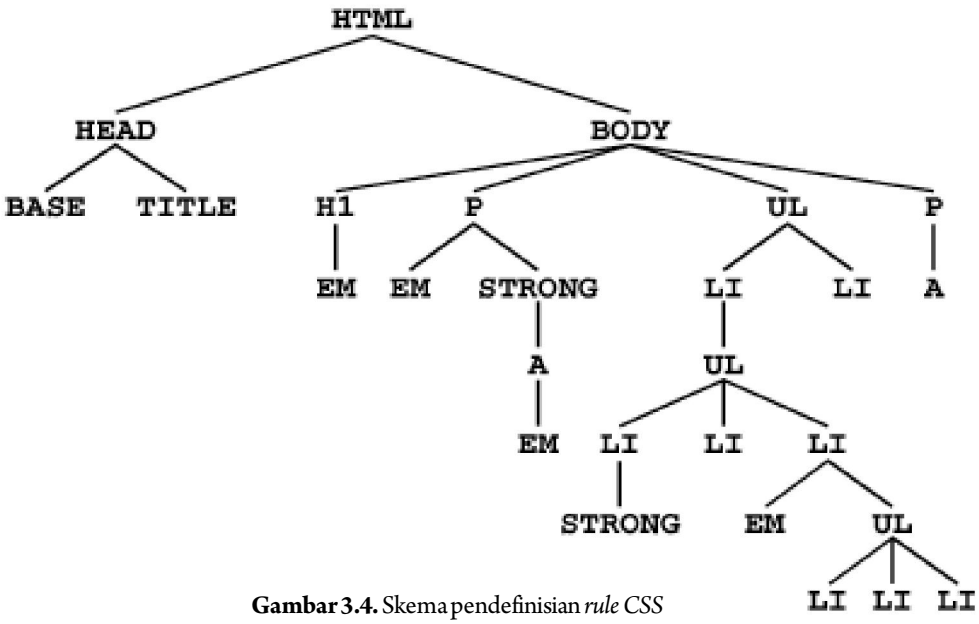
kedua definisi ini mungkin terjadi pada suatu elemen secara bersamaan. Warna apa yang akan muncul?

```
<p class="paragraf1"> teks </p>
```

Jawabannya adalah biru, kenapa? Karena class memiliki specificity lebih tinggi daripada tag <p>. Jadi, specificity adalah bobot relatif dari berbagai rules. Berdasarkan spesifikasi, selektor id memiliki specificity 100, class memiliki specificity 10, dan tag HTML memiliki specificity 1.

3.2.7. *Inheritance*

Tag-tag dalam HTML umumnya memiliki hubungan induk-anak (*parent-child*). Artinya, setiap tag yang melingkupi tag lainnya disebut sebagai induk dari tag yang dilingkupinya tersebut. Gambar di bawah ini memperlihatkan struktur tag-tag HTML dari sudut pandang *parent-child*.



Gambar 3.4. Skema pendefinisian *rule CSS*

Struktur ini penting untuk dipahami berkaitan dengan konsep inheritance dalam CSS. Inheritance berarti suatu style yang didefinisikan pada induk akan terdefinisi pula untuk anaknya. Lihat struktur di atas. Jika kita mendefinisikan tag <body> font-nya berwarna merah dan kita tidak mendefinisikan warna font lain untuk tag <h1>, maka otomatis elemen font elemen h1 akan berwarna merah.

3.2.8. Font

Di bawah ini terdapat beberapa property dan value-nya yang berhubungan dengan pengolahan font

Perlu diketahui, tidak semua property CSS bisa menurunkan efek dari tag induknya, misalnya seperti property border yang hanya berlaku untuk tag induk saja dan tidak berlaku untuk tag yang berada di dalam tag induknya.

PENGERTIAN SIFAT *INHERITANCE* DALAM CSS?

Sifat Inheritance atau bisa juga dibilang sifat turunan (pewarisan). Dimana maksud turunan (pewarisan) ini adalah sifat penurunan efek dari satu sebuah Tag HTML ke Tag HTML lainnya.

Untuk melakukan sifat Inheritance ini tag yang ingin mengelami sifat turunan itu harus berada di dalam tag lainnya. Sehingga jika Tag lainnya itu diberikan efek maka tag yang berada di dalamnya juga akan mengikuti.

Jika di dalam seubah Halaman WEB, biasanya Tag yang berada di dalam Tag <Body> mengalami sifat turunan ini.

Penurunan atau *inheritance* adalah fitur yang ada pada hampir seluruh bahasa pemrogaman termasuk CSS, *Inheritance* pada css akan menurunkan efek dari property CSS yang ditulis pada satu tag HTML ke tag HTML lainnya. Penurunan property css akan berlaku jika terdapat tag didalam tag HTML lainnya

Properties	Values
font-family	Arial, "Times New Roman", Verdana, Helvetica
font-style	normal, italic, oblique
font-variant	normal, small caps
font-weight	normal, bold, light, lighter, 100[..900]
font-size	x-small, small, medium, large, x-large larger, smaller 100px, 50px, ... 150%, 20%, ...

Untuk pengelompokan nilai, urutan pengisiannya adalah sebagai berikut:

```
font: style variant weight size family
```

Contoh

```
h1 {font: italic 14pt normal bold Helvetica}
h2 {font: 12pt bold Verdana}
```

3.2.9. Color

Ada beberapa cara untuk menyatakan *value property color* (sama seperti di HTML).

Contoh

```
p {color: blue}
p {color: #42342C}
p {color: #f25
p {color: rgb(255,4,0)}
p {color: rgb(40%,54%,25%)}
```

Value CSS adalah nilai dari *property*. Misalkan untuk *property background-color* yang digunakan untuk mengubah warna latar belakang dari sebuah *selector, value* atau nilainya dapat berupa *red, blue, black,* atau *white.*

3.2.10. Background

Properties	Values
background-color	*color* (blue, red, dll), transparent
background-image	None, url(http://hmif.if.itb.ac.id/web.gif)
background-repeat	repeat, repeat-x, repeat-y, no-repeat
background-attachment	scroll, fixed
background-position	top, center, bottom,

Untuk pengelompokan nilai, urutan pengisiannya adalah sebagai berikut:

```
background: color image repeat attachment position
```

Contoh

```
body {background: url("image.jpg") blue repeat fixed}
body {background: red none no-repeat top scroll center}
```

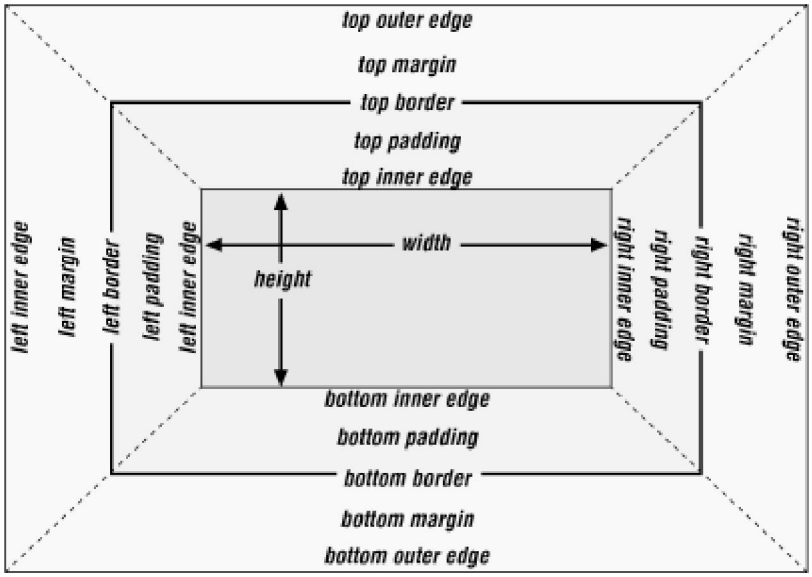
3.2.11. Text

Properties	Values
Text-indent	*length* (5em, 21em, ...)
Text-align	left, right, justify, center
Text-decoration	none underline overline line-trough blink
Text-shadow	*length* (1px, 2px, ...) *length length, color*
letter-spacing	none, *length*
Word-spacing	none, *length*
Line-height	normal, *number, percentage*
Text-transform	capitalize, uppercase, lowercase, none
white-space	normal, pre, nowrap

Ada beberapa property dalam box model CSS diantaranya Border, Padding, Margin dan content HTML itu sendiri. Didalam HTML, semua elemen seperti p, div, h dan lain-lain pada dasarnya dianggap kolom kotak yang membungkus isi (html) di dalam css, istilah "model kotak / box" digunakan ketika berbicara tentang design dan tata letak.

3.2.12. *Element Box*

Elemen box adalah kotak yang dihasilkan setiap elemen dalam dokumen HTML. Kotak ini menggambarkan sejumlah space yang ditempati oleh elemen tersebut. Setiap box dapat memengaruhi posisi dan ukuran elemen lainnya. Semua elemen HTML terdistribusi sedemikian sehingga tidak akan saling bertumpuk. Hal ini dimungkinkan karena adanya pengaturan batas batas elemen. Batas batas tersebut dapat dilihat pada Gambar 3.5 di bawah ini:



Gambar 3.5. *Element box* pada HTML

Pengaturan *margin, border,* dan *padding* dapat dibuat dengan CSS. Biasanya dilakukan dengan bantuan tag .

Masing-masing *property margin, border,* dan *padding* dapat diatur batas atas, bawah, kiri dan kanannya *(top, bottom, left, right)*. *Value*-nya dapat diisi dengan ukuran panjang *(length)* atau presentase (mengacu pada lebar elemen induknya).

Contoh:

```
div {margin-top: 12pt}
div {margin-bottom: 2px}
        div {border: 3em 1em 2em 4em}
        /* grouping, mengatur top, right, bottom, left
        */
        div {padding-left: 12%}
```

Penting untuk diingat, jika anda mengatur lebar dan tinggi menggunakan CSS, anda hanya mengatur lebar dan tinggi dalam *content area.* Untuk menghitung semua elemen, anda harus menambahkan *padding, border* dan *margin.*

Berbagai style untuk properti border dapat dilihat pada tabel berikut:

Properties	Values
border-color	*color* (*red, blue, ...*)
border-style	hidden, double, solid, dashed, dotted, ridge, outset, inset, none
border-[top|right|bottom|left]	*length* (*percentage*)
width	*length*
height	*length*

Tim Berners-Lee | Penemu E-mail (Elektronik Mail)

Sir Tim Berners-Lee Penemu *www*

Nama Lengkap:
Timothy John Berners-Lee

Tempat tanggal lahir:
London-Inggris, 8 Juni 1955

Tempat tinggal: Inggris, Amerika
Kebangsaan: Britania
Kuliah : Oxford
Pekerjaan: Ilmuwan komputer
Tempat kerja:
Konsorsium World Wide Web
Universitas Southampton Plessey
ITM
Hasil penemuan:
Penemu World Wide Web, Pemegang 3Com Pendiri Chair di bidang Ilmu Komputer dan Kecerdasan Buatan Laboratorium ITM, Situs web www.w3.org/People/Berners-Lee

(Sumber: https://id.wikipedia.org/wiki/Tim_Berners-Lee)

Sir Timothy John "Tim" Berners-Lee, KBE (TimBL atau TBL) (lahir di London, Inggris, 8 Juni 1955; umur 61 tahun) adalah penemu World Wide Web dan ketua World Wide Web Consortium, yang mengatur perkembangannya.

Pada 1980, ketika masih seorang kontraktor bebas di CERN, Berners-Lee mengajukan sebuah proyek yang berbasiskan konsep hiperteks (hypertext) untuk memfasilitasi pembagian dan pembaharuan informasi di antara para peneliti. Dengan bantuan dari Robert Cailliau dia menciptakan sistem prototipe bernama Enquire.

Setelah meninggalkan CERN untuk bekerja di John Poole's Image Computer Systems Ltd, dia kembali pada 1984 sebagai seorang rekan peneliti. Dia menggunakan ide yang mirip yang telah dia gunakan pada Enquire untuk menciptakan World Wide Web, di mana dia mendesain dan membangun browser yang pertama (bernama WorldWideWeb dan dikembangkan dalam NeXTSTEP) dan server web pertama yang bernama httpd. Situs web pertama yang dibuat Berners-Lee (dan oleh karena itu ia juga merupakan situs web pertama) beralamat di http://info.cern.ch/ (telah diarsip) dan dimasukkan online untuk pertama kalinya pada 6 Agustus 1991.

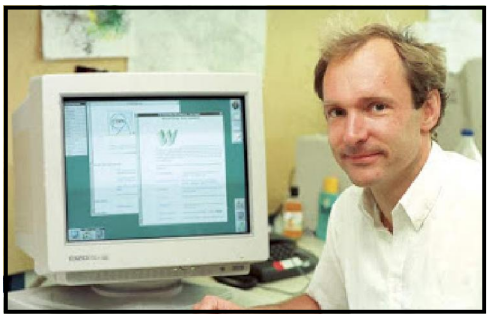
Pada 1994, Berners-Lee mendirikan World Wide Web Consortium (W3C) di Massachusetts Institute of Technology. Hingga kini, Berners-Lee masih tetap rendah hati dan tidak berkeinginan untuk mendapatkan status populer. Banyak yang masih tidak mengetahui kekuatan karya pria ini, World Wide Web. Salah satu kontribusi terbesarnya dalam memajukan World Wide Web adalah dengan tidak mempatenkannya sehingga masih dapat digunakan secara bebas. Pada 16 Juli 2004 dia diberi gelar kehormatan KBE oleh Ratu Elizabeth II sebagai penghormatan atas jasa-jasanya.

Pada tahun 1989, saat bekerja di di CERN, Laboratorium Fisika Partikel Eropa di Jenewa, Swiss, Tim Berners-lee mengusulkan proyek hypertext global, dikenal sebagai World Wide Web. Berdasarkan sebelumnya "Enquire" bekerja, ia dirancang untuk memungkinkan orang untuk bekerja sama dengan menggabungkan pengetahuan mereka dalam web dokumen hypertext. Dia menulis World Wide Web server pertama, "httpd", dan klien pertama, "WorldWideWeb" apa-kau-lihat-ini-apa-kau-mendapatkan peramban hypertext/Editor yang berlari di lingkungan NeXTStep. Pekerjaan ini dimulai pada bulan Oktober 1990, dan program "WorldWideWeb" pertama kali dibuat tersedia dalam CERN pada bulan Desember, dan di Internet pada umumnya pada musim panas 1991.

Melalui 1991 dan 1993, Tim terus bekerja pada desain Web, koordinasi umpan balik dari pengguna di Internet. Spesifikasi awalnya URI, HTTP dan HTML yang disempurnakan dan dibahas di kalangan yang lebih besar seperti penyebaran teknologi Web. Tim Berners-Lee lulus dari Universitas Queen di Oxford University, Inggris, 1976 Sementara di sana ia membangun komputer pertama dengan solder besi, gerbang TTL, prosesor M6800 dan televisi tua. Dia menghabiskan dua tahun dengan Plessey Telecommunications Ltd (Poole, Dorset, Inggris) produsen peralatan utama UK Telecom, bekerja pada sistem terdistribusi transaksi, relay pesan, dan teknologi bar code.

Pada tahun 1978 Tim meninggalkan Plessey untuk bergabung DG Nash Ltd (Ferndown, Dorset, Inggris), di mana ia menulis antara software hal-hal typesetting lainnya untuk printer cerdas, dan sistem operasi multitasking. Setahun setengah dihabiskan sebagai konsultan independen termasuk tugas (Jun-Desember 1980) enam bulan sebagai insinyur perangkat lunak konsultan di CERN. Sementara di sana, ia menulis untuk penggunaan pribadi sendiri Program pertamanya untuk menyimpan informasi termasuk menggunakan asosiasi acak. Dinamakan "Enquire" dan tidak pernah dipublikasikan, program ini membentuk dasar konseptual untuk pengembangan masa depan dari World Wide Web.

Dari tahun 1981 sampai 1984, Tim bekerja di John Poole Citra Computer Systems Ltd, dengan tanggung jawab desain teknis. Bekerja di sini termasuk real time kontrol firmware, grafis dan perangkat lunak komunikasi, dan bahasa makro generik. Pada tahun 1984, ia mengambil sebuah persahabatan di CERN, untuk bekerja pada sistem terdistribusi real-time untuk akuisisi data ilmiah dan sistem kontrol. Antara lain, dia bekerja pada perangkat lunak sistem FASTBUS dan merancang sebuah sistem panggilan prosedur heterogen terpencil.



Pada tahun 1994, Tim mendirikan World Wide Web Consortium di kemudian Laboratorium Ilmu Komputer (LCS) yang bergabung dengan Artificial Intelligence Lab pada tahun 2003 untuk menjadi Ilmu Komputer dan Kecerdasan Buatan Laboratorium (CSAIL) di Massachusetts Institute of Technology (MIT) . Sejak saat itu ia menjabat sebagai Direktur World Wide Web Consortium organisasi standar Web yang mengembangkan teknologi interoperable (spesifikasi, pedoman, perangkat lunak, dan alat-alat) untuk memimpin Web secara maksimal. Konsorsium memiliki situs tuan rumah yang terletak di MIT, di ERCIM di Eropa, dan di Universitas Keio di Jepang serta Kantor di seluruh dunia.

Pada tahun 1999, ia menjadi pemegang pertama dari kursi 3Com Pendiri. Pada tahun 2008 ia diangkat 3COM Pendiri Profesor Teknik di Sekolah Teknik, dengan janji bersama di Departemen Teknik Elektro dan Ilmu Komputer di CSAIL di mana ia juga mengepalai Informasi Desentralisasi Group (DIG). Pada bulan Desember 2004 ia diangkat menjadi Profesor di Departemen Ilmu Komputer di University of Southampton, Inggris. Dia adalah co-Direktur Web Ilmu Trust, diluncurkan pada tahun 2006 sebagai Web Science Research Initiative, untuk membantu menciptakan tubuh penelitian multidisiplin pertama yang meneliti World Wide Web dan menawarkan solusi praktis yang diperlukan untuk membantu memandu penggunaan dan desain masa depan. Dia adalah Direktur World Wide Web Foundation, dimulai pada 2008 untuk mendanai dan mengkoordinasikan upaya-upaya untuk memajukan potensi Web untuk manfaat kemanusiaan.

Pada bulan Juni 2009 lalu Perdana Menteri Gordon Brown mengumumkan bahwa ia akan bekerja sama dengan Pemerintah Inggris untuk membantu membuat data lebih terbuka dan dapat diakses di Web, membangun karya Kekuatan Informasi Task Force. Sir Tim adalah anggota dari The Sektor Publik Transparansi Dewan bertugas untuk mendorong maju agenda transparansi Pemerintah Inggris. Dia telah mempromosikan data pemerintah terbuka secara global, adalah anggota Dewan Transparansi Inggris. Dia adalah Presiden Data Terbuka Institute di Inggris.

Dia adalah penulis, dengan Mark Fischetti, buku "Weaving the Web" pada masa sekarang masa lalu dan masa depan Web. Pada 18 Maret 2013, Tim, bersama dengan Vinton Cerf, Robert Kahn, Louis Pouzin dan Marc Andreesen, dianugerahi Ratu Elizabeth Prize for Engineering untuk "tanah-melanggar inovasi dalam rekayasa yang telah memberikan manfaat global untuk kemanusiaan."

Hasil Karya:

1. Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential Tim Berners-Lee (Foreword), Dieter Fensel (Editor), James Hendler (Editor), Henry Lieberman (Editor), Wolfgang Wahlster (Editor) (The MIT Press, 2005) ISBN 0-262-56212-X
2. Weaving the Web: Origins and Future of the World Wide Web, Tim Berners-Lee (Texere Publishing, 1999) ISBN 0-7528-2090-7
3. The World Wide Web (WWW) is so ubiquitous that it seems strange to think that it has only been around for a few years. Indeed, use of the WWW became widespread in the mid 1990's, but its beginnings can actually be traced back to 1980 when Tim Berners-Lee, an Englishman who had recently graduated from Oxford, landed a temporary contract job as a software consultant at CERN (the famous European Particle physics Laboratory in Geneva). He wrote a program, called Enquire, which he called a "memory substitute," for his personal use to help him remember connections between various people and projects at the lab (Wright, 64). This was a very helpful tool since CERN was (and still is) a large international organization involving a multitude of researchers located around the world. (http://www.ibiblio.org/pioneers/lee.html)

Rangkuman

1. HTML (*HyperText Markup Language*) adalah format standard yang digunakan untuk membuat halaman web. HTML merupakan subset dari bahasa yang lebih luas, yaitu SGML (*Standard Generalized Markup Language*)
2. Sebuah file HTML umumnya terdiri dari 2 bagian utama: *head* dan *body*; keduanya dilingkupi oleh tag
3. Di dalam sebuah dokumen (induk) dapat ditampilkan satu atau beberapa dokumen lain (anak). Setiap dokumen anak yang disisipkan diletakkan dalam sebuah *iframe*
4. Elemen box adalah kotak yang dihasilkan setiap elemen dalam dokumen HTML. Kotak ini menggambarkan sejumlah space yang ditempati oleh elemen tersebut
5. CSS (*Cascading Style Sheet*) merupakan sebuah standard penggunaan dan pembuatan style pada suatu dokumen halaman web

Daftar Istilah

1. DHTML [*Dynamic HTML*]
2. DNS [*Domain Name System* (Layanan)]
3. Nama Domain
4. *Frames*
5. FTP *[File Transfer Protocol]*
6. *Hosting*
7. *Internet*
8. IP *[Internet Protocol]*
9. Tautan *[Hyperlink]*
10. Meta Tag
11. RSS *[Really Simple Syndication]*
12. *Search Engine*
13. *Server*
14. *Spider*
15. SSI [Server-Side Includes]
16. Sub-Domain (Nama)
17. Tag
18. Template
19. Script
20. Scripting Bahasa AJAX *[Asynchronous JavaScript dan XML]*
21. ASP [Active Server Pages]
22. ASP.NET
23. Bandwidth
24. JavaScript
25. URI *[Uniform Resource Identifier]*
26. URL *[Uniform Resource Locator]*
27. *Usability*
28. Validasi
29. W3C *[World Wide Web Consortium]*
30. XHTML *[Extensible HyperText Markup Language]*

Latihan Soal

1. Jelaskan pengertian dibawah ini?
   a. HTML
   b. CSS
   c. Method
   d. Action
   e. Button
   f. Reset

g. Submit
h. Image
i. Selector
j. Property
k. Value
l. Element Box

2. Sebutkan dan jelaskan skema dasar HTML!
3. Sebutkan dan jelaskan beberapa tag yang sering digunakan untuk memformat teks!
4. Sebutkan dan jelaskan beberapa tag yang digunakan untuk menampilkan tabel!
5. Sebutkan keuntungan menggunakan CSS!

KASUS PEMBOBOLAN ATM PADA KEGIATAN UKM

Cybercrime atau kejahatan dunia maya terdiri atas 3 (tiga) kategori utama, yaitu sebagai berikut:

1. Kejahatan Dunia Maya yang berkaitan dengan kerahasiaan, integritas dan keberadaan data dan sistem komputer;
2. Kejahatan Dunia Maya yang menggunakan komputer sebagai alat kejahatan; dan
3. Kejahatan Dunia Maya yang berkaitan dengan isi atau muatan data atau sistem komputer.

Bagai pisau yang sekaligus mempunyai fungsi mencelakakan dan bermanfaat bagi kehidupan manusia, demikian pula dengan internet dan komputer. Indonesia sebagai salah satu dari 5 negara terbesar pengguna internet di Asia, pun tak luput dari *cybercrime* tersebut.

Sepanjang tahun 2011 diwarnai dengan beberapa kasus-kasus *cybercrime*. Paparan ini akan lebih diciutkan lagi menjadi *cybercrime* yang terkait dengan pembobolan keamanan jaringan informasi, dan hanya menampilkan 3 (tiga) kasus saja yang berada dalam wilayah hukum negara Republik Indonesia.

1. Pembobolan distributor pulsa isi ulang. Diduga kerugian mencapai Rp. 1 Miliar. Kasus ini tergolong jarang terjadi, namun meskipun demikian harus menjadi perhatian khusus bagi aparat penegak hukum, karena banyak pengguna layanan pulsa isi ulang ini adalah pengusaha UKM dengan modal yang tidak seberapa.
2. Pembobolan ATM masih terus berlanjut. Dengan peralatan yang lumayan sederhana, beberapa ATM dapat "diserap" dananya dengan bebas, tanpa harus mempunyai akun di bank ATM bersangkutan. Kali ini lokasi kejahatan berada di Bali, dan tersangkanya yang ditangkap adalah Warga Negara Malaysia.
3. Kasus defacing website instansi pemerintah masih marak terjadi. Kali ini, website resmi Kepolisian Republik Indonesia diretas oleh hacker, dan diganti dengan content yang berbau SARA.

(Sumber : http://nella13.blogspot.com/2012/09/kasus-keamanan-komputer-3-kasus.html)

UNIVERSITAS NAROTAMA

BAB 4

Pemahaman Tentang PHP (Perl Hypertext Preprocessor)

Tujuan Instruksional Umum

Setelah mempelajari bab ini, Mahasiswa dipastikan dapat:

1. Menjelaskan pengertian PHP
2. Menjelaskan cara menyisipkan kode PHP ke HTML
3. Menjelaskan konsep dasar pembuatan PHP
4. Menjelaskan langkah – langkah pembuatan web yang baik

BAB

4

PEMAHAMAN TENTANG PHP *(Perl Hypertext Preprocessor)*

4.1. *Server-Side Scripting*

Situs *web* adalah layanan bagi *surfer* untuk melakukan penelusuran dengan cara me-*link* (*linking*). Situs yang baik adalah situs informasi yang di berikannya ***up to date*** dan lengkap yang hal ini merupakan masalah jika halaman web masih berbasis pada HTML yang statis. Masalahnya ialah seseorang harus repot mengganti dan mengubah isi file html nya beserta tag-tag nya.

Dengan *database* dan *programming-web* yang dinamik, hal ini akan terasa mudah sehingga pengelolanya tidak akan mengalami kerepotan dalam me-*maintenance web*-nya.

PHP disebut bahasa pemrograman *server side* karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti JavaScript yang diproses pada *web browser (client).*

PHP adalah singkatan dari *Pearl Hypertext Preprocessor*, merupakan *script-language* yang ditempatkan dan diproses di ***server***. Hasilnya akan dikirimkan ke klien melalui browser dalam bentuk kode HTML. Sifat ***server-side*** berarti pengerjaan/pemrosesan skrip akan dilakukan di server, baru kemudian hasilnya dikirimkan ke browser.

Keunggulannya dari sifatnya yang server-side tersebut antara lain:

- Tidak diperlukan kompatibilitas browser atau harus menggunakan browser tertentu, karena serverlah yang akan memproses skrip PHP. Hasil yang dikirimkan kembali ke browser umumnya bersifat teks atau gambar saja, sehingga pasti dikenal oleh browser apapun.
- Dapat memanfaatkan sumber-sumber aplikasi yang dimiliki oleh server, misalnya koneksi ke database.
- Skrip tidak dapat "diintip" dengan menggunakan fasilitas view HTML source.

- Koneksi layanan lain seperti protokol email seperti IMAP POP 3 ataupun SMTP
- PHP mampu bekerja pada Sistem Operasi yang bermacam-macam seperti Linux, Windows ataupun Sistem Operasi yang lain

Sebagai contoh, perhatikan skrip PHP sederhana berikut ini:

```
<HTML>
<HEAD> <TITLE> Contoh Skrip PHP </TITLE> </HEAD>
<BODY>
<?
   echo "<H1> Hallo, saya skrip PHP </H1>";
?>
</BODY>
</HTML>
```

Setelah skrip ini dikerjakan di server, server akan memberikan output kepada browser sebagai berikut:

```
<HTML>
<HEAD> <TITLE> Contoh Skrip PHP </TITLE> </HEAD>
<BODY>
<H1> Hallo, saya skrip PHP </H1>
</BODY>
</HTML>
```

Output inilah yang dieksekusi oleh browser. Dengan output yang berupa tag-tag HTML biasa, maka output tersebut pasti dapat dieksekusi oleh browser apapun.

Dapat dikatakan bahwa PHP dirancang untuk membentuk web yang dinamik dimana mampu menampilkan isi yang *up-date* seperti pengambilan dan penyimpanan data dari atau ke database. Database itu sendiri dapat diisi ataupun diubah oleh pemiliknya. Selain itu isi halaman web dapat dimanipulasi misalnya berubah-ubah hanya pada satu halaman web.

PHP dirancang untuk membentuk web yang dinamik dimana mampu menampilkan isi yang *up-date* seperti pengambilan dan penyimpanan data dari atau ke DATABASE. PHP dikembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group. Situs resmi PHP beralamat di *http://www.php.net.*

4.2. Menyisipkan Kode PHP dalam HTML atau Sebaliknya

Suatu skrip akan dikenali sebagai skrip PHP apabila dibatasi oleh karakter/delimiter seperti yang terdapat pada source berikut:

```
<html>
<head><title>Menyisipkan Kode PHP</title></head>
<body>
<!-- cara satu -->
<?
   echo "Halo PHP, ini skrip PHP pertamaku !!!<br>";
?>

<!—cara dua -->
<?php
   echo "Halo PHP 2<br>";
?>

<!—cara tiga -->
<script language="php">
   echo "Halo PHP 3<br>";
</script>
</body>
</html>
```

Pada awalnya PHP merupakan singkatan dari *Personal Home Page*. Sesuai dengan namanya, PHP digunakan untuk membuat website pribadi. Dalam beberapa tahun perkembangannya, PHP menjelma menjadi bahasa pemrograman web yang powerful dan tidak hanya digunakan untuk membuat halaman web sederhana, tetapi juga website populer yang digunakan oleh jutaan orang seperti Wikipedia, Wordpress, Joomla, dan lain-lain

Baris-baris pada skrip PHP dipisahkan dengan cara menambahkan karakter titik koma (;) pada akhir baris statement. Tanpa tanda (;) akan dianggap sebagai 1 baris statement. Contoh:

```
<?
  echo "Baris 1";
  echo "Baris 2";
?>
```

Jika terdapat skrip yang hanya terdiri dari satu baris, dapat digunakan alternatif penulisan selain cara di atas, yaitu:

```
<? echo "Hanya satu baris<br>" ?>
```

Sedangkan untuk menyisipkan kode HTML dalam PHP, dapat dilakukan dengan meyisipkannya pada fungsi echo, seperti contoh berikut:

```
<?php
   echo "<html>";
   echo "<head><title>Sisip HTML pada PHP</title></head>";
   echo "<body>";
   echo "<h2>Halo PHP !!!</h2>";
   echo "</body>";
   echo "</html>";
?>
```

4.3. Komentar dalam PHP

Untuk menambahkan komentar dalam skrip PHP dapat digunakan gaya C, C++, atau shell UNIX. Contoh:

```
<?php
    echo "Komentar gaya 1<br>"; // Komentar satu baris
gaya C++
    echo "Komentar gaya 2<br>";
    /* Komentar yang panjangnya
       lebih dari satu baris */
    echo "Komentar gaya 3<br>"; # Komentar gaya shell UNIX
?>
```

4.4. Pengenal *(Identifier)*

Pengenal (identifier) banyak digunakan dalam program untuk memberi nama variable, fungsi atau kelas suatu objek. Aturan penamaan identifier adalah penting dan harus unik, yang artinya suatu nama hanya sekali saja dideklarasikan dalam skrip PHP yang bersangkutan.

Aturan yang berlaku untuk penamaan identifier :

- Karakter yang dapat digunakan adalah huruf, angka atau garis bawah (_).
- Karakter pertama harus berupa huruf atau garis bawah
- Panjang identifier bebas berapa saja.
- Huruf kecil dan huruf besar dibedakan *(**case sensitive**)*

Hindarkan penggunaan nama yang sama dengan nama yang sudah dipakai atau sudah disediakan oleh PHP. Misalkan *printf, switch, echo* dan lain sebagainya.

Berikut ini adalah contoh identifier yang baik:

Identifier adalah suatu pengenal atau pengidentifikasi data yang kita deklarasikan agar kompiler dapat mengenalinya. Digunakan untuk mempermudah proses penanganan data atau nilai. Identifier dalam PHP terdiri dari fungsi, *variable* dan *classes.*

```
nama
user_name
_database_password
databasepassword88
```

Contoh yang salah:

```
23mei ( salah karena diawali angka )
user name ( salah karena ada spasi )
user-name ( salah karena ada tanda kurang )
```

4.5. Tipe Data

PHP mengenal dua tipe data sederhana, numerik dan literal serta tipe data array dan tipe numerik dapat menyimpan bilangan bulat, juga digunakan untuk menyimpan bilangan pecahan Tipe literal digunakan untuk menyimpan data berupa kumpulan huruf, kata, dan angka.Tipe boolean, yang dikenal dalam bahasa program yang lainnya, tidak ada dalam PHP.

Tipe data dasar pada PHP ada 3 jenis, yaitu:

- integer
- double
- string
- boolean

Integer, tipe data bilangan bulat dengan jangkauan +_ -2 milyar sampai +2 milyar. Double, tipe data bilangan real atau titik-mengambang, yaitu bilangan bagian pecahan. String adalah sederetan karakter baik angka, huruf maupun karakter lain yang tidak menyatakan literal numerik. Boolean adalah tipe data yang menyimpan nilai benar (*true*) atau salah (*false*).

Perlu diketahui, pada PHP dikenal sejumlah karakter yang menggunakan penulisan secara khusus yaitu didahului dengan symbol backslash (\ ).

Penulisan	Karakter yang dimaksud
\"	Petik ganda
\\	Backslash
\$	Tanda dolar
\n	Newline (pindah baris)
\r	Cariage return
\t	Tab
\x00 s.d \xFF	Karakter hexadecimal

4.6. Variabel

Variabel digunakan dalam program untuk menyimpan nilai yang berubah-ubah. Nilai variabel ini tergantung dari input masukan kita. Pada PHP variable ditandai dengan awalan karakter $. Setiap nama variabel dalam suatu skrip PHP harus unik, artinya suatu nama (*identifier*) hanya dipakai untuk satu variabel saja. Sebagai contoh anda dapat melihat contoh dibawah ini :

```
<html>
<head>
  <title>Contoh  Pemakaian variabel</title>
</head>
<body>
<?php
  $price = 3000;

  echo "Biaya ongkos untuk pergi ke Cisitu Rp $price";
?>
</body>
</html>
```

Sama seperti JavaScript, variabel dalam PHP juga memiliki lingkup global dan lokal, Dikatakan **global** jika tidak berada pada suatu fungsi sedangkan bersifat **lokal** bagi suatu fungsi jika berada pada suatu fungsi yang bersangkutan.

4.7. Operator

Operator adalah symbol yang digunakan dalam program untuk melakukan operasi, misalnya penjumlahan atau perkalian, pembandingan kesamaan dua buah nilai atau bahkan memberikan nilai ke variable. Nilai yang dioperasikan oleh operator (disebut operand atau argument). Bersama-sama operator memebentuk ekspresi (ungkapan). Sebagai contoh :

```
2 * 3 + 5 – 7
```

persamaan adalah sebuah ekspressi, dimana tanda *, +, dan – adalah operator, sedangkan 2, 3, 5, 7 adalah operand atau argument.

Adapun jenis-jenis Operator adalah sebagai berikut

- Operator Aritmetika
- Operator Assignment
- Operator String
- Operator Increment/Decrement
- Operator Logika
- Operator Perbandingan
- Operator Kontrol Error
- Operator tertiary
- Operator Kontrol Error

Operator PHP dapat anda lihat pada tabel dan daftar berikut:

Operator adalah alat yang digunakan untuk memanipulasi data seperti penjumlahan, pengurangan, perkalian, pembagian dan lainnya.
Operator digunakan untuk memanipulasi atau melakukan proses perhitungan pada suatu nilai *variable*.
Variable yang nilainya dimodifikasi oleh operator disebut *OPERAND*.

4.7.1. Operator Aritmetika

Contoh	Nama	Hasil
$a + $b	Penjumlahan	Jumlah $a dan $b
$a - $b	Pengurangan	Selisih $a dan $b
$a * $b	Perkalian	Perkalian antara $a dan $b
$a / $b	Pembagian	Bilangan bulat dari $a dibagi $b
$a % $b dengan $b	Modulus	Mengambil sisa hasil bagi $a

4.7.2. Operator Assignment (pemberian nilai)

Contoh	Operasi
$a = $b	Nilai $b diberikan kepada $a
$a += $b	Nilai $b dijumlahkan dgn $a kemudian diberikan ke $a
$a -= $b	Nilai $b dikurangkan dgn $a kemudian diberikan ke $a
$a *= $b	Nilai $b dikalikan dgn $a kemudian diberikan ke $a
$a /= $b	Nilai $b dibagikan dgn $a kemudian diberikan ke $a
$a %= $b	Nilai sisa hasil bagi $b dgn $a diberikan ke $a

4.7.3. Operator String

Contoh	Nama	Hasil
`$a . $b`	Concatenation	Menggabungkan string $a dan $b

4.7.4. Operator Increment/Decrement

Contoh	Nama	Hasil
`++$a`	Pre Increment	Tambahkan $a dgn 1 hasilnya masukkan kedalam $a
`$a++`	Post Increment	Berikan nilai $a lalu tambahkan 1 pada $a
`–$a`	Pre Decrement	Kurangkan $a dgn 1 hasilnya masukkan kedalam $a
`$a–`	Post Decrement	Berikan nilai $a lalu kurangkan1 pada $a

4.7.5. Operator Logika

Contoh	Nama	Hasil
(kond 1) && *(kond 2)*	And	Bernilai true jika kedua pernyataan kondisi benar.
(kond 1) || *(kond 2)*	Or	Bernilai true jika dua atau salah satu pernyataan kondisi benar.

4.7.6. Operator Perbandingan

Contoh	Nama	Hasil
$a == $b	Sama dengan	Benar jika $a sama dengan $b
$a != $b	Tidak sama	Benar jika $a tidak sama dengan $b
$a < $b	Lebih kecil	Benar jika $a kecil $b
$a <= $b	Kecil atau sama dengan	Benar jika $a kecil atau sama dengan $b
$a > $b	Lebih besar	Benar jika $a kecil $b
$a >= $b	Besar atau sama dengan	Benar jika $a Besar atau sama dengan $b

4.7.7. Operator *Tertiary*

Dalam PHP juga terdapat operator *tertinary* dengan 3 operan dengan skema:

```
operan_kondisional ? operan_statement_1 : operan_statement_2
```

Operan kondisional adalah keadaan yang kan dicek berupa operator **logika**, jika operan kondisional benar maka operan statement 1 akan dijalankan, jika tidak maka operan statement 2 lah yang dijalankan.

Contoh penggunaan :

```
$a = 5;
$b = $a == 5 ? 3 : 2;
echo $b; // b yang muncul bernilai 3 karena benar a = 5
```

4.7.8. Operator Kontrol Error

Operator Kontrol error *(error handling)* akan bekerja jika pada ekspresi PHP kita terdapat error. Operator ini diawali dengan keyword *or die*. Dapat anda lihat pada contoh berikut ini :

```
<?php
/* Error ini terjadi karena adanya petik ganda dalam petik ganda */
$res = "Nama saya jujun alias "jun"";
echo $res;
or die ("Maaf, terdapat kesalahan dalam skrip PHP ini !!!<br>");
?>
```

4.7.9. Statement Kontrol

Dalam skrip PHP terdapat 3 jenis proses, yaitu menjalankan instruksi secara beruntun (*sequences*), bercabang (sesuai kondisi) dan berulang-ulang (sesuai kondisi). Anda dapat melakukan ***abstraksi*** proses-proses ini dengan konsep statement ***pada JavaScript***, karena baik PHP maupun JavaScript memiliki skema yang sama.

Dalam PHP ada beberapa cara untuk menangani *error,* salah satunya yang paling simpel adalah menggunakan fungsi die().

Caranya adalah dengan

menyeleksi kondisi lalu jika tidak terpenuhi kita taruh fungsi die().

Jika fungsi die() dieksekusi maka program akan berhenti dan script program dibawahnya tidak

akan dijalankan.

4.7.9.1. Pernyataan If

Dipakai untuk mengambil keputusan berdasarkan suatu kondisi. *Statement* kondisi akan dijalankan bila kondisi bernilai benar.

Contoh:

```
<?php
// satu kasus
if(ekspressi)
{
  pernyataan;
}
// dua kasus
    if (ekspresi 1)
    {
      pernyataan 1;
    } else
    {
      pernyataan 2;
    }
// lebih dari 2 kasus
    if (ekspresi 1)
    {
      pernyataan 1;
    } else if (ekspresi 2)
    {
      pernyataan 2;
    } else if (ekspresi 3)
    {
      pernyataan 3;
    } else
    {
      pernyataan 4; // jika ekspresi 1,2,3 salah
    }
?>
```

Penggunaan switch hampir mirip dengan penggunaan if. Pernyataan switch juga menyediakan banyak efisiensi dan mudah dibaca. Pernyataan switch bekerja seperti pernyataan if, jika kondisi benar, maka blok kode yang sudah ditentukan akan dijalankan. Pernyataan switch hanya menerima satu argumen/kondisi, dan itu adalah nilai yang Anda buat sebagai ekspresi. Keyword case diikuti dengan nilai yang dibandingkan (atau ekspresi) dan ":(titik dua).

4.7.9.2. Pernyataan *Switch*

Dipakai untuk pengambilan keputusan yang menggunakan banyak pilihan yang didasarkan pada nilai dari ekspresi, yaitu berupa variabel.

Contoh:

```
swith(ekspressi)
{
    case ekspressi_case1:
          pernyataan_case1;
          break;
case ekspressi_case2:
          pernyataan_case2;
          break;
    case ekspressi_case3:
          pernyataan_case3;
          break;
    default:
    pernyataan default;
}
```

4.7.9.3. Pernyataan *While*

Digunakan untuk melakukan perulangan selama ekspresi masih bernilai benar. Tidak ada kemungkinan untuk menjalankan pernyataan perulangan sekali jika ekspresinya sudah bernilai salah.

Skema:

```
while(ekspressi)
{
      pernyataan_pernyataan
}
```

4.7.9.4. Pernyataan *Do While*

Sama seperti while, digunakan untuk melakukan perulangan selama ekspresi masih bernilai benar. Ada kemungkinan untuk menjalankan pernyataan perulangan sekali walaupun ekspresinya sudah bernilai salah.

Skema:

```
do
{
pernyataan_pernyataan;
} while(ekspressi);
```

Perulangan *while* dan *do-while* pada dasarnya hampir sama. Perbedaan terletak pada 'lokasi' pengecekan kondisi perulangan.

4.7.9.5. Pernyataan *For*

For statement terdiri atas 3 bagian yang dipisahkan oleh tanda semicolon (;) yang secara berurutan merupakan:

- ✓ *initial-expression:* berisi nilai awal, bersifat opsional, dan bisa berisi satu assignment atau lebih
- ✓ *condition:* berisi kondisi yang harus dipenuhi, dan harus ada. condition menentukan berlangsungnya loop, yang akan berhenti jika condition bernilai false
- ✓ *increment-expression:* berisi ekspresi yang akan mengubah nilai, bersifat opsional, dan bisa berisi satu assignment atau lebih

Skema :

```
for (initial_expression; condition; increment_expression);
{
   pernyataan_pernyataan;
}
```

Dalam struktur *while,* pengecekan untuk kondisi perulangan di lakukan di awal, sehingga jika kondisi tidak terpenuhi, maka perulangan tidak akan pernah dijalankan.

4.7.9.6. Pernyataan *Break*

Digunakan untuk keluar dari proses perulangan maupun pemilihan. Pemakaiannya bisa dilihat seperti pada *statement switch*. Skemanya:

```
break;
```

Contoh penggunaan *break* yang lain:

Namun pada perulangan do-while, pengecekan kondisi akan dilakukan di akhir perulangan, sehingga walaupun kondisi adalah FALSE, perulangan akan tetap berjalan minimal 1 kali.

```
➔ break statement di bawah ini akan menghentikan while
loop jika i = 3
<?php
    $i = 0;
    while ($i < 6) {
         echo "break $i <BR>";
         if ($i == 3) break;
         $i++;
    }
?>
```

4.7.9.7. Pernyataan *Continue*

Continue statement menghentikan eksekusi blok statement dalam loop while atau for dan melanjutkan eksekusi loop untuk iterasi/pengulangan selanjutnya. Skema *continue statement* adalah sebagai berikut:

```
continue;
```

Perintah *continue* juga digunakan untuk menginterupsi perulangan dalam PHP, namun jika perintah *break* digunakan untuk menghentikan perulangan, maka perintah *continue* hanya akan menghentikan perulangan untuk 1 iterasi saja, lalu proses perulangan akan dilanjutkan.

Tidak seperti *break statement, continue* tidak menghentikan keseluruhan *loop,* tetapi:

– pada *while loop,* mengembalikan ke *condition*

– pada *for loop,* mengembalikan ke *increment expression*

Contoh *continue:*

```
<?php
  $i = 0;
  $n = 0;
  while ($i < 5) {
    $i++;
    echo "continue $i <BR>");
    if ($i == 3) continue;
    $n += $i;
    echo "n= $n <BR>";
  }
?>
```

Proses yang dilakukan oleh *script* di atas:

- *continue* 1

  n = 1
- *continue* 2

  n = 3
- *continue* 3
- *continue* 4

  n = 7

- *continue* 5

  n = 12

Contoh Penggunaan Statement Kontrol *(pada contoh ini digunakan fungsi date, pembuatan fungsi yang akan dibahas pada bagian berikutnya)*:

```
<html>
<head>
<title>Statement Control</title>
</head>
<body>
<?php
/* Penggunaan if else */
   function day()
   {
   $nama_hari=date("D");
   if ($nama_hari=="Sun")
         echo "Minggu";
   elseif($nama_hari=="Mon")
   echo "Senin";
   elseif($nama_hari=="Tue")
   echo "Selasa";
   elseif($nama_hari=="Wed")
   echo "Rabu";
   elseif($nama_hari=="Thu")
   echo "Kamis";
   elseif($nama_hari=="Fri")
   echo "Jumat";
   else
   echo "Sabtu";
   }
   $looping=1;
/* LOOPING 2 kali dengan while */
   while ($looping<3)
   {
   echo day();
     echo "<br>";
     $looping++;
   }
/* Statement Control untuk hari */
   function hari()
   {
      $nama_hari = date("D");
      switch($nama_hari)
      {
         case "Sun":
               echo "Minggu";
            break;
                     case "Mon":
                           echo "Senin";
                           break;
                     case "Tue":
         echo "Selasa";
         break;
                           case "Wed":
                           echo "Rabu";
```

```
            break;
    case "Thu":
    echo "Kamis";
    break;
          case "Fri":
    echo "Jumat";
    break;
          default :
          echo "Sabtu";
          break;
        }
    }
  /* Proses LOOPING */
  echo "Looping ini menggunakan FOR untuk looping 2
kali<br>";
  for ($looping=1;$looping<3;$looping++)
  {
       echo hari();
       echo "<br>";
  }
?>
</body>
</html>
```

PENGERTIAN FUNGSI *(FUNCTION)* DALAM PHP

Dalam merancang kode program, kadang kita sering membuat kode yang melakukan tugas yang sama secara berulang-ulang, seperti membaca tabel dari *database,* menampilkan penjumlahan, dan lain-lain. Tugas yang sama ini akan lebih efektif jika dipisahkan dari program utama, dan dirancang menjadi sebuah fungsi.

Fungsi *(Function)* dalam bahasa pemograman adalah kode program yang dirancang untuk menyelesaikan sebuah tugas tertentu, dan merupakan bagian dari program utama. Kita dapat membuat fungsi sendiri, atau menggunakan fungsi yang dibuat oleh programmer lain.

Dalam dunia pemograman terdapat istilah *'lazy progamming'* yang artinya bukanlah programmer yang malas. Tetapi, daripada membuat kode program umum dari dasar, kita bisa menggunakan fungsi yang telah dibuat oleh programmer lain. PHP bahkan menyediakan ribuan fungsi bawaan yang tersedia untuk membantu kita dalam merancang program.

Mengetahui cara penggunaan fungsi ini akan menghemat waktu pembuatan program dan membuat kode program menjadi lebih efisien. *Lazy programming is smart programming.*

4.7.10. Fungsi dalam PHP

Script yang berukuran besar umumnya melibatkan fungsi-fungsi yang diciptakan sendiri oleh pemogram. Fungsi adalah blok kode yang ditujukkan untuk melaksanakan suatu tugas tertentu. Fungsi digunakan berkali-kali pada semua bagian yang dibutuhkan dalam skrip tersebut.

Mendeklarasikan fungsi dapat dilakukan dengan menggunakan pernyataan dengan bentuk seperti di bawah ini.

```
function nama_fungsi ( parameter )
{
    blok kode
}
```

Sebagai contoh fungsi di bawah ini meminta argumen berupa literal boolean:

```
<html>
<head>
<title>Contoh Fungsi Pemeriksa TIPE</title>
</head>
<body>
<?php
function true_false($argumen)
{
    if ($argumen)
          return "benar";
    else
          return "salah";
}
```

```
$ada = isset ($music);
echo " music ada ngaa ?".true_false($ada)."";
$music = "Rock";
$ada = isset ($music);
echo " <br>music ada ngaa ?".true_false($ada)."";
$lihat_kosong = empty($music);
echo " <br>music ada ngaa ?".true_false($lihat_kosong)."";
unset($music);
echo " <br>music ada ngaa ?".true_false($ada)."";
?>
</body>
</html>
```

Dalam PHP juga sudah terdapat fungsi yang *built-in,* yaitu fungsi bawaan dari PHP sehingga kita tinggal memakainya. mengkonversinya dari suatu tipe data ke tipe data yang lain sebagai berikut:

Fungsi	Keterangan
gettype($x)	Untuk memperoleh tipe data yang terdapat pada sebuah variabel
is_aray($x)	Untuk mengetahui variabel berupa array atau tidak
is_bool($x)	Untuk mengetahui variabel merupakan Boolean *(true or false)*
is_double($x)	Untuk mengetahui variabel bertipe *double* atau tidak
is_float($x)	Identik dengan *is_double*
is_int($x)	Identik dengan *is_float*
is_integer($x)	Identik dengan *is_int*
is_numeric($x)	Untuk mengetahui variabel bertipe bilangan atau *string*
is_object($x)	Untuk mengetahui variabel berupa *object* atau tidak
is_real($x)	Identik dengan *is_double*
is_string($x)	Untuk mengetahui variabel berupa *string* atau tidak
empty($x)	Untuk mengetahui apakah variabel bernilai kosong atau tidak
isset($x)	Untuk mengetahui apakah variabel sudah ada atau tidak
unset($x)	Untuk menghapus variabel

Selain itu terdapat operasi fungsi untuk *array, string,* dan *date* yang akan dijelaskan satu per satu pada bagian berikutnya.

4.7.10.1. Operasi *Array*

Array merupakan variabel yang menyimpan lebih dari satu nilai, Setiap nilainya akan mempunyai indeks yang unik dengan *default indeks* angka 0,1,2,...,dst. Sebagai contoh *array* dapat Anda lihat di bawah ini:

```
$name = array("emil","ochie","ambo","mia");
Atau dapat juga ditulis sebagai berikut
$name[] = "emil";
$name[] = "ochie";
$name[] = "ambo";
$name[] = "mia";
Atau dapat juga ditulis sebagai berikut
$name[0] = "emil";
$name[1] = "ochie";
$name[2] = "ambo";
$name[3] = "mia";
```

Array adalah satu rangkaian elemen *(variable)* bertipa sama yang menempati memori secara teratur yang dapat diakses secara individu dengan menambahkan suatu *index (subscript)* kepada suatu nama unik. Array disebut juga larik.

Secara praktek angka tersebut tidak harus dimulai dari nol, anda bisa menenentukannya dengan bebas, misalnya

```
$name[34] = "emil";
$name[64] = "ochie";
$name[33] = "ambo";
$name[21] = "mia";
Atau pula dengan cara membuat index dari variable berupa
string, misalnya :
$name[my_name] = "emil";
$name[my_assist] = "ochie";
$name[my_pren] = "ambo";
$name[my_induksemang_daugter] = "mia";
Ada cara penulisan array yang lain dengan menunjukkan
index misalnya :
$name = array(34=>"emil", 64=>"ochie", 33=>"ambo",
21=>"mia");
```

Coba anda lihat contoh di bawah ini:

```
<html>
<head>
<title>Contoh Array</title>0
</head>
<body>
<?php
$name = array(34=>"emil", 64=>"ochie", 33=>"ambo",
21=>"mia");
echo "Array yang berindex 34: $name[34]";
echo "Array yang berindex 64: $name[64]";
echo "Array yang berindex 33: $name[33]";
echo "Array yang berindex 21: $name[21]";
?>
</body>
</html>
```

Mengakses elemen array dengan melakukan perulangan, seperti :

```
<?php
  $x = array("Saya","Kamu","Dia","Kalian");
  for ($i=0;$i < count($x); $i++)
```

```
  {
     echo "$x[$i] , ";
     // menampilkan : Saya, Kamu, Dia, Kalian
  }
?>
```

4.7.10.1.1. *Array* Multi Dimensi

PHP memungkinkan sebuah *array* mengandung *array* dan bahkan sejumlah *array* dengan kata lain *Array* Multi dimensi adalah *array* dalam *array*. Di bawah ini adalah *script* untuk membentuk *array* multidimensi dan mengakses setiap elemennya.

```
 <html>
<head>
<title>Multi Dimensi ARRAY</title>
</head>
<body>
<?php
$something = array ("Jurusan" => array ("Informatika","Elektro"),
                 "Fakultas" => array ("FTIf","FTI","FTSP"),
                    "Training" => array ("WebMaster","J2ME"));

// Menampilkan isi dari array multidimensi
while ( list($index1,$nilai1) = each($something))
{
   echo("$index1 : <br>");
   $nomor = 1;
   while (list($index2,$nilai2) = each($nilai1))
   {
   echo ("$nomor.$nilai2 <br>");
   $nomor++;
   }
}
?>
</body>
</html>
```

Ada beberapa fungsi yang berkaitan dengan operasi array yang sering digunakan antara lain:

Jenis Fungsi	Deskripsi
count	Menghitung Jumlah Elemen Array Contoh : $x = array("aku","saya","kamu"); $jumlah = count($x); echo "$jumlah "; // akan menampilkan literal 3
ksort	Mengurutkan Array berdasarkan indeksnya dengan urutan naik (*ascending*)

Jenis Fungsi	Deskripsi
krsort	Mengurutkan Array berdasarkan indeksnya dengan urutan terbalik (*descending*)
sort	Mengurutkan data pada Array dengan urutan naik (*ascending*)
rsort	Mengurutkan data pada Array dengan urutan terbalik (des*cending*)
asort	Mengurutkan data Array dengan pola naik (*ascending*) dengan tetap mempertahankan indeksnya.
arsort	Mengurutkan data Array secara menurun (des*cending*) dengan tetap mempertahankan indeksnya.
shuffle	Mengacak Array
array_rand	Memperoleh satu atau sejumlah indeks pada Array secara random
in_array	Memeriksa keberadaan suatu nilai pada Array

explode	Membuat Array dari suatu literal string dengan memakai suatu karakter pemisah.
	Contoh :
	$x = "Saya – Kamu – Dia –Kalian";
	$hasil = explode("-",$x);
	for ($i=0;$i < count($hasil); $i++)
	{
	echo "$hasil[$i] , ";
	// menampilkan : Saya, Kamu, Dia, Kalian
	}

implode	Mengubah array menjadi literal string dengan memakai suatu karakter sebagai penggabung. Contoh : $x = array("Saya","Kamu","Dia","Kalian"); $hasil = implode("-",$x); echo "$hasil"; // menampilkan : Saya–Kamu-Dia-Kalian

4.7.11. Operasi *String*

String merupakan sederetan karakter huruf, angka maupun karakter lain yang padu. Pendefenisian/pendeklarasian *string* dilakukan dengan proses *assignment* langsung atau sebagai konstanta *string.*

Perhatikan contoh berikut ini:

```
<?php
  $kalimat = "Selamat Belajar PHP"; // pendeklarasian string
  echo $kalimat;
  echo "<br>Semoga Sukses !!!"; // konstanta string
?>
```

Ada beberapa fungsi yang berkaitan dengan operasi string yang sering digunakan antara lain:

Jenis Fungsi	Deskripsi, Skema dan Contoh
substr	Mendapatkan isi dari suatu string. substr("makan",1,3) hasil adalah "aka"
str_repeat	Mengulang string yang ditunjuk str_repeat("a",3) hasilnya adalah "aaa"
str_replace	Mengganti bagian dari suatu string dengan string yang lain pada suatu object str_replace("saya","gw","saya makan") pada hasil "gw makan"
ereg	Fungsi ekspressi regular ereg (pola, sumber) nb: *case sensitive* tambahan ^ di awal akan meminta string yang cocok di awal, dan $ pada akhir dengan [] akan meminta paling sedikit satu substring harus ada.
eregi	Sama dengan ereg Beda *non case sensitive*
Crypt	Melakukan enkripsi pada string yang ditunjuk

```
<html>
<head>
  <title>Contoh Pemakaian operasi-operasi pada string</title>
</head>
<body>
<form name="form">
  <input type="text" name="email">
  <input type="submit" value="cek">
</form>
<?php
  if(isset($email))
   if(empty($email))
         echo "Tolong isi ...!!!";
   else
   /* untuk cek emailnya */
   if(ereg("^.+@.+\\..+$",$email))
         echo "Sudah benar";
   else
         echo "Salah tuch";
?>
</body>
</html>
```

4.7.12. Operasi *Date*

4.7.12.1. Fungsi *Date*()

Fungsi date digunakan untuk menampilkan tanggal dan/atau waktu sekarang. Sintaks penggunaannya adalah sebagai berikut:

```
date(format[,timestamp])
```

Timestamp adalah waktu yang diukur dari jumlah detik sejak waktu UNIX Epoch, yaitu 1 Januari 1970, 00:00:00 GMT.

Format adalah karakter-karakter yang digunakan untuk memformat tampilan tanggal dan/atau waktu.

Timestamp adalah waktu yang diukur dari jumlah detik sejak waktu UNIX Epoch, yaitu 1 Januari 1970, 00:00:00 GMT. Maksudnya adalah jika angka timestamp dituliskan 10 itu berarti tanggal yang dimaksud adalah 1 January 1970, 00:00:10 GMT. Waktu ini akan menyesuaikan dengan waktu lokal, jadi jika waktu lokal Indonesia adalah GMT +7, maka jika angka timestamp dituliskan 10 itu berarti tanggal yang dimaksud adalah 1 January 1970, 07:00:10. Jika timestamp tidak disebutkan, maka yang diambil adalah waktu lokal pada saat itu.

Karakter-karakter yang digunakan untuk format adalah:

Karakter	Arti
a	“am” atau “pm”
A	“AM” atau “PM”
B	Swatch Internet time
d	Hari dalam satu bulan, 2 digit dengan diawali nol, dari “01” sampai “31”
D	Hari dalam satu minggu, tekstual, 3 huruf; misal “Fri”, “Sun”
F	Bulan, tekstual, lengkap, misalnya “March”, “May”
g	Jam, format 12 jam tanpa diawal nol, dari “1” sampai “12”
G	Jam, format 24 jam tanpa diawal nol, dari “1” sampai “23”
h	Jam, format 12 jam, dari “01” sampai “12”
H	Jam, format 24 jam, dari “00” sampai “23”
i	Menit, dari “00” sampai “59”
I	“1” jika Daylight Savings Time, “0” jika tidak.
j	Hari dalam satu bulan, tanpa diawali nol, dari “1” to “31”
l	Hari dalam satu minggu, tekstual, lengkap, misalnya “Friday”, “Monday”.
L	Bernilai “1” untuk tahun kabisat, “0” untuk bukan.
m	Bulan dalam angka “01” sampai “12”

Karakter	Arti
M	Bulan, tekstual, 3 huruf, misal "Jan", "Mar"
n	Bulan dalam angka tanpa diawali nol, dari "1" sampai "12"
r	Format tanggal RFC 822, misalnya "Thu, 21 Dec 2000 16:01:07 +0200"
s	Detik, dari "00" sampai "59"
S	Akhiran yang menunjukkan angka dalam Bahasa Inggris, tekstual, 2 huruf, misalnya "th", "nd"
t	Jumlah hari dalam satu bulan, dari "28" sampai "31"
T	Setting zona waktu pada komputer, misal "MDT"
U	Jumlah detik sejak Unix Epoch.
w	Hari dalam angka untuk satu minggu, "0" untuk Minggu sampai "6" Sabtu.
Y	Tahun, 4 digit, misal "2001"
y	Tahun, 2 digit, misal "99"
z	Hari dalam angka untuk satu tahun; dari "0" sampai "365"
Z	Setting zona waktu dalam detik, dari "-43200" sampai "43200". Sebelah barat UTC bernilai negatif, dan sebelah timur UTC bernilai positif.

Contoh penggunaannya adalah sebagai berikut:

```
<HTML>
<HEAD>
<TITLE> Date </TITLE>
</HEAD>

<BODY>
<FONT SIZE=5>
<?
echo "Sekarang adalah tanggal ";
echo date('d-F-Y');
echo "<BR>dan jam ";
echo date('h:i:s A');
?>
</FONT>
</BODY>
</HTML>
```

4.7.12.2. Fungsi *Getdate()*

Fungsi *getdate*() digunakan untuk mengambil nilai waktu lokal sekarang atau waktu *timestamp* dan waktu memasukkannya ke dalam *array* asosiatif. Sintaksnya adalah sebagai berikut:

```
getdate([timestamp])
```

Elemen *array* yang dapat dipergunakan adalah sebagai berikut:

- "minutes" = menit
- "seconds" = detik
- "mday" = hari dalam satu bulan
- "hours" = jam, dalam format 24 jam.
- "wday" = hari dalam satu minggu, numeris, 0 untuk Minggu hingga 6 untuk Sabtu
- "mon" = bulan, numeris.
- "year" = tahun, numeris.
- "yday" = hari dalam satu tahun, misalnya "299"
- "weekday" = hari dalam satu minggu, tekstual penuh, misalnya "Friday"
- "month" = bulan, tekstual penuh, misalnya "January"

Sebagai contoh misalnya ingin dibuat halaman selamat datang yang akan menyapa pengunjung dengan salam Selamat Pagi/Siang/Sore/Malam sesuai dengan waktu saat itu. Skripnya adalah sebagai berikut:

FUNGSI *GETDATE()*

Dalam pemrograman website dinamis, fungsi waktu sangat vital karena mencatat semua aktivitas yang terjadi pada website tersebut. Peran tanggal berfungsi untuk menandai waktu pos sebuah artikel, mencatat jumlah kunjungan per hari dan masih banyak lagi fungsi yang lainnya.

Berikut ini penjelasan masingmasing fungsi date(), mktime() dan strtotime(). Fungsi date() Adalah fungsi untuk menampilkan waktu server sekarang. Fungsi ini mengambil waktu dari server dan menampilkannya sesuai format yang diminta oleh script. PHP menyediakan beberapa parameter untuk fungsi date(). Parameter ini berkaitan dengan format data yang dikembalikan oleh fungsi tersebut.

Untuk menangani tanggal dan waktu, PHP beberapa fungsi diantaranya, fungsi date(), mktime() dan strtotime(). Contoh penggunaan fungsi-fungsi ini adalah pada waktu mencatat dan menyimpan waktu pos sebuah artikel pada blog. Fungsi-fungsi ini juga dipakai saat membuat penjadwalan (scheduler) seperti kontrol perawatan alat dan atau sekedar membuat pemberitahuan tanggal ulang tahun seseorang.

```
<HTML>
<HEAD>
  <TITLE> Getdate </TITLE>
</HEAD>
<BODY>
  <CENTER>
  <H1>
  <?php
    $skr = getdate();
    $bulan = $skr['month'];
    $hari = $skr['mday'];
    $tahun = $skr['year'];
    $jam = $skr['hours'];
    if ($jam <= 11) {
      echo "Selamat Pagi";
    } elseif ($jam > 11 and $jam <= 15) {
      echo "Selamat Siang";
    } elseif ($jam > 15 and $jam <= 18) {
      echo "Selamat Sore";
    } elseif ($jam > 18) {
      echo "Selamat Malam";
    }
  ?>
  </H1>
  <H2> Selamat datang di situs kami </H2>
  <H3> Sekarang adalah tanggal
  <?php
    echo "$hari $bulan $tahun";
  ?>
  </H3>
</BODY>
</HTML>
```

4.8. Mengikutkan Skrip dari File Lain

4.8.1. Pernyataan *Include*

PHP memungkinkan adanya pernyataan Include sedemikian sehingga anda hanya cukup menuliskan kode yang sama (*yang tersimpan dalam suatu file*) sekali saja dalam banyak berkas.

```
include("nama_file");
Misalnya pada program dibawah ini
/* nama file – Includetest.php  */
<html>
<head>
<title>Test Include</title>
</head>
<body>
<?php
  echo "Test Include<br>";
  echo "Ini include yang pertama<br>";
  include "header.php";
  echo "<br>";
  echo "Ini include kedua<br>";
  include "header.php";
echo "<br>";
?>
</body>
</body>
</html>

/* –– file header.php –– */
<?
  echo "INI HEADERNYA";
?>
```

4.8.2. Pernyataan *Require*

Pernyataan require memungkinkan kita untuk menggabungkan suatu script PHP atau text dari file yang lain dengan script PHP yang memanggilnya. Skema:

```
require("nama_file");
```

4.9. Menangani Berkas

Untuk membuat suatu aplikasi bukutamu berbasis *file* maka kita harus mengerti bagaimana cara menangani *file.* Dalam PHP sendiri telah disediakan fitur penanganan *file.* Secara umum operasi terhadap *file* melalui tiga tahap, yaitu meliputi:

1. membuka file
2. memanipulasi file
3. menutup file

PERNYATAAN *INCLUDE* DAN *REQUIRE* PADA PHP

Pernyataan *Include* dan Pernyataan *Require* pada PHP dapat digunakan untuk Menyertakan File lain, dalam bentuk PHP, HTML maupun Teks, ke suatu halaman PHP. Perbedaan antara Pernyataan *Include* dengan Pernyataan *Require* terletak pada perlakuannya terhadap suatu kegagalan atau kesalahan:

Pernyataan ***Include*** akan menghasilkan peringatan (E_WARNING) dan eksekusi kode-kode berikutnya tetap berlanjut.

Pernyataan ***Require*** akan menghasilkan kesalahan (E_COMPILE_ERROR) dan eksekusi kode-kode berikutnya akan dihentikan.

Menyertakan File Lain ke suatu halaman PHP dengan kedua cara ini akan banyak menghemat waktu terutama jika kita ingin melakukan perubahan, oleh karena, kita hanya melakukan pembaharuan pada satu file sumbernya saja; tidak pada keseluruhan halaman dimana file sumber tersebut diikut sertakan.

Pernyataan *Include* pada PHP

Menyertakan File Lain ke suatu halaman PHP bisanya didasarkan alasan bahwa suatu PHP, HTML, atau Teks yang sama akan digunakan pada berbagai halaman PHP yang membangun website yang akan kita bentuk. Misalnya kita akan menggunakan halaman untuk 'judul', atau untuk 'menu', ataupun untuk 'catatan kaki', pada keseluruhan halaman yang akan kita gunakan untuk mebangun suatu website.

Gunakan Pernyataan Include untuk Menyertakan File Lain ke suatu halaman PHP, jika kita ingin eksekusi tetap berlanjut dengan ketiadaan file yang diikut sertakannya. Sintaks Pernyataan Include pada PHP

```
include 'NamaFile'
```

Pernyataan Require pada PHP

Seperti halnya Pernyataan Include, Pernyataan Require pada PHP juga digunakan untuk Menyertakan File Lain ke dalam suatu halaman PHP.

Gunakan Pernyataan Require untuk Menyertakan File Lain ke suatu halaman PHP, jika file yang akan diikut-sertakan menjadi kunci penentu bagi aliran eksekusi secara keseluruhan, olehkarena, penghetian eksekusi akan menjamin keamanan dan integritas aplikasi yang telah kita bangun saat file sumber menghilang secara tidak disengaja. Adapun sintaks Pernyataan Require pada PHP

```
require 'NamaFile'
```

http://m261an.com/belajar-php/PHPc_Include Require.php

Selain ketiga hal diatas terdapat satu konsep lagi yakni penunjuk offset pada berkas file. Penunjuk berkas berperan saat pembacaan dan perekaman data. Pada saat anda melakukan pembacaan maka data yang ditunjuk oleh penunjuk berkaslah yang akan dibaca. Begitu pula saat perekaman data. Data yang ditulis dimulai pada posisi yang ditunjuk oleh penunjuk berkas.

4.9.1. Fungsi *Fopen* (Membuka *File*)

Bentuk pemanggilan fungsi ini:

```
$variabel_penyangga = fopen(nama_file, mode);
```

Nama_file adalah nama file yang akan dibuka

Nilai balik dari fungsi *fopen* adalah *variabel handle* yang isinya merupakan *handle/id* koneksi ke *file* yang telah dibuka, bernilai *false* jika gagal dibuka.

Variabel penyangga akan berfungsi sebagai penyagga *file* selama proses manipulasi.

Mode adalah mode yang digunakan dalam membuka *file,* ada beberapa mode yang dapat dipakai, dibawah ini adalah daftar mode pembukaan *file.*

Mode	Keterangan
r	Mode hanya bisa baca, penunjuk berkas diawal file
r+	Mode baca dan tulis, penunjuk berkas diawal file
w	Mode penulisan saja, bila file belum ada akan diciptakan, jika ada akan dioverwrite
w+	Mode baca dan tulis, sama seperti diatas namun bisa membaca
a	Mode untuk penambahan data, penunjuk berkas di akhir file, apabila file tidak ada akan diciptakan
a+	Mode baca dan tulis, sama seperti a hanya bisa membaca file, apabila file belum ada akan diciptakan
b	Mode binary artinya file dibaca sebagai binary file bukan ascii file, jadi data diperlakukan sebagai data biner bukan character ASCII

4.9.2. Fungsi *Fclose* (Menutup File)

Bentuk pemanggilan fungsi ini:

```
fclose(id_koneksi)
```

Dengan:

Id_koneksi adalah return value hasil pemanggilan *fopen*

Nilai balik berupa *TRUE* jika berhasil ditutup dan *FALSE* jika gagal ditutup

4.9.3. Fungsi *Fgets* (Membaca Data dari *File*)

Bentuk pemanggilan:

```
string fgets(id_koneksi, panjang)
```

String adalah nilai balik fungsi

Id_koneksi adalah return value pemanggilan fungsi fopen

Panjang adalah panjang data yang akan dibaca

4.9.4. Fungsi *Fputs* (Menulis Data ke *File*)

Bentuk pemanggilan fungsi:

```
fputs(id_koneksi, string)
```

Id_koneksi adalah return value pemanggilan fungsi fopen

String adalah string yang akan ditulis ke file

4.9.5. Fungsi Feof

(Apakah penunjuk berkas ada diakhir atau tidak)

bentuk pemanggilan:

```
feof(id_koneksi)
```

id_koneksi adalah *return value* pemanggilan fungsi *fopen*

bernilai *TRUE* jika penunjuk berkas di akhir *file*

bernilai *FALSE* jika penunjuk berkas tidak di akhir *file*

Berikut ini adalah contoh penggunaan fungsi-fungsi diatas untuk melakukan operasi *file*

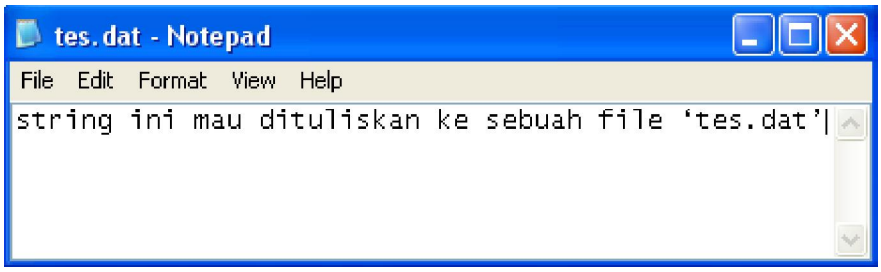
contoh operasi tulis ke file:

```
script:

<?php    $string = "string ini mau dituliskan ke sebuah file 'tes.dat'";   $fp
= fopen("tes.dat","w"); //membuka dengan mode tulisif($fp){
fputs($fp,$string);        fclose($fp);      }?>
```

Maka akan terbentuk sebuah *file* 'tes.dat' pada direktori yang sama dengan *script,* kalau dilihat dengan *notepad* hasilnya adalah:

```
<?php
      $string = "string ini mau dituliskan ke sebuah file 'tes.dat'";

      $fp = fopen("tes.dat","w"); //membuka dengan mode tulis
      if($fp){
            fputs($fp,$string);
            fclose($fp);
      }
?>
```

maka akan terbentuk sebuah *file* 'tes.dat' pada direktori yang sama dengan *script,* kalau dilihat dengan *notepad* hasilnya adalah:



contoh operasi baca dari file:

```
<?php
        $fp = fopen("tes.dat","r");
        if($fp){
                $string = fgets($fp,1024);
                fclose($fp);
        }
        //tampilkan
        echo $string
?>
```

4.10. Menangani *Form*

4.10.1. Metode *Get* dan *Post*

Perbedaannya metode ***get*** akan mengirimkan informasi dan ***menunjukkan infomasi yang dikirim*** yang menempel pada alamat url dari skrip php yang memproses. Sedangkan metode ***post tidak akan menunjukkan informasi yang dikirim***, dalam arti bersifat rahasia (*private*).

Metode *Get* dan *Post* digunakan untuk mengirimkan suatu informasi dimana informasi yang dikirim akan disimpan sebagai literal/tipe data *string* oleh skrip PHP yang memrposesnya. Perbedaannya metode ***get*** akan mengirimkan informasi dan ***menunjukkan infomasi yang dikirim*** yang menempel pada alamat url dari skrip php yang memproses. Sedangkan metode ***post tidak akan menunjukkan informasi yang dikirim***, dalam arti bersifat rahasia (*private*).

Jika Anda aktif melakukan browsing di dunia maya, Anda pasti tidak asing dengan *form* HTML. *Form* HTML ini dijumpai misalnya pada saat Anda diminta mengisi formulir pendaftaran di suatu situs, mengisi buku tamu, mengisi jajak pendapat, dan lain-lain. Nah, jika form tidak lagi asing bagi Anda, mungkin yang terasa asing adalah bagaimana *form* tersebut diolah.

Untuk menyegarkan kembali ingatan Anda tentang *form* HTML, berikut akan diberikan sintaks penggunaan form HTML tersebut:

```
<FORM [NAME=namaform] METHOD={post|get} ACTION=pengolah>
...komponen form...
</FORM>
```

Untuk mengatur agar form tersebut diolah oleh skrip PHP, maka argumen *ACTION* harus diisi dengan nama sebuah file PHP. *File* inilah yang nantinya akan mengolah *form* tersebut.

Pertanyaan berikutnya adalah: apanya yang diolah? Tentu saja adalah input yang diberikan kepada form form tersebut. Seperti kita tahu, komponen form HTML dapat berupa kotak teks, radio button, dan check box. Nah, nilai-nilai yang diberikan ke dalam komponen-komponen inilah yang nantinya diolah. Nilai-nilai tersebut nantinya akan "ditangkap" oleh skrip PHP dan disimpan dalam bentuk variabel. Nama variabel yang menyimpannya sama dengan nama komponen form HTML yang didefinisikan oleh argumen NAME. Perhatikan contoh form berikut ini:

... komponen *form* HTML dapat berupa kotak teks, *radio button,* dan *check box.* Nilai-nilai yang diberikan ke dalam komponen-komponen inilah yang nantinya diolah. Nilai-nilai tersebut nantinya akan "ditangkap" oleh skrip PHP dan disimpan dalam bentuk variabel

```
<HTML>
<HEAD>
    <TITLE> Form </TITLE>
</HEAD>
<BODY>
<FORM NAME="contoh" METHOD="post" ACTION="proses.php">
    Nama <INPUT TYPE="text" NAME="nama"> <BR>
    Umur <INPUT TYPE="text" NAME="umur"> <BR>
    Email <INPUT TYPE="text" NAME="email"> <BR>
    <INPUT TYPE="submit">
</FORM>
</BODY>
</HTML>
```

Simpanlah form ini dengan nama cthform.html.

Nantinya jika form tersebut di-*submit, input* berupa nama, umur, dan email akan diolah di *file* proses.php. Di dalam *file* proses.php ini nantinya akan dikenal variabel dengan nama $_POST["nama_komponen"] atau $_GET["nama_komponen"] sesuai dengan nama komponen *form* HTML *(didefinisikan oleh atribut NAME)* dan metode yang digunakan (*get* atau *post*). Jadi untuk contoh *form* di atas, pada *file* proses.php akan dikenal *array* dengan nama $_POST['nama'], $_POST['umur'], dan $_POST['email'].

Sebagai alternatif, sejak PHP 3 juga tersedia *array* dengan nama $HTTP_POST_VARS dan $HTTP_GET_VARS. Kedua *array* ini bersifat asosiatif dan sebagai indeksnya adalah nama dari komponen *form* HTML yang didefinisikan oleh argumen NAME. Jadi untuk contoh *form* di atas, pada *file* proses.php akan dikenal *array* dengan nama $HTTP_POST_VARS['nama'], $HTTP_POST_VARS['umur'], dan $HTTP_POST_VARS['email'].

Dengan pemahaman tersebut, sekarang kita akan membuat *file* proses.php yang mengolah input dari cthform.html di atas:

```
<HTML>
<HEAD>
    <TITLE> Proses </TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF>
<?php
    $nama = $_POST['nama'];
    $umur = $_POST['umur'];
    $email = $_POST['email'];
    echo "<H1> Hallo $nama </H1>";
                    ?>
```

```
<H2> Selamat Datang Di Situs Kami </H2>
<?php
    echo "Umur Anda saat ini adalah ";
    echo "$umur tahun <BR>";
    echo "Alamat email Anda adalah ";
    echo "$email";
?>
</BODY>
</HTML>
```

4.10.2. Verifikasi form

Dalam pembuatan form, tentunya Anda ingin agar form tersebut diisi dengan benar. Untuk itu diperlukan suatu verifikasi form. Verifikasi form dapat dibuat dengan PHP (*server side script*) dan dapat juga dibuat dengan client side script seperti misalnya JavaScript. Jika verifikasi form tidak memerlukan data-data dari server sebagai pembanding, maka sebaiknya digunakan client side script. **Pertimbangannya adalah client side script tidak memerlukan bandwidth karena langsung diolah oleh browser.**

Berikut akan diberikan contoh verifikasi form dengan menggunakan PHP.

Verifikasi *form* dapat dibuat dengan PHP *(server side script)* dan dapat juga dibuat dengan *client side script* seperti misalnya JavaScript. Jika verifikasi *form* tidak memerlukan data-data dari server sebagai pembanding, maka sebaiknya digunakan ***client side script.***

```
<HTML>
<HEAD>
<TITLE> Verifikasi </TITLE>
</HEAD>
<BODY>
<?
// Verifikasi
$pesan = "Tekan tombol Back browser Anda";
if (empty($_POST['nama'])) {
    echo "Anda belum memasukkan nama Anda <BR>";
    echo "$pesan";
} elseif (empty($_POST['umur']) || ($_POST['umur'] < 0)
|| (!is_numeric($_POST['umur']))) {
    echo "Input umur Anda salah <BR>";
    echo "$pesan";
} elseif (empty($_POST['email']) ||
(!ereg("@",$_POST['email']))) {
    echo "Periksa kembali email Anda <BR>";
    echo "$pesan";
} else {
$nama = $_POST['nama'];
$umur = $_POST['umur'];
$email = $_POST['email'];
Echo "<H1> Hallo $nama </H1>";
?>
<H2> Selamat Datang Di Situs Kami </H2>
<?
Echo "Umur Anda saat ini adalah ";
Echo "$umur tahun <BR>";
Echo "Alamat email Anda adalah ";
Echo "$email";
}
```

```
?>
</BODY>
</HTML>
```

Sekarang coba jalankan file form.htm dengan sengaja melakukan beberapa kesalahan, misalnya tidak mengisi nama, mengisi umur dengan bilangan negatif atau dengan karakter selain bilangan, mengisi alamat email tanpa tanda "@". Setelah Anda menekan tombol submit, maka akan nampak pesan kesalahan sesuai dengan kesalahan yang Anda lakukan. Tetapi jika Anda mengisi form tersebut dengan benar, maka Anda akan menerima ucapan selamat datang.

4.11. *Appendix* (Mengakses Informasi *Client* dan *Server*)

Pada dasarnya untuk melihat keseluruhan informasi dari *client* dan server yang pada saat itu disimpan di sisi *server* sebagai variabel *server,* anda dapat menuliskan skrip berikut:

```
<?php
     phpinfo();
?>
```

Tetapi ada baiknya jika kita mengambil informasi tersebut kemudian menampilkannya dalam halaman web kita dengan format yang selaras dengan halaman web kita. Misalkan kita ingin memberitahu kepada pengunjung situs kita berapa IP Address dari komputer mereka, dan lain sebagainya.

Rasmus Lerdorf

Penemu PHP *(Perl Hypertext Preprocessor)*

Rasmus Lerdorf

Nama Lengkap:
Rasmus Lerdorf

Tempat tanggal lahir:
Qeqertarsuaq, Greenland,
22 November 1968

Kuliah : University of Waterloo
Pekerjaan: Ilmuwan komputer
Situs web http://lerdorf.com

Hasil penemuan:
Penemu PHP

(Sumber: https://id.wikipedia.org/wiki/Rasmus_Lerdorf

Rasmus Lerdorf dilahirkan pada tanggal 22 November 1968 di Qeqertarsuaq, Greenland merupakan seorang programmer dari Denmark/Greenland dan pencipta bahasa pemrograman PHP. Rasmus menulis dua versi pertama dari PHP. Rasmus juga berpartisipasi dalam pengembangan versi-versi PHP selanjutnya yang dikembangkan oleh sebuah kelompok pengembang, termasuk di dalamnya Andi Gutmans dan Zeev Suraski yang selanjutnya mendirikan Zend Technologies. Pada tahun 1993 Rasmus lulus dari Universitas Waterloo. Sejak bulan September 2002, dia bekerja di Yahoo! sebagai seorang Insinyur Arsitek Infrastruktur.

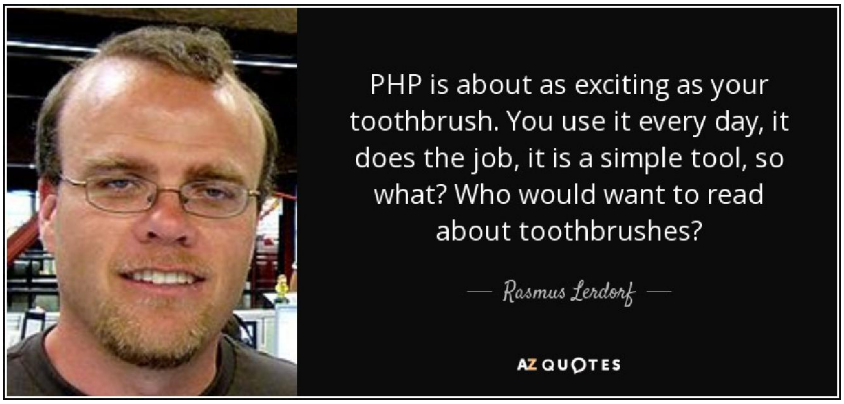
Saya lahir di Pulau Disco di lepas pantai barat Greenland dan menghabiskan tahun-tahun awal saya di Denmark. Saya pindah ke Kanada bersama orang tua saya. Pada tahun 1980 dan ke King City pada tahun 1983 untuk memulai kelas 9 di KCSS. Ketika saya tidak bermain sepak bola atau bola basket, saya bermain-main dengan komputer sekolah. Awalnya mereka adalah Commodore PET, diikuti. Oleh ICON Unisys yang sangat menarik, juga dikenal sebagai berang-berang bionik. Mesin ini menjalankan sistem operasi QNX seperti UNIX.

Dan karena dibangun khusus untuk sekolah di Ontario, tidak ada yang tahu tentang mereka. Tidak ada perangkat lunak untuk mereka, jadi temanku, Ben Eng dan aku, menghabiskan banyak waktu untuk menulis potongan-potongan perangkat lunak untuk mereka. Saya juga akhirnya mengajar Kelas malam komputer pendidikan untuk orang dewasa dan kelas akreditasi guru. Saya pikir ini awal paparan UNIX, jaringan dan pengajaran pengaturan (tata) panggung untuk kepentingan saya selanjutnya.

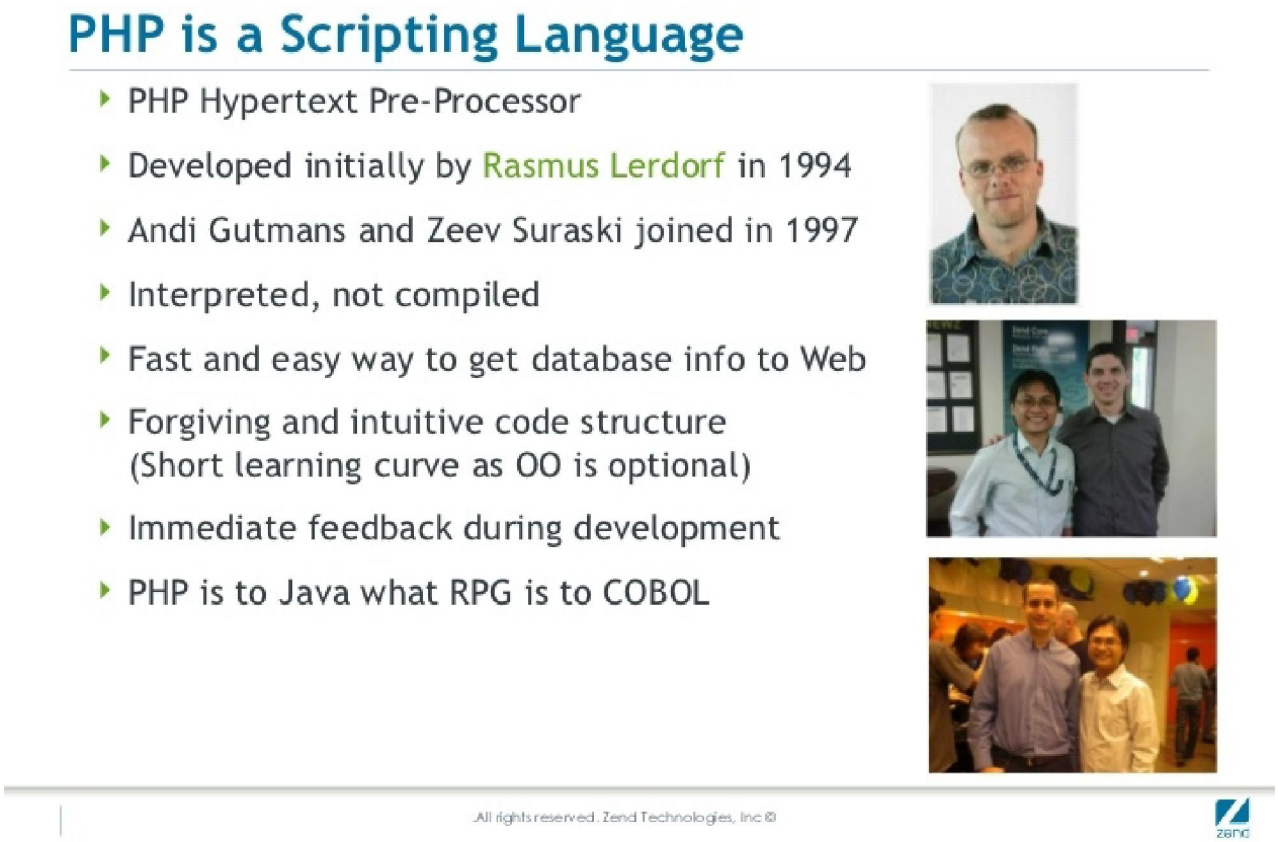
Setelah study di King, saya mendapatkan gelar Teknik Desain Sistem dari Waterloo. Saya menyukai program co-op di sana dan bekerja di seluruh Kanada. Saya mengakhiri dua persyaratan kerja terakhir saya untuk sebuah perusahaan bernama Nutec di Porto Alegre, Brasil. Saya tinggal dengan software ini, perusahaan setelah saya lulus, tapi mereka memindahkan saya ke Silicon Valley pada tahun 1993. Pada saat itu web baru saja mulai muncul dan saya merasa ketagihan dengan keberadaannya. Ketika Nutec tidak bergerak cukup cepat menuju Web, saya berhenti dan kembali ke Toronto untuk melakukan konsultasi kerja. Sebagai bagian dari pekerjaan konsultasi ini, saya mengumpulkan versi pertama bahasa *scripting web* yang sekarang dikenal sebagai PHP dan saya Juga melakukan beberapa pekerjaan awal pada *web server* Apache.

Proyek web besar pertama saya adalah proyek UTorDial di U T. Ini adalah sistem yang memungkinkan siswa dan fakultas untuk *online* melalui Kolam *modem dialup*. Proyek-proyek tersebut secara bertahap bertambah besar setelah itu. Saya mengerjakan proyek untuk Bell Global Solutions di Toronto dan kemudian pindah ke Research Triangle Park di North Carolina untuk bergabung dengan tim WebSphere IBM. Ledakan dot-com telah terjadi Tepat saat itu, dan seperti banyak orang lain, saya berakhir di Silicon Valley untuk memulai sebuah *startup*.

Tetapi satu konstan sepanjang berbagai pekerjaan ini adalah bahwa saya menghabiskan semua waktu luang saya, dan sebagian besar waktu kerja saya juga di proyek PHP yang saat itu telah berkembang menjadi proyek *open source* besar dengan ratusan kontributor. Saya bergabung dengan Yahoo! Pada tahun 2002 untuk membantu mereka membakukan *platform* penyajian web mereka di PHP. Membuat PHP bekerja pada tujuan web terbesar di dunia ini adalah tantangan yang tidak bisa saya lewatkan.



Saat ini, pekerjaan saya di PHP dan Yahoo! telah membawa saya ke ratusan konferensi, bertemunya ribuan orang di seluruh dunia. Saya cukup beruntung untuk mengajar PHP ke Douglas Adams di London dan Arthur C. Clarke di Sri Lanka sebelum mereka meninggal. Aneh rasanya, saya sebenarnya tidak menyukai pemrograman. Semangat saya adalah untuk mengembangkan sistem yang menyederhanakan masalah rumit. Setelah 15 tahun mengerjakan masalah web, hal itu tetap menarik minat saya, namun masalah selanjutnya yang harus dipecahkan adalah melarikan diri kecepatan kehidupan yang padat di Silicon Valley bersama istri saya Christine dan Carl putra saya yang berusia 6 tahun. Saya tidak tahu di mana kita akan berakhir, tapi saya yakin saya bisa menemukan beberapa masalah menarik untuk dipecahkan di sana.



Rangkuman

1. Pengenal *(identifier)* banyak digunakan dalam program untuk memberi nama *variable*, fungsi atau kelas suatu objek
2. Fungsi adalah blok kode yang ditujukkan untuk melaksanakan suatu tugas tertentu. Fungsi digunakan berkali-kali pada semua bagian yang dibutuhkan dalam skrip tersebut.
3. *For statement* terdiri atas 3 bagian yang dipisahkan oleh tanda semicolon (;) yang secara berurutan merupakan dapat menjalankan sebuah fungsi dinamis.
4. Operator adalah symbol yang digunakan dalam program untuk melakukan operasi, misalnya penjumlahan atau perkalian, pembandingan kesamaan dua buah nilai atau bahkan memberikan nilai ke variable
5. Pengenal *(identifier)* banyak digunakan dalam program untuk memberi nama *variable,* fungsi atau kelas suatu objek

Daftar Istilah

1. DHTML *(Dynamic HTML)*
2. DNS [*Domain Name System* (Layanan)]
3. Nama Domain
4. Frames
5. FTP *(File Transfer Protocol)*
6. Tautan *(Hyperlink)*
7. Meta Tag
8. RSS *(Really Simple Syndication)*
9. SSI *(Server-Side Includes)*
10. Script
11. Scripting Bahasa AJAX *(Asynchronous JavaScript dan XML)*
12. ASP *(Active Server Pages)*
13. ASP.NET
14. Bandwidth
15. JavaScript
16. URI *(Uniform Resource Identifier)*
17. URL *(Uniform Resource Locator)*
18. W3C *(World Wide Web Consortium)*
19. XHTML *(Extensible HyperText Markup Language)*
20. XML *(Extensible Markup Language)*

Latihan Soal

1. Jelaskan pengertian Server-side scripting serta keunngulannya?
2. Sebutkan dan jelaskan tipe dasar pada PHP!
3. Sebutkan jenis – jenis Operator!
4. Sebutkan fungsi yang terkait dengan operasi untuk memeriksa tipe variabel!
5. Sebutkan dan jelaskan fungsi yang berkaitan dengan opeasi array dan string!
6. Jelaskan pengertian dari Include dan Require?
7. Jelaskan kegunaan dari metode GET dan Post?

UNIVERSITAS NAROTAMA

Pemahaman Tentang MySQL

Tujuan Instruksional Umum

Setelah mempelajari bab ini, Mahasiswa dipastikan dapat:

1. Menjelaskan Pengertian MySQL
2. Menjelaskan Langkah – Langkah Menggunakan MySQL
3. Menjelaskan Perintah - Perintah Dasar MySQL

BAB

5

Pemahaman Tentang MySQL

5.1. Pengertian SQL

MySQL (bisa dibaca dengan **mai-es-ki-el** atau bisa juga **mai-se-kuel**, terserah Anda) adalah suatu perangkat lunak database relasi (bahasa kerennya ***Relational Database Management System*** atau **RDBMS**), seperti halnya **ORACLE, Postgresql, MS SQL**, dan sebagainya.

Jangan disalah-artikan MySQL dengan SQL. **SQL** (singkatan dari ***Structured Query Language***) sendiri didefinisikan sebagai suatu sintaks perintah-perintah tertentu atau bahasa (program) yang digunakan untuk mengelola suatu *database.*

Jadi, MySQL dan SQL adalah dua 'mahluk' yang berbeda. Mudahnya, MySQL adalah *software*nya, dan SQL adalah bahasa perintahnya. Jelas kan?

5.2. Menjalankan SQL Pertama Kali

Paket PHPTriad biasanya di-*install* pada direktori atau *folder* **C:\Apache**. Karena PHPTriad merupakan kumpulan paket dari **Apache (Versi 1.3.23), MySQL (versi 3.23.48) dan PHP (versi 4.1.1)**, sehingga masing-masing software tersebut akan disimpan pada direktori khusus. Untuk MySQL ada pada direktori **C:\apache\mysql**. Sedangkan untuk *file-file* eksekusinya (*file* biner) ada di direktori **C:\apache\mysql\bin.** Pada sistem *prompt* masuklah ke direktori **mysql\bin**

```
C:\>CD apache\mysql\bin <Enter>
```

Jalankan dahulu program MySQL Server atau **MySQL Daemon** — disingkat **mysqld** (pada saat ini komputer kita bertindak seolah-olah sebagai sebuah *server*).

```
C:\apache\mysql\bin>mysqld –console <enter>
```

Bila perintah ini berjalan dengan baik, maka pada layar akan ditampilkan pesan:

```
mysqld: ready for connections
```

Perhatikan pada *'connections'* dengan bentuk jamak, dengan demikian komputer kita tersebut telah menjalankan fungsinya sebagai *mysql server* dan dapat melayani lebih dari satu koneksi.

Jangan tutup jendela layar tersebut! Jika Anda menutup jendela tersebut, maka program MySQL *server* akan dihentikan juga oleh sistem. Kemudian, bukalah sebuah jendela *system prompt* (atau ***DOS prompt***) yang baru. Dan masuklah ke direktori **c:\apache\mysql\bin**

```
C:\>cd apache\mysq\bin <enter>
C:\apache\mysql\bin>
```

Kemudian jalankan program MySQL cli*ent* dengan mengetikkan mysql pada sistem prompt.

```
c:\apache\mysql\bin\> mysql <enter>
```

Bila perintah tersebut berhasil, maka pada layar akan tampil ucapan selamat datang dari mysql:

```
Welcome to the MySQL monitor. Commands end with ;
or \g.
Your MySQL connection id is xx to server version:
3.23.47-nt.

Type 'help;' or '\h' for help. Type '\c' to clear
the buffer.
mysql>
```

SELAMAT! Anda sekarang sudah berhasil menjalankan MySQL dan berhasil masuk kedalam sistem MySQL. Pesan tersebut berisi ucapan selamat datang, pemberitahuan tentang penulisan perintah MySQL yang **(tidak) SELALU** diakhiri dengan simbol **;** Versi MySQL yang kita gunakan (3.23.47-nt), dan *connection id* yang berupa nomor urut kita masuk ke *server.*

5.3. Hak Akses

Yang perlu kita perhatikan adalah: **Kenapa sedemikian mudahnya kita mengakses MySQL (cukup mengetikkan 'mysql')? Kok tidak ada pertanyaan seperti *username* dan *login*?**

Memang inilah yang harus diwaspadai. Pertama kali MySQL dijalankan dia akan membolehkan siapapun untuk masuk ke dalam sistem. Ini **SANGAT BERBAHAYA SEKALI!** Sekarang yang harus kita lakukan pertama kali adalah **PEMBATASAN HAK AKSES**.

Mari kita lihat apa saja yang sudah ada di MySQL. Masuklah ke dalam sistem MySQL dengan mengetikkan **mysql** pada sistem prompt. Jangan lupa, sebelumnya kita harus sudah mengaktifkan MySQL *server*nya dulu

Pertama kali MySQL dijalankan dia akan membolehkan siapapun untuk masuk ke dalam sistem. Ini **SANGAT BERBAHAYA SEKALI!** Sekarang yang harus kita lakukan pertama kali adalah **PEMBATASAN HAK AKSES**.

```
C:\apache\mysql\bin>mysql <enter>
Welcome to the MySQL monitor. Commands end with ;
or \g.
Your MySQL connection id is xx to server version:
3.23.47-nt.
Type 'help;' or '\h' for help. Type '\c' to clear
the buffer.
mysql>_
```

Pada MySQL prompt ketikkan sebagai berikut:

```
mysql>SHOW DATABASES ;
+------------+
| Database   |
+------------+
| mysql      |
| test       |
+------------+
2 rows in set (0.00 sec)
```

MySQL berarti merujuk ke perangkat lunak/*software database* MySQL, sedangkan mysql (dengan huruf kecil) berarti merujuk ke sebuah nama *database.*

Perintah diatas menampilkan *database* apa saja yang sudah ada sejak MySQL pertama kali diinstal. Disini ada 2 *database,* yaitu *database* **mysql** dan **test**. *Database* **mysql** berisi data-data tentang *user, password, nama host,* dan sebagainya (nanti kita lihat isinya yach..sabar..). Dan *database* **test** merupakan *database* untuk latihan bagi siapa saja yang mau. Kalau tidak diperlukan, kita bisa hapus *database* **test** ini.

Catatan:

MySQL berarti merujuk ke perangkat lunak/*software database* MySQL, sedangkan mysql (dengan huruf kecil) berarti merujuk ke sebuah nama *database.*

Sekarang kita lihat apa saja isi dari database **mysql** tersebut Aktifkan database **mysql**:

```
mysql>USE mysql ;
Database changed
```

Kita sudah mengaktifkan *database* MySQL dan sistem memberi respon dengan pesan *'Database changed'*.

Tampilkan isi *database* MYSQL:

```
mysql>SHOW TABLES ;
+---------------------+
|  Tables_in_mysql    |
+---------------------+
| columns_priv        |
| db                  |
| func                |
| host                |
| tables_priv         |
| user                |
+---------------------+
6 rows in set (0.00 sec)
```

Ada 6 tabel yang dapat dijelaskan kegunaannya sebagai berikut:

Nama Tabel	Kegunaan
columns_priv	Data hak akses *user (user privileges)* pada kolom-kolom tertentu di dalam suatu tabel.
db	Nama *database* yang bisa diakses oleh user tertentu.
func	
host	Nama-nama *host* yang diijinkan mengakses MYSQL
tables_priv	Data hak akses *user (user privileges)* pada tabel-tabel tertentu di dalam suatu *database.*
user	Nama-nama pengguna *(user)* yang diijinkan untuk mengakses MYSQL.

Yang akan kita lihat dahulu adalah isi **tabel *user***. Tuliskan perintah dibawah ini pada MySQL prompt:

```
mysql>DESCRIBE user;
```

atau:

```
mysql> SHOW COLUMNS FROM user ;
```

Perintah tersebut untuk menampilkan kolom-kolom (*field*) apa saja yang ada di dalam sebuah tabel. Hasilnya adalah sebagai berikut:

```
+----------------+-----------------+------+-----+---------+-------+
| Field          | Type            | Null | Key | Default | Extra |
+----------------+-----------------+------+-----+---------+-------+
| Host           | char(60) binary |      | Pri |         |       |
| User           | char(16) binary |      | Pri |         |       |
| Password       | char(16) binary |      |     |         |       |
| Select_priv    | enum('N','Y')   |      |     | N       |       |
| Insert_priv    | enum('N','Y')   |      |     | N       |       |
| Update_priv    | enum('N','Y')   |      |     | N       |       |
| Delete_priv    | enum('N','Y')   |      |     | N       |       |
| Create_priv    | enum('N','Y')   |      |     | N       |       |
| Drop_priv      | enum('N','Y')   |      |     | N       |       |
| Reload_priv    | enum('N','Y')   |      |     | N       |       |
| Shutdown_priv  | enum('N','Y')   |      |     | N       |       |
| Process_priv   | enum('N','Y')   |      |     | N       |       |
| File_priv      | enum('N','Y')   |      |     | N       |       |
| Grant_priv     | enum('N','Y')   |      |     | N       |       |
| References_priv| enum('N','Y')   |      |     | N       |       |
| Index_priv     | enum('N','Y')   |      |     | N       |       |
| Alter_priv     | enum('N','Y')   |      |     | N       |       |
+----------------+-----------------+------+-----+---------+-------+
   17 rows in set (0.00 sec)
```

Kita lihat sepintas bahwa tabel ***user*** ini mencakup *nama user,nama host* dari mana si user mengakses mysql, *password* dari user yang bersangkutan, dan sisanya adalah jenis-jenis hak akses (*user privileges*) yang dimiliki *user* tersebut. Mengenai jenis akses akan kita bahas nanti pada bab tersendiri yach... karena ceritanya memang agak panjang dan teoritis...

Sedangkan mulai MySQL **versi 4.0.xx** (atau yang lebih baru), ada beberapa perubahan yang terjadi pada **tabel *user***. Mari kita lihat apa saja perbedaannya dibanding dengan MySQL versi yang lebih lama:

```
mysql> desc user ;
```

Field	Type	Null	Key	Default	Extra
Host	varchar(60)binary		PRI		
User	varchar(16)binary		PRI		
password	varchar(16)			N	
Select_priv	enum('N','Y')			N	
Insert_priv	enum('N','Y')			N	
Update_priv	enum('N','Y')			N	
Delete_priv	enum('N','Y')			N	
Create_priv	enum('N','Y')			N	
Drop_priv	enum('N','Y')			N	
Reload_priv	enum('N','Y')			N	
Shutdown_priv	enum('N','Y')			N	
Process_priv	enum('N','Y')			N	
File_priv	enum('N','Y')			N	
Grant_priv	enum('N','Y')			N	
References_priv	enum('N','Y')			N	
Index_priv	enum('N','Y')			N	
Alter_priv	enum('N','Y')			N	
Show_db_priv	enum('N','Y')			N	
Super_priv	enum('N','Y')			N	
Create_tmp_table_priv	enum('N','Y')			N	
Lock_tables_priv	enum('N','Y')			N	
Execute_priv	enum('N','Y')			N	
Repl_slave_priv	enum('N','Y')			N	
ssl_type	enum ('','ANY','X509', 'SPECIFIED')				
ssl_cipher	blob				
x509_issuer	blob				
x509_subject	blob				
max_questions	int(11)			0	
max_updates	int(11) unsigned			0	
max_connections	int(11) unsigned			0	

```
31 rows in set (0.29 sec)
```

Terlihat ada penambahan **14 baris *privileges* baru**, seperti *super_priv, max_connections*, dan sebagainya. Tetapi untuk sementara ini kita tidak akan membahasnya dulu secara detail.

Sekarang kita kembali ke bahasan sebelumnya yach... Untuk saat ini kita perlukan hanya 3 kolom pertama saja, yaitu kolom (**field**) **host, user** dan **password**:

```
mysql> SELECT user, host, password FROM user ;
+-----+-----------+-------------+
| user | host      | password    |
+-----+-----------+-------------+
| root | localhost |             |
| root |  %        |             |
|      | localhost |             |
|      |  %        |             |
+-----+-----------+-------------+
4 rows in set (0.00 sec)
```

Nah, dari sini baru ketahuan kenapa bisa terjadi siapa saja yang **nyelonong** ke dalam MySQL kita. Perhatikan deh, untuk *user* **root** (ada 2 baris) ***sama sekali tidak memiliki password apapun***. Dan kemudian, ada user yang 'kosong' (ada 2 baris, tidak bernama, tidak berwujud) dan anehnya lagi tidak ber-password. Kesemuanya ini **SANGAT SANGAT BERBAHAYA!** Untuk itu harus diambil tindakan tegas! (hehehe... seperti aparat keamanan aja yach gaya bicaranya...:))

Perhatikan deh, untuk *user root* (ada 2 baris) ***sama sekali tidak memiliki password apapun***. Dan kemudian, ada user yang 'kosong' (ada 2 baris, tidak bernama, tidak berwujud) dan anehnya lagi tidak ber-password. Kesemuanya ini **SANGAT BERBAHAYA!**

Tindakan keamanan pertama: Hapus user yang tidak berwujud itu:

```
mysql> DELETE FROM user WHERE user=' ' ;
Query OK, 2 rows affected (0.00 sec)
```

Keliatannya sudah OK nih. Tapi coba kita periksa lagi:

```
mysql> SELECT user, host, password FROM user ;
+------+-----------+-------------+
| user |    host   |   password  |
+------+-----------+-------------+
| root | localhost |             |
| root |  %        |             |
+------+-----------+-------------+
2 rows in set (0.00 sec)
```

Ternyata emang OK tuch. *User* yang tidak berwujud itu telah lenyap. Tugas selanjutnya, atau **TINDAKAN PENGAMANAN KEDUA adalah MEMBERIKAN PASSWORD KEPADA ROOT**.

5.4. Pembuatan Password untuk Root

Anda sudah *tea break*? Gimana, lebih segar kan sekarang? Sampai dimana pembahasan kita terakhir? Oh ya, kita sudah berhasil menginstall MySQL, menjalankan MySQL, tapi kemudian kita heran kok begitu mudah kita *'nyelonong'* masuk ke MySQL (tanpa harus *login,* tanpa harus memberikan *password*). Lalu kita tahu sumber penyakitnya ada dimana.

Tahap pengamanan pertama sudah kita lakukan, yaitu menghapus *user* 'tanpa wujud'. Sekarang kita lakukan tahap pengamanan kedua: **PEMBUATAN PASSWORD untuk ROOT.**

Kita periksa kembali hasil yang lalu:

```
<mysql> SELECT user, host, password FROM user ;
+——————+————————+————————+
|  user  |    host     |  password  |
+——————+————————+————————+
|  root  |  localhost  |            |
|  root  |      %      |            |
+——————+————————+————————+
2 rows in set (0.00 sec)
```

Yang akan kita lakukan sekarang adalah mengisi kolom *password* yang kosong tersebut dengan sebuah *password* yang **HANYA ANDA** saja yang tahu. Perintah yang digunakan adalah:

```
UPDATE nama_tabel SET nama_kolom_dengan_kriteria WHERE
kriteria ;
```

Ok, sekarang kita coba dengan perintah sebagai berikut:

```
mysql> UPDATE user SET
password=password('rahasiaanda')
    -> WHERE user='root';
Query OK, 2 rows affected (0.00 sec)
```

Setelah itu dilanjutkan dengan mengetikkan perintah:

```
mysql> FLUSH PRIVILEGES ;
Query OK, 0 row affected
```

Perintah ***FLUSH PRIVILEGES*** adalah suatu perintah untuk mengaktifkan perubahan-perubahan yang terjadi pada *user,* seperti hak akses, penggantian *password* pada *user,* dan sebagainya. Mengenai perintah ini akan dibahas lebih lanjut nanti pada bagian **Manajemen *User***. Perintah *FLUSH PRIVILEGES* ini hukumnya **WAJIB** dilaksanakan setelah Anda melakukan perubahan (apapun juga) secara langsung ke dalam tabel ***user*** atau ke dalam *database* **MYSQL**.

Keliatannya OK tuch... kita periksa yach:

```
mysql>  SELECT user, host, password FROM user ;
+——————+————————+——————————————+
|  user  |    host     |      password      |
+——————+————————+——————————————+
|  root  |  localhost  |  xxxxxxxxxxx       |
|  root  |      %      |  xxxxxxxxxxx       |
+——————+————————+——————————————+
2 rows in set (0.00 sec)
```

Pada kolom password kita lihat sudah berubah isinya menjadi 'xxxxxxxxx' (sengaja saya sembunyikan....hehehe). Bentuk ini merupakan hasil pengacakan yang dilakukan oleh MySQL dengan menggunakan fungsi ***password***().

Jadi, Anda tidak bisa menebak-nebak apa bentuk asli dari karakter-karakter tersebut. Nah, untuk keamanan Anda, HANYA ANDA saja yang tahu apa *password* yang digunakan tersebut. Hanya Anda saja yang INGAT apa *password* tersebut. Tapi bagaimana kalau kita lupa? Nah, sebagai *root* Anda masih bisa untuk mengutak-ngatik *password*. Tapi akan kita bahas itu nanti di bab-bab berikutnya. Sekarang keluar dulu dari sistem MySQL:

... untuk keamanan Anda, HANYA ANDA saja yang tahu apa *password* yang digunakan tersebut. Hanya Anda saja yang INGAT apa *password* tersebut. Tapi bagaimana kalau kita lupa? Nah, sebagai *root* Anda masih bisa untuk mengutak-ngatik *password*.

```
mysql> quit bye
```

Kita masuk lagi ke mysql:

```
c:\apache\mysql\bin> mysql <enter>
Error 1045: Access denied for user
'ODBC@localhost' (using password: NO)
```

Lho kok error? Nah itu tandanya perintah2 yang kita lakukan sebelumnya sudah mulai bereaksi Kita kembali masuk lagi dengan bentuk perintah yang sedikit berbeda:

```
mysql -u nama_user -h nama_host -p
```

Sekarang ikuti dulu perintah dibawah ini:

```
c:\apache\mysql\bin> mysql -u root -h localhost -p
Enter password:************
```

Ketikkan password Anda yang tadi (rahasia Anda kan?)

```
Welcome to the MySQL monitor. Commands end with ;
or \g.
Your MySQL connection id is xx to server version:
3.23.47-nt.

Type 'help;' or '\h' for help. Type '\c' to clear
the buffer.
mysql> _
```

Sekarang semuanya sudah jelaskan? Tidak ada lagi seorang pun yang tidak terdaftar dapat nyelonong ke sistem kita.

5.5. Localhost atau %?

Masih ingat dengan penggunaan ***localhost*** dan %? Tidak? OK. Sekarang kita lihat lagi deh contoh-contoh sebelumnya:

mysql> SELECT user, host, password FROM user ;

user	host	password
root	localhost	xxxxxxxxxxxx
root	%	xxxxxxxxxxxx

```
2 rows in set (0.00 sec)
```

LOCALHOST berarti tempat dimana program MySQL *server* berada. Kalau Anda menyimpan program-program MySQL di komputer Anda saat ini dan Anda langsung mengaksesnya dari sana, itu berarti komputer Anda adalah *localhost*.

Nah, tampilan diatas adalah isi dari tabel ***user***. Kita cuma punya satu user saat ini, yaitu **root** (yang tidak lain adalah Anda sendiri...). *Password*nya..ada tuch (tapi diacak). *Host,* bisa diartikan nama server dimana kita mengakses MySQL. Isi kolom *host* ada dua, yaitu **localhost** dan **%.**

LOCALHOST berarti tempat dimana program MySQL *server* berada. Kalau Anda menyimpan program-program MySQL di komputer Anda saat ini dan Anda langsung mengaksesnya dari sana, itu berarti komputer Anda adalah *localhost*.

Tanda % itu maksudnya Anda melakukan pengaksesan MySQL **dari mana saja** (tidak selalu dari localhost). Misalkan, Anda sering bepergian ke luar negeri,sedangkan Anda harus rutin untuk mengakses server MySQL tersebut, maka Anda harus diberikan tanda **%** pada nama hostnya.

Konsep persen (**%**) ini agak membingungkan pada awalnya. Karena, sebenarnya kita bisa saja mengakses server MySQL kita dengan program **TELNET** yang berarti seolah-olah langsung mengakses server dari tempat kita duduk, berarti konsepnya akan sama dengan *localhost.*

Bagaimana cara kita untuk membuat user baru, dan menentukan lokasi dimana dia bisa mengakses akan kita bahas pada bab-bab berikutnya: **Manajemen *User* pada MySQL**.

5.6. Perintah-perintah Dasar MySQL

Pada tutorial yang lalu, sebenarnya kita telah menggunakan beberapa perintah dasar MySQL, seperti ***USE, SELECT, UPDATE,*** dan sebagainya. Dan kita akan bahas lagi perintah-perintah tersebut.

Catatan: Untuk menggantikan istilah sistem prompt, digunakan simbol **#** atau **$**

5.7. Menjalankan MySQL

Bentuk Umum
mysql nama_database -u nama_user **-h** nama_host **-p** kata_sandi_atau_password

Contoh:

```
# mysql latih1db -u arbie -h localhost -p <enter>
Enter password:
```

Hal diatas dapat dijalankan kalau Anda sudah memiliki sebuah database dengan nama **latih1db**. Bila Anda memang mengakses komputer yang berada satu tempat dengan server MySQL, kata 'localhost' bisa dihilangkan menjadi:

```
# mysql -u arbie -p <enter>
Enter password:
```

Bila Anda sebagai root pada komputer tersebut, maka nama *'root'* juga bisa dihilangkan menjadi:

```
# mysql -p <enter>
Enter password:
```

Tetapi, Anda **jangan sekali-kali** mengetikkan kata sandi langsung setelah perintah **-p**, seperti:

```
# mysql -pkata_sandi <enter>
```

Ini **SANGAT BERBAHAYA SEKALI!** Kemungkinannya sangat besar bagi orang lain untuk 'mencuri pandang' pada *password* yang diketikkan tersebut. Terutama bagi mereka yang menggunakan Unix/Linux, perintah ***ps*** dapat dilakukan oleh siapa saja yang bisa memonitoring proses apa yang sedang dilakukan oleh *user* lain. Jadi, tetaplah pada bentuk yang lebih aman, yaitu:

```
mysql nama_database -u nama_user -h nama_host -p
<enter>
```

Catatan: (terutama bagi non-windows)

Bagaimana kalau program MySQL Server berada pada komputer yang berbeda dengan komputer kita mengaksesnya? Sebenarnya sederhana saja, ganti saja kata *localhost* dengan nama server atau alamat server dimana MySQL tersebut berada.

Contoh:

```
# mysql latih1db -u arbie -h> 202.155.10.10 -p
<enter>
Enter password:
```

Atau:

```
# mysql latih1db -u arbie -h www.arbiedesign.com -
p <enter>
Enter password:
```

5.8. Keluar Dari MySQL

Bentuk Umum
Quit atau Exit

Perintah **Quit** atau **Exit** merupakan perintah yang tidak perlu diakhiri tanda **;** sebagai akhiran perintah.

Contoh:

```
mysql> quit
atau
mysql> exit
```

5.9. Memilih *Database*

Bentuk Umum
USE nama_database

Contoh:

```
mysql> USE test ;
mysql> USE mysql ;
```

Atau dengan cara lain, Anda bisa langsung mengaktifkan *database* mulai dari saat Anda *login* ke MySQL, dengan cara:

```
# mysql test -u nama_user -h localhost -p <enter>
# mysql mysql -u nama_user -h localhost -p <enter>
```

Dengan perintah diatas, begitu Anda masuk ke MYSQL maka akan langsung memilih dan mengaktifkan *database* **test** atau *database* **MYSQL**.

5.10. Melihat Isi *Database*

Bentuk Umum
SHOW TABLES ;

Bila di dalam database tersebut telah berisi tabel, maka nama-nama tabel akan ditampilkan di layar. Tapi, bila *database*nya masih kosong, maka sistem akan menampilkan pesan:

```
Empty set (0.01 sec)
```

5.11. Melihat Struktur *Table*

Bentuk Umum
SHOW COLUMNS FROM nama_tabel **; atauDESCRIBE** nama_tabel **;**

Contoh:

```
mysql> SHOW COLUMNS FROM user;
mysql> SHOW COLUMNS FROM db ;
```

atau

```
mysql> DESCRIBE user ;
mysql> DESCRIBE db ;
```

5.12. Membuat *Database* Baru

Bentuk Umum
CREATE DATABASE nama_database ;

Contoh:

```
mysql> CREATE DATABASE latih1db ;
mysql> CREATE DATABASE karyawandb ;
mysql> CREATE DATABASE personalia ;
```

5.13. Menghapus Database

Bentuk Umum
DROP DATABASE nama_database ;

Menghapus dalam MySQL diberi nama 'DROP'. Hati-hati dengan perintah ini, karena sistem MySQL tidak akan memberi peringatan atau konfirmasi dahulu kepada kita untuk melanjutkan proses atau membatalkan proses. Begitu perintah ini dilaksanakan, seluruh isi database (berikut semua tabel-tabel yang ada didalamnya dan semua recordnya) akan dihapus dan tidak bisa dibatalkan.

Contoh:

```
mysql> DROP DATABASE latih1db ;
mysql> -DROP -DATABASE karyawandb ;
mysql> -DROP -DATABASE personalia ;
```

5.14. Membuat Tabel Baru

Bentuk Umum
CREATE TABLE nama_tabel (nama_kolom1 spek_kolom1, ..., nama_kolom_berikutnya spek_kolom_berikutnya) ;

Sebelumnya, kita ulangi pembuatan database baru dan kita beri nama personaliadb. Setelah database baru tersebut ada, kita aktifkan dengan perintah use.

Contoh:

```
mysql> CREATE DATABASE personaliadb ;
Query OK, 1 row affected (0.02 sec)
mysql> USE personaliadb ;
Database changed
```

Sekarang kita coba bikin sebuah tabel yang kita beri nama tabel karyawan.

```
mysql> CREATE TABLE karyawan ;
ERROR 1113: A table must have at least 1 column.
```

Kenapa error? Ya, karena untuk membuat sebuah tabel kita harus mempunyai minimal 1 kolom (atau 1 field) saja. Kita coba lagi sebagai berikut:

```
mysql> CREATE TABLE karyawan
    -> (noid INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    ->  nama VARCHAR(35) NOT NULL)
    ->  ;
Query OK, 0 rows affected (0.01 sec)
```

Kolom noid, merupakan nomor identitas karyawan, yang bersifat INT (bilangan integer atau bilangan bulat), UNSIGNED maksudnya selalu bernilai positif (tidak ada tanda - atau negatif), AUTO_INCREMENT berarti nilai selanjutnya merupakan nilai sebelumnya ditambah nilai satu, PRIMARY KEY berarti kolom noid merupakan kolom indeks. Dalam satu tabel hanya boleh ada satu PRIMARY KEY.

Kemudian pada kolom nama, diberi lebar sebanyak 35 karakter (bisa menampung nama sepanjang maksimum 35 karakter, kurang dari 35 karakter juga OK-OK aja), Untuk meyakinkan bahwa tabel karyawan sudah berhasil kita buat, silakan diperiksa dengan perintah di bawah ini:

```
mysql> SHOW TABLES ;
+--------------------------+
| Tables_in_personaliadb |
+--------------------------+
|         karyawan         |
+--------------------------+
1 row in set (0.00 sec)
```

5.15. Melihat Struktur Table

Bentuk Umum
SHOW COLUMNS FROM nama_tabel ; atauDESCRIBE nama_tabel ;

Contoh:

```
mysql> DESCRIBE karyawan ;
```

Atau

```
mysql> SHOW COLUMNS FROM karyawan ;
```

Dan hasilnya akan sama saja:

```
+-------+-----------------+------+-----+---------+----------------+
| Field |      Type       | Null | Key | Default |     Extra      |
+-------+-----------------+------+-----+---------+----------------+
| noid  | int(10) unsigned |      | PRI |  NULL   | auto_increment |
| nama  |    varchar(35)   |      |     |         |                |
+-------+-----------------+------+-----+---------+----------------+
 2 rows in set (0.02 sec)
```

5.16. Menghapus Tabel

Bentuk Umum
DROP TABLES nama_tabel ;

Untuk menghapus tabel, bentuk perintahnya mirip dengan menghapus database. Kehati-hatian (extra) pun sangat diperlukan disini. Karena sistem MySQL tidak akan menanyakan konfirmasi ulang kepada Anda untuk menghapus maupun membatalkan penghapusan tabel. Tabel beserta dengan seluruh isinya akan dihapus tanpa bisa dibatalkan (undelete atau undo kalau di Windows).

Sebagai contoh kita buat dulu tabel lain, namakan saja tabel contoh1:

```
mysql>  CREATE TABLE contoh1
    ->  (noid INT )
    ->  ;
Query OK, 0 rows affected (0.00 sec)
```

Kemudian kita hapus:

```
mysql>  DROP TABLE contoh1 ;
Query OK, 0 rows affected (0.00 sec)
```

Kita periksa lagi:

```
mysql>  SHOW TABLES ;
Empty set (0.01 sec)
```

5.17. Perintah-perintah Dasar MySQL

Sebelumnya kita buat dahulu tabel karyawan (ini bagi mereka yang telah menghapus tabel karyawan sebelumnya, dan sekarang kita buat lagi yang lebih lengkap kolomnya) sebagai berikut:

```
mysql> CREATE TABLE karyawan
   -> (noid INT UNSIGNED AUTO_INCREMENT PRIMARY
KEY,
   -> nama VARCHAR(35) NOT NULL,
   -> kelamin VARCHAR(2),
   -> kota VARCHAR(25),
   -> tgllahir DATE NULL DEFAULT "0000-00-00")
   -> ;
Query OK, 0 rows affected (0.02 sec)
```

5.18. Mengisi Data Baru

Bentuk Umum 1
INSERT INTO nama_tabel (nama_kolom1, nama_kolom2, ...) VALUES (data_kolom1, data_kolom2, ...) ;

Contoh:

```
mysql> INSERT INTO karyawan
    -> (nama, kelamin, kota, tgllahir)
    -> VALUES ("Ahmad Subari","L","Bandung","1977-
01-01")
    -> ;
Query OK, 1 row affected (0.03 sec)
```

5.19. Menampilkan Data

Bentuk Umum
SELECT nama_kolom1, nama_kolom2, ... FROM nama_tabel WHERE kriteria SORT BY kriteria

Contoh:

```
mysql> SELECT noid, nama, kelamin, kota, tgllahir
    -> FROM karyawan
    -> ;
+-------+--------------+----------+----------+-------------+
| noid |    nama      | kelamin |   kota   | tgllahir    |
+-------+--------------+----------+----------+-------------+
|    1 |Ahmad Subari | L        | Bandung | 1977-01-01 |
+-------+--------------+----------+----------+-------------+
1 row in set (0.01 sec)
```

Perintah untuk menampilkan seluruh kolom dapat juga diberikan dengan simbol * (asterik atau bintang), sehingga perintah diatas dapat juga kita tuliskan:

```
mysql> SELECT * FROM karyawan ;
+-------+--------------+----------+----------+-------------+
| noid |    nama      | kelamin |   kota   | tgllahir    |
+-------+--------------+----------+----------+-------------+
|    1 |Ahmad Subari | L        | Bandung | 1977-01-01 |
+-------+--------------+----------+----------+-------------+
1 row in set (0.01 sec)
```

5.20. Mengisi Data Baru (bentuk lainnya)

Bentuk Umum 2
INSERT INTO nama_tabel SET nama_kolom1 = data_kolom1, nama_kolom2 = data_kolom2, ..

Contoh:

```
mysql> INSERT INTO karyawan
    -> SET nama= "Rini Suwandito",
    -> kelamin= "P",
    -> kota= "Jakarta",
    -> tgllahir ="1978-01-02",
    -> ;
Query OK, 1 row affected (0.03 sec)
```

Dibanding dengan cara pertama, maka cara kedua ini lebih jelas posisi data akan diletakkan pada kolom (field) mana saja. Hanya saja, bila jumlah data yang akan kita masukkan banyak, maka dengan cara kedua ini sangat merepotkan sekali. Baiklah, sekarang kita lihat hasil dari perintah kedua ini.

```
mysql> SELECT * FROM karyawan ;
+------+----------------+----------+----------+-------------+
| noid |    nama        | kelamin |   kota   | tgllahir    |
+------+----------------+----------+----------+-------------+
|  1   |Ahmad Subari    | L        | Bandung | 1977-01-01 |
+------+----------------+----------+----------+-------------+
|  2   |Rini Suwandito | P        | Jakarta | 1978-01-02 |
+------+----------------+----------+----------+-------------+
2 row in set (0.01 sec)
```

Ya, semua data sudah masuk ke dalam database. Sekarang kita coba pemasukkan data dengan cara Ketiga.

Bentuk Umum 3
INSERT INTO nama_tabel VALUES(data1, data2, ..., data_ke_n)

Contoh:

```
mysql> INSERT INTO karyawan
    -> VALUES ("Boy Sungkar","L","Cimahi","1979-
03-01")
    -> ;
ERROR 1136: Column count doesn't match value count
at row 1
```

Lho, kok error? Kenapa nich? Perhatikan pada pesan error yang terjadi. Disana dinyatakan bahwa jumlah kolom tidak sesuai dengan jumlah masukan data yang ada.

Kita ingat kalo jumlah kolom ada 5, yaitu noid, nama, kelamin, kota dan tgllahir. Sedangkan data yang kita masukkan untuk 4 kolom saja, yaitu nama, kelamin, kota dan tgllahir. Jadi. Walaupun kolom noid ini sifatnya AUTO_INCREMENT, khusus untuk bentuk ketiga ini dia harus diisi juga dengan nilai DEFAULTNYA yaitu "NULL". Sehingga perintah diatas kita ubah sedikit menjadi...

```
mysql> INSERT INTO karyawan
    -> VALUES ("NULL","Boy
Sungkar","L","Cimahi","1979-03-01")
    -> ;
Query OK, 1 row affected (0.03 sec)

mysql>SELECT * FROM karyawan ;
```

```
+----+-------------+--------+--------+------------+
|noid|    nama     |kelamin |  kota  |  tgllahir  |
+----+-------------+--------+--------+------------+
|  1 | Ahmad Subari|    L   |Bandung |1977-01-01  |
|  2 |Rini Suwandito|   P   |Jakarta |1978-01-02  |
|  3 |Boy Sungkar  |    L   |Cimahi  |1979-03-02  |
+----+-------------+--------+--------+------------+
3 row in set (0.01 sec)
```

5.21. Mengubah Struktur Tabel

Ada saatnya kita sadar kalo ternyata struktur tabel yang pernah kita buat perlu penyempurnaan. Penyempurnaan bisa dalam hal penambahan kolom, pengubahan lebar kolom, atau bisa saja penghapusan kolom dan sebagainya. Apa pun juga yang Anda lakukan pada kolom tersebut tentu akan mempunyai dampak langsung pada data-data yang sudah ada. Nah, sekarang kita perlu menambahkan sebuah kolom baru, yaitu kolom kodepos, pada tabel karyawan yang selama ini kita gunakan.

Kita lihat dulu struktur tabel yang sudah ada.

```
mysql>DESCRIBE karyawan ;
+----------+------------------+------+-----+-------------+----------------+
|Field     | Type             |Null  | Key | Default     | Extra          |
+----------+------------------+------+-----+-------------+----------------+
|noid      |int(10) unsigned  |      | PRI | NULL        | auto_increment |
|nama      |varchar(35)       |      |     |             |                |
|kelamin   |char(2)           | YES  |     | NULL        |                |
|kota      |varchar(25)       | YES  |     | NULL        |                |
|tgllahir  |date              | YES  |     | 0000-00-00  |                |
+----------+------------------+------+-----+-------------+----------------+
5 rows in set (0.02 sec)
```

<>Bentuk Umum
ALTER TABLE nama_tabel jenis_pengubahan

Jenis pengubahan yang dapat dilakukan antara lain penambahan kolom (ADD), penghapusan kolom dan indeks (DROP), penggantian nama kolom sekaligus jenis kolomnya (RENAME), dan pengubahan jenis kolom (MODIFY). Sekarang kita coba untuk menambah kolom baru saja dulu.

```
mysql>ALTER TABLE karyawan ADD kodepos VARCHAR(5);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Lihat hasilnya seperti apa ...

```
mysql>DESCRIBE karyawan ;
```

```
+---------+-----------------+-----+-----+------------+----------------+
|Field    | Type            |Null | Key | Default    | Extra          |
+---------+-----------------+-----+-----+------------+----------------+
|noid     |int(10) unsigned |     | PRI | NULL       | auto_increment |
|nama     |varchar(35)      |     |     |            |                |
|kelamin  |char(2)          | YES |     | NULL       |                |
|kota     |varchar(25)      | YES |     | NULL       |                |
|tgllahir |date             | YES |     | 0000-00-00 |                |
|kodepos  |varchar(5)       | YES |     | NULL       |                |
+---------+-----------------+-----+-----+------------+----------------+
```

```
5 rows in set (0.02 sec)
```

Setelah struktur tabel berhasih diubah, apa pengaruhnya pada data-data yang ada? Dan kemudian bagaimana caranya kita mengisi data kodepos yang belum masuk ke tabel karyawan? Mari kita lihat dulu isi tabel karyawan...

```
mysql>SELECT * FROM karyawan ;
```

```
+------+---------------+---------+---------+------------+---------+
| noid | nama          | kelamin | kota    | tgllahir   | kodepos |
+------+---------------+---------+---------+------------+---------+
| 1    | Ahmad Subari  | L       | Bandung | 1977-01-01 | NULL    |
| 2    | Rini Suwandito| P       | Jakarta | 1978-01-02 | NULL    |
| 3    | Boy Sungkar   | L       | Cimahi  | 1979-03-02 | NULL    |
+------+---------------+---------+---------+------------+---------+
```

```
3 row in set (0.01 sec)
```

Perhatikan bahwa pada kolom kodepos diisi dengan NULL, dimana NULL adalah nilai default suatu kolom bila tidak memiliki nilai apapun juga.

5.22. Mengubah Data

Bentuk Umum
UPDATE nama_tabel SET nama_kolom=pengubahan_data WHERE kriteria

Contoh:

```
mysql> UPDATE karyawan SET kodepos="40111"
    -> WHERE nama="Ahmad Subari"
    -> ;
Query OK, 1 rows affected (0.01sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE karyawan SET kodepos="40222"
```

```
    -> WHERE noid="2"
    -> ;
Query OK, 1 rows affected (0.01sec)
Rows matched: 1 Changed: 1 Warnings: 0
Kita periksa dulu hasilnya..
mysql>SELECT * FROM karyawan ;
```

noid	nama	kelamin	kota	tgllahir	kodepos
1	Ahmad Subari	L	Bandung	1977-01-01	40111
2	Rini Suwandito	P	Jakarta	1978-01-02	40222
3	Boy Sungkar	L	Cimahi	1979-03-02	NULL

```
3 row in set (0.01 sec)
```

Nah, sampai disini kita sudah memasukkan data secara satu per satu. Sekarang bagaimana caranya kalo kita mau memasukkan data secara kolektif. Misalkan ada 10 data lagi yang harus kita masukkan kedalam tabel karyawan.

Salah satu caranya seperti dibawah ini...

```
mysql> INSERT INTO karyawan
    -> VALUES
    -> ("NULL","Dion","L","Bogor","1978-02-
02","40123"),
    -> ("NULL","Celine","P","Jakarta","1976-05-
22","40333"),
    -> ("NULL","Deemaz","L","Jakarta","1977-07-
07","40223"),
    -> ("NULL","Zachryal","L","Bogor","1979-04-
11","40331"),
    -> ("NULL","Erika Sulaeman","P","Cimahi","1980-
09-19","40228"),
    -> ("NULL","Fahrudin","L","Soreang","1978-11-
02","40423"),
    -> ("NULL","Halimun","P","Bandung","1979-08-
11","40123"),
    -> ("NULL","Gempitawati","P","Bandung","1978-08-
22","40334"),
    -> ("NULL","Wawan Suriawan","L","Jakarta","1977-
07-09","40323"),
    -> ("NULL","Taufik Hidayah","L","Bogor","1978-
09-17","40132"),
    -> ("NULL","Dion","L","Bogor","1978-02-
02","40123"),
    -> ;
Query OK, 11 rows affected (0.01sec)
mysql>SELECT * FROM karyawan ;
```

```
+------+----------------+---------+---------+------------+---------+
| noid | nama           | kelamin | kota    | tgllahir   | kodepos |
+------+----------------+---------+---------+------------+---------+
| 1    | Ahmad Subari   | L       | Bandung | 1977-01-01 | 40111   |
| 2    | Rini Suwandito | P       | Jakarta | 1978-01-02 | 40222   |
| 3    | Boy Sungkar    | L       | Cimahi  | 1979-03-02 | NULL    |
| 4    | Dion           | L       | Bogor   | 1978-02-02 | 40123   |
| 5    | Celine         | P       | Jakarta | 1976-05-22 | 40333   |
| 6    | Deemaz         | L       | Jakarta | 1977-07-07 | 40223   |
| 7    | Zachryal       | L       | Bogor   | 1979-04-11 | 40331   |
| 8    | Erika Sulaeman | P       | Cimahi  | 1980-09-19 | 40228   |
| 9    | Fahrudin       | L       | Soreang | 1978-11-02 | 40423   |
| 10   | Halimun        | P       | Bandung | 1979-08-11 | 40123   |
| 11   | Gempitawati    | P       | Bandung | 1978-08-22 | 40334   |
| 12   | Wawan Suriawan | L       | Jakarta | 1977-07-09 | 40323   |
| 13   | Taufik Hidayah | L       | Bogor   | 1978-09-17 | 40132   |
| 14   | Dion           | L       | Bogor   | 1978-02-02 | 40123   |
+------+----------------+---------+---------+------------+---------+

14 row in set (0.01 sec)
```

5.23. Menampilkan data dengan kriteria tertentu

Sebelumnya buatlah dulu table karyawan2 seperti berikut ini:

```
CREATE TABLE karyawan2 (
noid INT(4) UNSIGNED NOT NULL PRIMARY KEY,
nama VARCHAR(50) NOT NULL,
kelamin ENUM("L","P") NOT NULL,
tempatlahir VARCHAR(35),
tgllahir DATE NOT NULL DEFAULT "0000-00-00",
alamat1 VARCHAR(35),
alamat2 VARCHAR(35),
kota VARCHAR(35),
kodepos CHAR(5),
telprmh VARCHAR(20),
ponsel VARCHAR(20)
);
```

Setelah itu isi table tersebut dengan cara yang sama seperti yang diatas. Selama ini untuk menampilkan seluruh data, kita menggunakan perintah SELECT * FROM, yang mana bila datanya sangat banyak dan melebar, tidak akan cukup tampilan layar kita menampilkannya. Untuk itu, ada perintah lainnya yang dapat kita gunakan, yaitu dengan tambahan kriteria tertentu.

Bentuk Umum
SELECT kolom1,kolom2,kolomN FROM nama_tabel WHERE kriteria ORDER BY kriteria

Baiklah, kita mulai dengan cara menampilkan struktur tabel karyawan-karyawan dulu supaya gak lupa apa aja nama-nama kolomnya.

```
mysql> DESCRIBE karyawan2 ;
```

```
+------------+-----------------+-----+-----+------------+----------------+
|Field       | Type            |Null | Key | Default    | Extra          |
+------------+-----------------+-----+-----+------------+----------------+
|noid        |int(10) unsigned |     | PRI | NULL       | auto_increment |
|nama        |varchar(50)      |     |     |            |                |
|kelamin     |enum('L','P')    | YES |     | NULL       |                |
|tempatlahir |varchar(35)      | YES |     | NULL       |                |
|tgllahir    |date             |     |     | 0000-00-00 |                |
|alamat1     |varchar(35)      | YES |     | NULL       |                |
|alamat2     |varchar(35)      | YES |     | NULL       |                |
|kota        |varchar(35)      | YES |     | NULL       |                |
|kodepos     |varchar(35)      | YES |     | NULL       |                |
|telprumah   |varchar(20)      | YES |     | NULL       |                |

|ponsel      |varchar(20)      | YES |     | NULL       |                |
+------------+-----------------+-----+-----+------------+----------------+

11 rows in set (0.02 sec)
```

5.24. Menampilkan data kolom tertentu saja

Sekarang kita coba untuk menampilkan data kolom-kolom tertentu saja

```
mysql> SELECT noid, nama, kelamin, tempatlahir
    -> FROM karyawan2 ;
```

noid	nama	kelamin	tempatlahir
1	Suwanto Dipura	L	Majalengka
2	Mayangsariwati	P	Bogor
3	Purwanto Rojolele	L	Yogyakarta
4	Rita Sariwati	L	Cibinong
5	Suseno Mangun	L	Ciamis
6	Ratna Sari	P	Cianjur
7	Charlie Simanjuntak	L	Medan

```
| 8  | Desi Sukmini       | P | Jakarta        |
| 9  | Thomas Eng         | L | Hongkong       |
| 10 | Ratih Sanggar      | P | Solo           |
| 11 | Achmad Widjaja     | L | Jakarta        |
| 12 | Delima Rahayu      | P | Demak          |
| 13 | Dadang Sukandar    | L | Cianjur        |
| 14 | Winda Sukma        | P | Yogyakarta     |
| 15 | Zaenal Nur         | L | Denpasar       |
| 16 | Noeri Sakti        | P | Jakarta        |
| 17 | Ahmaddin Sudin     | L | Banda Aceh     |
| 18 | Wati Sutinah       | P | Mojokerto      |
| 19 | Wanto Sukarno      | L | Surabaya       |
| 20 | Wawan Wanisar      | L || Bogor         |
| 21 | Rendi Ganteng      | L | Bogor          |
| 22 | Magda Magdalena    | P | Bandar Lampung |
| 23 | Jumanto Dipuro     | L | Tegal          |
| 24 | Deemas Sudirman    | L || Batam         |
| 25 | Zulailah Zulha     | P | Demak          |
+----+--------------------+---+----------------+
25 rows in set (0.01 sec)
```

5.25. Menampilkan data kolom tertentu dengan urutan

```
mysql> SELECT noid, nama, kelamin, tempatlahir
  -> FROM karyawan2
  -> ORDER BY nama ;
+----------+------------------------------+-------------+------------------------+
| noid     | nama                         | kelamin     | tempatlahir            |
+----------+------------------------------+-------------+------------------------+
|       11 | Achmad Widjaja               | L           | Jakarta                |
|       17 | Ahmaddin Sudin               | L           | Banda Aceh             |
|        7 | Charlie Simanjuntak          | L           | Medan                  |
|       13 | Dadang Sukandar              | L           | Cianjur                |
|       24 | Deemas Sudirman              | L           | Batam                  |
|       12 | Delima Rahayu                | P           | Demak                  |
|        8 | Desi Sukmini                 | P           | Jakarta                |
|       23 | Jumanto Dipuro               | L           | Tegal                  |
|       22 | Magda Magdalena              | P           | Bandar Lampung         |
|        2 | Mayangsariwati               | P           | Bogor                  |
|       16 | Noeri Sakti                  | P           | Jakarta                |
|        3 | Purwanto Rojolele            | L           | Yogyakarta             |
|       10 | Ratih Sanggar                | P           | Solo                   |
|        6 | Ratna Sari                   | P           | Cianjur                |
|       21 | Rendi Ganteng                | L           | Bogor                  |
|        4 | Rita Sariwati                | P           | Cibinong               |
|        5 | Suseno Mangun                | L           | Ciamis                 |
|        1 | Suwanto Dipura               | L           | Majalengka             |
```

9	Thomas Eng	L	Hongkong
19	Wanto Sukarno	L	Surabaya
18	Wati Sutinah	P	Mojokerto
20	Wawan Wanisar	L	Bogor
14	Winda Sukma	P	Yogyakarta
15	Zaenal Nur	L	Denpasar
25	Zulailah Zulha	P	Demak

```
25 rows in set (0.01 sec)
```

5. 26. Menampilkan data baris dan kolom tertentu

```
mysql> SELECT noid, nama, kelamin, tempatlahir
    -> FROM karyawan2
    -> WHERE kelamin="P" ;
```

noid	nama	kelamin	tempat lahir
2	Mayangsariwati	P	Bogor
4	Rita Sariwati	P	Cibinong
6	Ratna Sari	P	Cianjur
8	Desi Sukmini	P	Jakarta
10	Ratih Sanggar	P	Solo
12	Delima Rahayu	P	Demak
14	Winda Sukma	P	Yogyakarta
16	Noeri Sakti	P	Jakarta
18	Wati Sutinah	P	Mojokerto
22	Magda Magdalena	P	Bandar Lampung
25	Zulailah Zulha	P	Demak

```
11 rows in set (0.01 sec)
```

5.27. Menampilkan Data Baris dan Kolom Tertentu dengan urutan

```
mysql> SELECT noid, nama, kelamin, tempatlahir
    -> FROM karyawan2
    -> WHERE kelamin="P"
    -> ORDER BY nama ;
```

noid	nama	kelamin	tempatlahir
12	Delima Rahayu	P	Demak
8	Desi Sukmini	P	Jakarta
22	Magda Magdalena	P	Bandar Lampung
2	Mayangsariwati	P	Bogor
16	Noeri Sakti	P	Jakarta
10	Ratih Sanggar	P	Solo
6	Ratna Sari	P	Cianjur

```
|  4 | Rita Sariwati    |  P  | Cibinong      |
| 18 | Wati Sutinah     |  P  | Mojokerto     |
| 14 | Winda Sukma      |  P  | Yogyakarta    |
| 25 | Zulailah Zulha   |  P  | Demak         |
+----+------------------+-----+---------------+
11 rows in set (0.01 sec)
```

5.28. Operator Pembanding dan Operator Logika

Didalam MySQL ada operator pembanding yang sering digunakan seperti:

Operator Pembanding	Keterangan
>	Lebih besar
<	Lebih kecil
>=	Lebih besar atau sama dengan
<=	Lebih kecil atau sama dengan
=	Sama dengan
<>	Tidak sama dengan

Operator Logika	Keterangan
AND atau &&	Dan
OR atau ||	Atau
NOT atau !	Lebih besar atau sama dengan
<=	Lebih kecil atau sama dengan
<>	Tidak sama dengan

Sekarang mari kita coba operator-operator tersebut:

```
mysql> SELECT noid, nama, kelamin, tempatlahir
FROM karyawan2 ;
+------+----------------------+---------+--------------------+
| noid |         nama         | kelamin |    tempatlahir     |
+------+----------------------+---------+--------------------+
|  1   | Suwanto Dipura       |    L    | Majalengka         |
|  2   | Mayangsariwati       |    P    | Bogor              |
|  3   | Purwanto Rojolele    |    L    | Yogyakarta         |
|  4   | Rita Sariwati        |    L    | Cibinong           |
|  5   | Suseno Mangun        |    L    | Ciamis             |
|  6   | Ratna Sari           |    P    | Cianjur            |
|  7   | Charlie Simanjuntak  |    L    | Medan              |
|  8   | Desi Sukmini         |    P    | Jakarta            |
|  9   | Thomas Eng           |    L    | Hongkong           |
| 10   | Ratih Sanggar        |    P    | Solo               |
| 11   | Achmad Widjaja       |    L    | Jakarta            |
| 12   | Delima Rahayu        |    P    | Demak              |
| 13   | Dadang Sukandar      |    P    | Cianjur            |
```

```
| 14 | Winda Sukma        |  P  | Yogyakarta                |
| 15 | Zaenal Nur         |  L  | Denpasar                  |
| 16 | Noeri Sakti        |  P  | Jakarta                   |
| 17 | Ahmaddin Sudin     |  L  | Banda Aceh                |
| 18 | Wati Sutinah       |  P  | Mojokerto                 |
| 19 | Wanto Sukarno      |  L  | Surabaya                  |
| 20 | Wawan Wanisar      |  L  | Bogor                     |
| 21 | Rendi Ganteng      |  L  | Bogor                     |
| 22 | Magda Magdalena    |  P  | Bandar Lampung            |
| 23 | Jumanto Dipuro     |  L  | Tegal                     |
| 24 | Deemas Sudirman    |  L  | Batam                     |
| 25 | Zulailah Zulha     |  P  | Demak                     |
+------+-----------------------+----------+-------------------------------------+
25 rows in set (0.01 sec)

mysql> SELECT noid, nama, kelamin, tempatlahir
    -> FROM karyawan2
    -> WHERE tempatlahir='jakarta' ;
+---------+---------------------+---------+---------------------------------+
| noid |      nama        | kelamin |          tempatlahir        |
+---------+---------------------+---------+---------------------------------+
|   8   | Desi Sukmini      |  P  | Jakarta                     |
|  11   | Achmad Widjaja    |  L  | Jakarta                     |
|  16   | Noeri Sakti       |  P  | Jakarta                     |
+---------+---------------------+---------+---------------------------------+
3 rows in set (0.00 sec)

mysql> SELECT noid, nama, kelamin, tempatlahir
    -> FROM karyawan2
    -> WHERE tempatlahir='jakarta' AND kelamin='P'
    -> ;
+---------+---------------------+---------+---------------------------------+
| noid |      nama        | kelamin |          tempatlahir        |
+---------+---------------------+---------+---------------------------------+
|   8   | Desi Sukmini      |  P  | Jakarta                     |
|  16   | Noeri Sakti       |  P  | Jakarta                     |
+---------+---------------------+---------+---------------------------------+
2 rows in set (0.00 sec)

mysql> SELECT noid, nama, kelamin, tempatlahir
    -> FROM karyawan2
    -> WHERE tempatlahir='jakarta'
    -> AND tempatlahir='yogyakarta' ;
Empty set (0.00 sec)
```

Lho... kenapa perintah di atas kok kosong? Padahal kan ada karyawan yang lahir di Jakarta dan Yogyakarta? Apa perintahnya salah? Kita coba lagi dengan perintah:

```
mysql> SELECT noid, nama, kelamin, tempatlahir
 -> FROM karyawan2
-> WHERE tempatlahir='jakarta'
-> OR tempatlahir='yogyakarta' ;
+---------+-----------------------------+------------+------------------------+
| noid    |          nama               | kelamin    |      tempatlahir       |
+---------+-----------------------------+------------+------------------------+
|    3    | Purwanto Rojolele           |      L     | Yogyakarta             |
|    8    | Desi Sukmini                |      P     | Jakarta                |
|   11    | Achmad Widjaja              |      L     | Jakarta                |
|   14    | Winda Sukma                 |      P     | Yogyakarta             |
|   16    | Noeri Sakti                 |      P     | Jakarta                |
+---------+-----------------------------+------------+------------------------+
5  rows in set (0.01 sec)
```

Nah.. sekarang bisa kan? Apa karena perintah 'AND' diganti dengan 'OR' baru bisa dilaksanakan dengan baik yach? Hmmm...saya masih penasaran... Kita coba lagi dengan kombinasi perintah lainnya...

```
mysql> SELECT noid, nama, kelamin, tempatlahir
    -> FROM karyawan2
    -> WHERE (tempatlahir='jakarta'
    -> OR tempatlahir='yogyakarta')
    -> AND kelamin='P' ;
+---------+-----------------------------+------------+------------------------+
| noid    |          nama               | kelamin    |      tempatlahir       |
+---------+-----------------------------+------------+------------------------+
|    8    | Desi Sukmini                |      P     | Jakarta                |
|   14    | Winda Sukma                 |      P     | Yogyakarta             |
|   16    | Noeri Sakti                 |      P     | Jakarta                |
+---------+-----------------------------+------------+------------------------+
3 rows in set (0.01 sec)
```

Perhatikan juga, kita harus menggunakan tanda kurung '(' dan ')' untuk mengapit perintah "(tempatlahir='jakarta' OR tempatlahir='yogyakarta')". Hal ini untuk menghindari kerancuan yang akan terjadi akibat adanya perintah AND yang mengikuti perintah OR. Sekarang kalo perintah diatas kita tulis ulang tanpa tanda kurung, maka hasilnya akan rancu.

Mari kita buktikan:

```
mysql> SELECT noid, nama, kelamin, tempatlahir
    -> FROM karyawan2
    -> WHERE tempatlahir='jakarta'
    -> OR tempatlahir='yogyakarta'
    -> AND kelamin='P' ;
```

```
+---------+-----------------------------+------------+-------------------------+
| noid    | nama                        | kelamin    | tempatlahir             |
+---------+-----------------------------+------------+-------------------------+
|     8   | Desi Sukmini                |      P     | Jakarta                 |
|    11   | Achmad Widjaja              |      L     | Jakarta                 |
|    14   | Winda Sukma                 |      P     | Yogyakarta              |
|    16   | Noeri Sakti                 |      P     | Jakarta                 |
+---------+-----------------------------+------------+-------------------------+
4 rows in set (0.01 sec)
```

Jelas dimana terjadinya kerancuan? Ya, Anda benar.... Ada "Achmad Widjaja" yang nyelonong masuk, padahal dia tidak semestinya muncul....

5.29. Menampilkan terurut

```
mysql> SELECT noid, nama, kelamin, tempatlahir
    -> FROM karyawan2
    -> ORDER BY nama DESC ;
```

```
+------+----------------------+----------+--------------------+
| noid |        nama          | kelamin  |  tempat lahir      |
+------+----------------------+----------+--------------------+
|  25  | Zulailah Zulha       |     P    | Demak              |
|  15  | Zaenal Nur           |     L    | Denpasar           |
|  14  | Winda Sukma          |     P    | Yogyakarta         |
|  20  | Wawan Wanisar        |     L    | Bogor              |
|  18  | Wati Sutinah         |     P    | Mojokerto          |
|  19  | Wanto Sukarno        |     L    | Surabaya           |
|   9  | Thomas Eng           |     L    | Hongkong           |
|   1  | Suwanto Dipura       |     L    | Majalengka         |
|   5  | Suseno Mangun        |     L    | Ciamis             |
|   4  | Rita Sariwati        |     P    | Cibinong           |
|  21  | Rendi Ganteng        |     L    | Bogor              |
|   6  | Ratna Sari           |     P    | Cianjur            |
|  10  | Ratih Sanggar        |     P    | Solo               |
|   3  | Purwanto Rojolele    |     L    | Yogyakarta         |
|  16  | Noeri Sakti          |     P    | Jakarta            |
|   2  | Mayangsariwati       |     P    | Bogor              |
|  22  | Magda Magdalena      |     P    | Bandar Lampung     |
|  23  | Jumanto Dipuro       |     L    | Tegal              |
|   8  | Desi Sukmini         |     P    | Jakarta            |
|  12  | Delima Rahayu        |     P    | Demak              |
|  24  | Deemas Sudirman      |     L    | Batam              |
|  13  | Dadang Sukandar      |     L    | Cianjur            |
|   7  | Charlie Simanjuntak  |     L    | Medan              |
|  17  | Ahmaddin Sudin       |     L    | Banda Aceh         |
|  11  | Achmad Widjaja       |     L    | Jakarta            |
+------+----------------------+----------+--------------------+
25 rows in set (0.01 sec)
```

Ya, cukup dengan menambahkan kata "**DESC**" maka Anda akan mendapatkan urutan Z-A, bukan?

Contoh-contoh di atas kebanyakan menggunakan data-data dengan jenis karakter. Sekarang bagaimana kalau yang menjadi objeknya adalah data-data yang bukan karakter? Misalnya, tanggal. Ayo, bagaimana dong contohnya.....?!

```
mysql> SELECT nama, kelamin, tgllahir
    -> FROM karyawan2
    -> ORDER BY nama  ;
```

nama	kelamin	tgl lahir
Achmad Widjaja	L	0000-00-00
Ahmaddin Sudin	L	0000-00-00
Charlie Simanjuntak	L	0000-00-00
Dadang Sukandar	L	0000-00-00
Deemas Sudirman	L	0000-00-00
Delima Rahayu	P	0000-00-00
Desi Sukmini	P	0000-00-00
Jumanto Dipuro	L	0000-00-00
Magda Magdalena	P	0000-00-00
Mayangsariwati	P	0000-00-00
Noeri Sakti	P	0000-00-00
Purwanto Rojolele	L	0000-00-00
Ratih Sanggar	P	0000-00-00
Ratna Sari	P	0000-00-00
Rendi Ganteng	L	0000-00-00
Rita Sariwati	P	0000-00-00
Suseno Mangun	L	0000-00-00
Suwanto Dipura	L	0000-00-00
Thomas Eng	L	0000-00-00
Wanto Sukarno	L	0000-00-00
Wati Sutinah	P	0000-00-00
Wawan Wanisar	L	0000-00-00
Winda Sukma	P	0000-00-00
Zaenal Nur	L	0000-00-00
Zulailah Zulha	P	0000-00-00

```
25 rows in set (0.01 sec)
```

Kemarin kita udah utak-atik masalah operator ganda. Udah disinggung juga tentang operator precedence. Kalo kita lihat lagi operator-operator yang bisa digunakan di dalam MySQL, maka dibawah ini adalah urutan otoritas dari para operator tersebut:

Paling Tinggi	BINARY
	NOT ! - (unary minus) * / % + - << >> & | < <= = <=> != <> >= > IN IS LIKE REGEXP RLIKE BETWEEN AND &&
Paling Rendah	OR ||

Bisa kita lihat, semakin keatas posisi operator, maka semakin tinggi tingkat kekuasaan operator tersebut. Begitu pula sebaliknya, semakin rendah posisinya maka akan semakin lemah.

Untuk operator yang sama kuat, misal + dan - digabung dengan operator * / %, maka akan ditentukan kekuatannya tergantung dari posisi mana yang paling kiri/paling awal ditemukan. Dan untungnya posisi kekuatan ini dapat diubah dengan bantuan tanda kurung "(" dan ")".

Contoh:

```
mysql> select 10+15-11*2, (10+15-11)*2,
    -> 2*6-5, 2*(6-5) ;
+-------------+---------------+--------+----------+
| 10+15-11*2 | (10+15-11)*2 | 2*6-5 | 2*(6-5) |
+-------------+---------------+--------+----------+
|           3 |            28 |      7 |        2 |
+-------------+---------------+--------+----------+
1 row in set (0.00 sec)
```

Nah, Anda bisa perhatikan bahwa walaupun angka dan operatornya sama, tapi hasilnya bisa berbeda. Dan itu karena adanya peranan dari tanda kurung "(" dan ")" yang akan mengubah peta posisi kekuatan operator....

Sekarang kita kan coba menggunakan operator **LIKE**. Operator LIKE ini digunakan untuk mencari data yang "menyerupai" atau "hampir sama" dengan kriteria tertentu. Biasanya untuk mencari data string/teks. Simbol **"%"** digunakan untuk membantu pelaksanaan operator LIKE. Posisi **"%"** sangat berpengaruh dalam menentukan kriteria. Bingung? Jangan bingung. Kita lihat saja contohnya yach.

Tampilkan data karyawan yang namanya **berawalan** huruf "**a**":(perhatikan posisi simbol persennya "%")

```
mysql> SELECT noid, nama FROM karyawan2
    -> WHERE nama LIKE "a%" ;
```

```
+------+-----------------+
| noid | nama            |
+------+-----------------+
|   11 | Achmad Widjaja  |
|   17 | Ahmaddin Sudin  |
+------+-----------------+
2 rows in set (0.00 sec)
```

Tampilkan data karyawan yang namanya berawalan huruf "**d**":

```
mysql> SELECT noid, nama FROM karyawan2
    -> WHERE nama LIKE "d%"
    -> ;
+------+-----------------+
| noid | nama            |
+------+-----------------+
|    8 | Desi Sukmini    |
|   12 | Delima Rahayu   |
|   13 | Dadang Sukandar |
|   24 | Deemas Sudirman |
+------+-----------------+
4 rows in set (0.00 sec)
```

Tampilkan data karyawan yang namanya **berakhiran** huruf "**i**":

```
mysql> SELECT noid, nama FROM karyawan2
    -> WHERE nama LIKE "%i"
    -> ;
+---+---------+
| noid | nama           |
+---+---------+
|    2 | Mayangsariwati |
|    4 | Rita Sariwati  |
|    6 | Ratna Sari     |
|    8 | Desi Sukmini   |
|   16 | Noeri Sakti    |
+---+---------+
5 rows in set (0.00 sec)
```

Tampilkan data karyawan yang namanya **berakhiran** "**wati**":

```
mysql> SELECT noid, nama FROM karyawan2
    -> WHERE nama LIKE "%wati" ;
+---+---------+
| noid | nama           |
+---+---------+
|    2 | Mayangsariwati |
|    4 | Rita Sariwati  |
+---+---------+
2 rows in set (0.00 sec)
```

Nah, sudah jelas dengan pemakaian LIKE? Kalau Anda perhatikan, operator LIKE ini tidak membedakan huruf besar dengan huruf kecil. Semuanya dianggap sama. Lalu bagaimana kalau kita ingin membedakan antara huruf besar dan kecil?

Bagaimana caranya agar operator LIKE dapat membedakan huruf besar dan kecil... Sederhana saja, cukup dengan menambahkan kata **BINARY** saja setelah perintah LIKE (jadinya **LIKE BINARY**)...Ya, kita coba aja dech....

```
mysql> SELECT noid, nama FROM karyawan3
    -> WHERE nama LIKE BINARY "a%"
    -> ;
Empty set (0.00 sec)
```

Lho kok ***empty set***? Sekarang coba dengan perintah yang berbeda sedikit:

```
mysql> SELECT noid, nama FROM karyawan3
    -> WHERE nama LIKE BINARY "A%"
    -> ;
+------+----------------+
| noid | nama           |
+------+----------------+
|  11  | Achmad Widjaja |
|  17  | Ahmaddin Sudin |
+------+----------------+
2 rows in set (0.00 sec)
```

Nah, sekarang baru berhasil. Setelah huruf 'a' diganti dengan huruf 'A', maka muncullah hasilnya. Ya, kan?

Sekarang bagaimana kalau saya ingin menampilkan data, bukan diawal atau diakhir kalimat, tapi berada **diantara** sebuah kata/kalimat. Gimana tuch?

```
mysql> SELECT noid, nama FROM karyawan3
    -> WHERE nama LIKE BINARY "%Di%"
    -> ;
+------+----------------+
| noid | nama           |
+------+----------------+
|    1 | Suwanto Dipura |
|   23 | Jumanto Dipuro |
+------+----------------+
2 rows in set (0.00 sec)
```

mysql> SELECT noid, nama FROM karyawan3

```
    -> WHERE nama LIKE BINARY " %W%" ;
+------+----------------+
| noid | nama           |
+------+----------------+
|   11 | Achmad Widjaja |
|   17 | Winda Sukma    |
|   18 | Wati Sutinah   |
|   19 | Wanto Sukarno  |
|   20 | Wawan Wanisar  |
+------+----------------+
5 rows in set (0.00 sec)
mysql> SELECT noid, nama FROM karyawan3
    -> WHERE nama LIKE BINARY " %wan%"
    -> ;
```

```
+-------+---------------------+
| noid  | nama                |
+-------+---------------------+
|     1 | Suwanto Dipura      |
|     3 | Purwanto Rojolele   |
|    20 | Wawan Wanisar       |
+-------+---------------------+
3 rows in set (0.00 sec)
```

Sudah jelas kan tentang LIKE? Bisa kan kalo mencoba sendiri untuk operator **NOT LIKE**? Prinsipnya sama aja kok dengan LIKE...

Operator **REGEXP** (singkatan dari **REG**ular **EXP**ressions) merupakan bentuk lain dari operator **LIKE**, dengan fungsi yang lebih disempurnakan. Operator REGEXP biasanya ditemani juga dengan simbol-simbol tetentu dalam melaksanakan tugasnya, seperti:

Simbol	Keterangan
.	Satu tanda titik (.) untuk mewakili satu karakter
[?]	Untuk mewakili beberapa karakter atau range yang ditentukan.
^	Untuk posisi awal dari sebuah kriteria yang ditentukan
$	Untuk posisi akhir dari sebuah kriteria yang ditentukan

Tampilkan nama karyawan yang berawalan huruf 'a':

```
mysql> SELECT noid, nama FROM karyawan2
    -> WHERE nama REGEXP "^a" ;
+-------+-----------------------+
| noid  |         nama          |
+-------+-----------------------+
|    11 | Achmad Widjaja        |
|    17 | Ahmaddin Sudin        |
+-------+-----------------------+
2 rows in set (0.00 sec)
```

Tampilkan data karyawan yang namanya **berawalan** huruf "**d**":

```
mysql> SELECT noid, nama FROM karyawan2
    -> WHERE nama REGEXP "^d"
    -> ;
+-------+-------------------+
| noid  |      nama         |
+-------+-------------------+
|     8 | Desi Sukmini      |
|    12 | Delima Rahayu     |
|    13 | Dadang Sukandar   |
|    24 | Deemas Sudirman   |
+-------+-------------------+
4 rows in set (0.00 sec)
```

Tampilkan nama karyawan yang berawalan huruf 'a' sampai dengan huruf 'd':

```
mysql> SELECT noid, nama FROM karyawan2
    -> WHERE nama REGEXP "^[a-d]
    -> ORDER BY nama ;
+-------+----------------------+
| noid |        nama          |
+-------+----------------------+
|   11 | Achmad Widjaja       |
|   17 | Ahmaddin Sudin       |
|    7 | Charlie Simanjuntak  |
|   13 | Dadang Sukandar      |
|   24 | Deemas Sudirman      |
|   12 | Delima Rahayu        |
|    8 | Desi Sukmini         |
+-------+----------------------+
7 rows in set (0.00 sec)
```

Tampilkan data karyawan yang namanya **berakhiran** huruf "**i**":

```
mysql> SELECT noid, nama FROM karyawan3
    -> WHERE nama REGEXP "i$"
    -> ;
+-------+-----------------+
| noid |     nama         |
+-------+-----------------+
|    2 | Mayangsariwati   |
|    4 | Rita Sariwati    |
|    6 | Ratna Sari       |
|    8 | Desi Sukmini     |
|   16 | Noeri Sakti      |
+-------+-----------------+
5 rows in set (0.00 sec)
```

Tampilkan data karyawan yang namanya **berakhiran** "**wati**":

```
mysql> SELECT noid, nama FROM karyawan2
    -> WHERE nama REGEXP "wati$" ;
+-------+-----------------+
| noid |     nama         |
+-------+-----------------+
|    2 | Mayangsariwati   |
|    4 | Rita Sariwati    |
+-------+-----------------+
2 rows in set (0.00 sec)
```

Tampilkan nama karyawan yang panjangnya 10 karakter:

```
mysql> SELECT noid, nama FROM karyawan2
    -> WHERE nama REGEXP "^.........$"
    -> ;
+-------+-----------------+
| noid |     nama         |
+-------+-----------------+
|    6 | Ratna Sari       |
|    9 | Thomas Eng       |
|   15 | Zaenal Nur       |
+-------+-----------------+
3 rows in set (0.00 sec)
```

Atau perintah di atas bisa juga ditulis dengan:

```
mysql> SELECT noid, nama FROM karyawan2
    -> WHERE nama REGEXP "^.{10}$"
    -> ;
+-------+--------------+
| noid |   nama       |
+-------+--------------+
|     6 | Ratna Sari |
|     9 | Thomas Eng |
|    15 | Zaenal Nur |
+-------+--------------+
3 rows in set (0.00 sec)
```

Dr Fujio Masuoka | Penemu *Flashdisk*

Dr. Fujio Masuoka

Nama Lengkap:
Dr. Fujio Masuoka

Tempat tanggal lahir:
Takasaki, Gunma, Jepang,
8 Mei 1943 (umur 74)

Kebangsaan: Jepang
Bidang: Teknik Elektro
Dikenal untuk memori Flash
Penghargaan ternama IEEE Morris N. Liebmann Memorial Award

Apakah Anda sering menggunakan *flashdisk* untuk menyimpan berbagai macam data pribadi, pekerjaan kantor hingga materi kuliah? Pernahkan Anda berpikir siapa yang menemukan teknologi *flashdisk* tersebut? Ya, dia adalah Dr. Fujio Masuoka, seorang warga Negara Jepang yang lahir di kota Takasaki, Gunma, Jepang pada tanggal 8 Mei 1943.

Masuoka merupakan seorang penemu *flashdisk memory* dengan proses perjalanan yang cukup panjang. Sejak kecil ia didorong oleh ibunya untuk belajar matematika hingga menyewa guru les pribadi ke rumah. Saat masuk universitas, ia percaya bahwa kemajuan teknologi dapat dicapai melalui cara kerja yang teoritis. Namun demikian, ia juga ahli di bidang ekonomi dan hukum, bahkan beliau telah meraih gelar *Bachelor of Science, Master of Science* dan *PhD* di bidang teknik listrik dari Tohoku University masing-masing di tahun 1966, 1968 dan 1971. Segera setelah lulus, Masuoka bergabung di perusahaan *Toshiba Research and Development Center* pada bulan April 1971.

Selama 3 bulan menjalani pekerjaan barunya, Dr Yoshiyuki Takeishi yang merupakan pimpinan perusahaan Toshiba menunjukkan sebuah Memori *Ultraviolet Erasable Intel Elektrik Programmable* (UV EPROM) kepada Masuoka, kemudian ia tertarik untuk mempelajari teknologi Intel tersebut, dan hanya butuh waktu 2 bulan ia berhasil menemukan sebuah struktur baru, yakni sebuah memori jenis MOS *read-only* memori yang dikenal dengan nama Samos dan menjadi hak paten pertama Masuoka di tahun 1972.

Sekitar tahun 1972 dan 1984, Masuoka mulai membuat terobosan baru yakni dengan membuat pasangan sel memori dinamis dengan struktur poli-silikon ganda. Namun pada tahun 1977, ia dipindah ke divisi semikonduktor Toshiba dan berhasil mengembangkan sebuah memori 1 Mbit DRAM. Pada tahun 1980, Masuoka kemudian dialihkan ke divisi produk rekayasa memori Toshiba untuk mengembangkan memori *flash.* Namun setelah itu, pada tahun 1984 ia malah digeser ke divisi desain memori rekayasa Toshiba yang mana ia mulai menyempurnakan dan mematenkan memori flash NOR.

Untuk membuat teknologi tersebut adalah 10 juta yen, yang Toshiba awalnya enggan untuk berinvestasi. Masuoka meyakinkan *Toshiba's consumer electronics research executives* bahwa chip flash memori 4 Mbit dapat digunakan kedalam kamera digital konsumen dengan memori flash yang berfungsi sebagai "film digital." Dengan pendanaan dari divisi elektronik konsumen, Masuoka terus mengembangkan dan mempresentasikan *flashdisk* 4 Mbit NAND-type *flash memory* pada Konferensi Solid-State Sirkuit Internasional (ISSCC) di New York City pada 1989.

Setahun kemudian, Masuoka mulai mengembangkan flash memory 256Kbit, dan berhasil mengembangkan proyek lain yakni NAND-type *flash memory* yang merupakan cikal bakal lahirnya flashdisk. Meskipun demikian, proyek tersebut masih belum bisa dikomersialisasikan karena produknya masih belum siap dan butuh penyempurnaan.

Untuk membuat dan memproduksi pra-fabrikasi komersial chip memori flash 4 Mbit, Masuoka membutuhkan biaya sekitar 10 juta yen. Biaya tersebut dibutuhkan untuk mengembangkan teknologi tinggi yang diproyeksikan untuk berbagai pola sirkuit pada

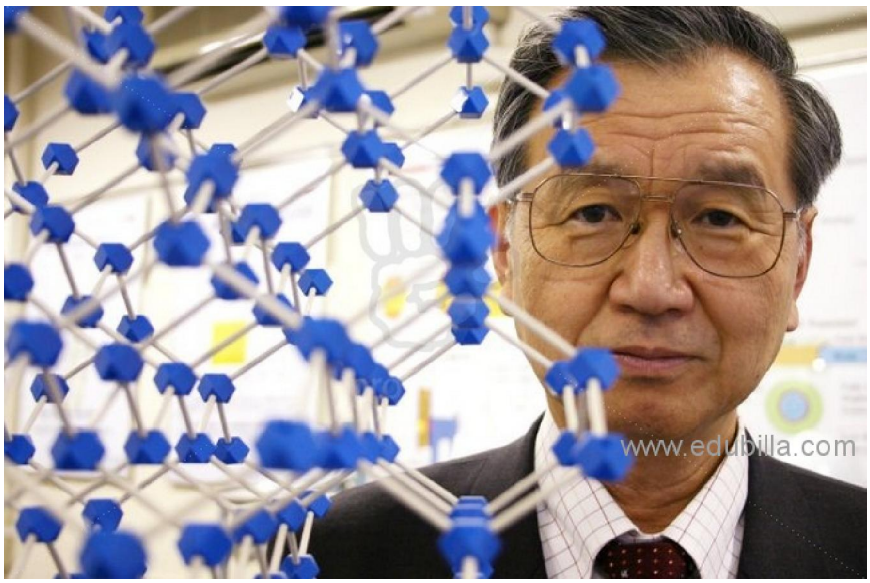
Model Flashdisk Sekarang

(Sumber: http://www.buahatiku.com/wp-content/uploads/2015/10/flashdrive-rotary.jpg)

setiap lapisan mikroprosesor. Pada awalnya Toshiba enggan berinvestasi, namun kemudian Masuoka berusaha meyakinkan Toshiba's consumer electronics research executives bahwa chip flash memori 4 Mbit dapat digunakan bahkan oleh kamera digital sekalipun. Dengan pendanaan yang dibantu oleh divisi elektronik konsumen Toshiba, Masuoka terus mengembangkan dan mempresentasikan flashdisk 4 Mbit NAND-type flash memory pada Konferensi Solid-State Sirkuit Internasional (ISSCC) di New York City pada 1989.

Tahun 1994, Masuoka bergabung dengan Tohoku University dimana ia menjadi profesor selama 13 tahun sebelum diangkat sebagai Profesor dari universitas Lembaga Penelitian Komunikasi Listrik. Untuk karyanya pada memori flash, Masuoka telah menerima berbagai penghargaan termasuk penghargaan di Negaranya sendiri, Jepang yakni hadiah dari Perdana Menteri Jepang Watanabe pada tahun 1977 dan Penghargaan Penemuan Nasional pada tahun 1980. Pada tahun 2007 silam, Masuoka bahkan dianugerahi Medali Ribbon Purple oleh Kaisar Akihito.

Flashdisk Tidak hanya ukurannya saja yang berkembang, tetapi bentuk dan fungsinya juga mengalami perubahan. Ada flash drive yang memakai rotary design sehingga kita tidak perlu khawatir kehilangan penutupnya. Tersedia juga flash drive yang dilapisi karet supaya tahan air atau dilengkapi dengan clip carabineer sehingga mudah digantungkan. Bahkan telah dibuat flash drive berbentuk model kartu kredit. Namanya wallet-friendly USB. Ukurannya hanya 86 x 54 x 1,9 mm. Jadi, dapat disimpan dengan aman di dalam dompet.



Untuk masalah keamanan yang dimiliki flash drive saat ini sebatas melindungi data yang ada agar tidak terakses oleh orang yang bukan pemiliknya. Cara kerja yang dipakai saat ini antara lain menggunakan full disk encryption atau physical authentication tokens. Sistem terbaru yang diperkenalkan tengah tahun 2005 lalu adalah biometric fingerprinting. Akan tetapi, metode sekuritas ini sangat mahal karena menggunakan teknologi tinggi Pada kenyataannya pemanfaatan flash drive telah berkembang untuk berbagai hal. Contohnya di sebuah artikel diuraikan langkah-langkah men-setting flash drive untuk memboot Windows XP. Syarat utamanya memang motherboard dan BIOS dari komputer kita dapat mendukung manajemen booting dari flash drive. Beberapa aplikasi juga dapat dijalankan dari flash drive tanpa harus meng-install-nya terlebih dahulu ke komputer.

Perkembangan flashdisk sangat cepat mengikuti perkembangan jaman, mulai dari bentuk hingga fungsinya. Ada flash drive dengan rotary desain yang memungkinkan kita untuk tidak mudah kehilangan bagian penutupnya hingga flash drive yang anti air dengan clip carabineer yang mudah digantungkan. Bentuk flash drive paling unik pun ada yang dibuat seperti bentuk kartu kredit, namanya wallet friendly yang berukuran 86x54x1,9mm.

(Sumber: http://www.ib-article.com/2012/09/dr-fujio-masuoka-biography-inventors.html)

Rangkuman

1. MySQL (bisa dibaca dengan **mai-es-ki-el** atau bisa juga **mai-se-kuel**, terserah Anda) adalah suatu perangkat lunak database relasi (bahasa kerennya **Relational Database Management System** atau **RDBMS**), seperti halnya **ORACLE, Postgresql, MS SQL**, dan sebagainya.
2. MySQL berarti merujuk ke perangkat lunak/software database MySQL, sedangkan mysql (dengan huruf kecil) berarti merujuk ke sebuah nama database.

Daftar Istilah

1. PHP
2. MySQL
3. Apache
4. Web Server
5. HTTP
6. Alamat IP
7. **URL**

Latihan Soal

1) Jelaskan perbedaan dar MySQL dan SQL?
2) Sebutkan perintah dalam MySQL di bawah ini!
   a. Melihat struktur tabel
   b. Membuat tabel baru
   c. Menghapus tabel
   d. Mengisi data baru
   e. Menampilkan data

STUDI KASUS YANG BERKAITAN DENGAN UKM

Menteri Koperasi dan UKM Sjarifuddin Hasan mengingatkan pelaku usaha kecil dan menengah yang mendapat kesempatan melakukkan studi banding ke negeri jiran Malaysia, harus memanfaatkan momentum itu menjalin *networking*. "Selain itu harus bisa saling berbagi pengalaman, *best practices*, mengidentifikasi peluang kerja konkret dengan pelaku usaha kecil dan menengah Malaysia," katanya dikutip Rabu (2/10/2013).

Ke-800 UKM merupakan binaan atau mitra pemasaran dari PT Sinde Budi Sentosa, dan diharapkan bisa membuka wawasan mereka sekaligus membuka inspirasi menghadapi dinamika persaingan global ke depan yang menuntut daya saing.

Sjarifuddin memahami kesempatan yang diberikan Sinde adalah bentuk apresiasi atas kesetiaan UKM sebagai mitra kerja pemasaran. Pemerintah juga sangat mendukung program Sinde karena berdampak positif bagi pertumbuhan wirausaha baru. Dikemukakan, pemberdayaan usaha mikro, kecil dan menengah (UMKM) di Idonesia memerlukan kepedulian, kerja sama maupun sinergitas dari pemerintah pusat, pemerintah daerah dan dunia usaha melalui agenda kemitraan.

Kemitraan yang dimaksud dalam konteks lingkungan UMKM sudah tidak asing, karena kemitraan dianggap menjadi salah satu alternatif upaya mengatasi berbagai problem internal. Kemitraan dimaknai sebagai bentuk hubungn bisnis antara pelaku usaha kecil dan menengah dengan usaha skala besar. "Kemitraan yang telah terjalin antara PT Sinde Budi Sentosa dan UKM merupakan contoh tumbuhnya iklim usaha yang kondusif.

Kami berharap kepercayaan ini bisa terus ditingkatkan dan menjadi contoh bagi usaha besar lainnya," tutur Sjarifuddin Hasan. Kepada perwakilan dari 800 UKM yang studi banding ke Malaysia, Menteri Koperasi mengingatkan supaya aktif menggali wawasan di Malaysia. Dengan demikian bisa menghasilkan kreativitas yang berdaya saing di pasar nasional maupun global.

Pada akhirnya diharapkan mampu mendukung pertumbuhan dan kemajuan industri nasional secara umum, maupun industri UMKM secara khusus. Dia mengingatkan PT Sinde Budi Sentosa pada awalnya hanya perusahaan farmasi sederhana, namun mampu berkembang menjadi besar. "Perusahaan ini telah memiliki jaringan usaha dan distribusi yang luas di negara-negara Asia, Eropa, dan Timur Tengah.

Karena itu program rutin yang diberikan Sinde kepada UKM Indonesia, harus bisa menjadi acuan meningkatkan kapasitas usaha."

*(Sumber : http://*suarapengusaha.com/2013/10/02/pengusaha-ukm-perlu-bangun-jaringan-di-malaysia/)

UNIVERSITAS NAROTAMA

BAB 6

Pemahaman Tentang Database

Tujuan Instruksional Umum

Setelah mempelajari bab ini, Mahasiswa dipastikan dapat:

1. Menjelaskan langkah–langkah aplikasi Databse dengan mengunakan MySQL
2. Menjelaskan langkah–langkah mengakses Database Melalui PHP
3. Menjelaskan cara menggunakan DNS

BAB

6

PEMAHAMAN TENTANG DATABASE

6.1. Menggunakan MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia.Berikut adalah fungsi – fungsi dalam MySQL yang digunakan untuk mengakses database MySQL menggunakan PHP.

Fungsi mysql_connect

Fungsi mysql_connect digunakan untuk membuat hubungan ke databse MySQl yang terdapat pada suatu host. Bentuk panggilannya :

```
mysql_connect(host,nama_pemakai,password)
```

Dalam hal ini :

- Host menyatakan nama host tempat MySQL berada
- Pemakai adalah nama pemakai yang digunakan untuk melakukan hubungan
- Password adalah password milik pemakai

Contoh:

```
$id_mysql =
mysql_connect("localhost","marsha","cantik");
```

Merupakan perintah untuk membentuk hubungan ke database MySQL yang terdapat pada **localhost** dengan nama pemnakai **marsha** dan password berupa **cantik**. Pengenal hubungan akan disimpan dalam variabel id_mysql.

Fungsi mysql_close

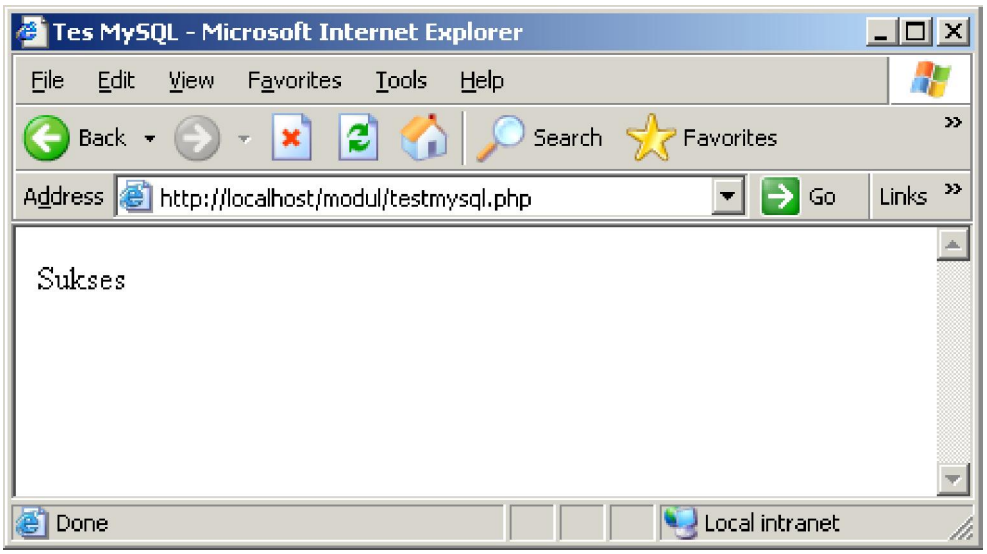
Fungsi ini berguna untuk menutup hubungan ke database MySQL. Bentuk pemanggilannya:

```
mysql_close(pengenal_hubungan)
```

berikut adalah contoh untuk menguji apakah hubungan ke MySQL dapat dilakukan atau tidak.

```
<?php
//testmysql.php

  $pemakai  = "root";
  $password = "";
  $id_mysql = mysql_connect("localhost",
                            $pemakai,
                            $password);
  if (! $id_mysql)
    die("Database MySQL tak dapat dibuka");

  mysql_close($id_mysql);

  print("Sukses");
?>
```

Hasilnya adalah:



Gambar 6.1 Tampilan hasil dari program

Fungsi mysql_select_db

Fungsi berguna untuk memilih database, seperti kalau Anda menggunakan perintah USE pada program klien MySQL. Perintah panggialnnya:

```
mysql_select_db(database, pengenal_hubungan)
```

Fungsi mysql_query

Fungsi ini berguna untuk mengeksekusi permintaan terhadap sebuah tabel atau sejumlah tabel. Bentuk pemanggilannya:

```
mysql_query(permintaan, pengenal_hubungan)
```

Fungsi mysql_db_query

Fungsi ini berguna untuk menjalankan suatu permintaan terhadap suatu database. Bentuk pemanggialnnya:

```
mysql_db_query(database, permintaan,
pengenal_hubungan)
```

Fungsi mysql_num_row

Fungsi ini berguna untuk memperoleh jumlah baris dari suatu permintaan (query) yang menggunakan SELECT. Bentuk pemanggialnnya :

```
mysql_num_row(pengenal_hasil)
```

Fungsi mysql_affected_rows

Fungsi ini berguna untuk memperoleh jumlah baris yang dikenai operasi INSERT, DELETE dan UPDATE. Bentuk pemanggialnnya:

```
mysql_affected_rows([pengenal_hubungan])
```

Fungsi mysql_num_fields

Fungsi berguna untuk memperoleh jumlah kolom pada suatu permintaan. Bentuk pemanggialnnya:

```
mysql_num_fields(pengenal_hasil)
```

Fungsi mysql_fetch_row

Fungsi ini menghasilkan sutu array yang berisi seluruh kolom dari ebuah bris pada suatu himpunan hasil. Bentuk pemanggialnnya:

```
mysql_fetch_row(pengenal_hasil)
```

Fungsi mysql_fetch_array

Fungsi ini mempunyai kegunaan serupa dengan mysql_fetch_row. Hanya saja setiap kolom akan disimpan dua kali pada array hasil. Yang pertama memiliki indeks angka dan yang kedua berindeks nama kolom. Bentuk pemanggialnnya:

```
mysql_fetch_array(pengenal_hasil)
```

Fungsi mysql_fetch_field

Fungsi ini berguna untuk memperoleh informasi sutu kolom. Format pemanggilannya:

```
mysql_fetch_field(pengenal_hasil[nomor_kolom])
```

MySQL adalah sebuah database server yang sangat aman. MySQL memiliki kemampuan memanajemen *user* dalam mengakses. Jadi, tidak sembarang *user* dapat mengakses sebuah *database* yang diciptakan MySQL. Maka sebelum anda memiliki *User* untuk mengakses MySQL anda juga dapat, Mengakses *database* MySQL menggunakan *User Root.*

Fungsi mysql_data_seek

Fungsi ini berguna untuk memindah pointer pada suatu himpunan hasil supaya menunjuk ke baris tertentu. Format pemanggialnnya:

```
mysql_data_seek(pengenal_hasil,nomor_baris)
```

Fungsi mysql_field_seek

Fungsi ini berguna untuk memindah pointer pada sutu himpunan hasil supaya menunjuk ke kolom tertentu.

```
mysql_field_seek(pengenal_hasil, nomor_kolom)
```

Fungsi mysql_create_db

Untuk menciptakan database MySQL. Format pemanggialnnya:

```
mysql_create_db(pengenal_hasil)
```

Fungsi mysql_drop_db

Fungsi ini berguna untuk menghapus database MySQL. Formatnya adalah:

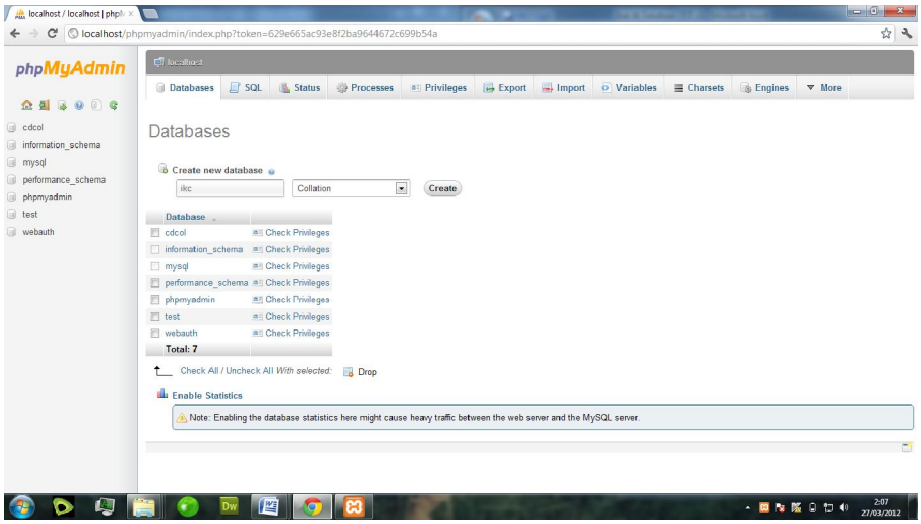
```
mysql_drop_db(database[,pengenal_hubungan])
```

6.2. Aplikasi Database menggunakan MySQL

6.2.1. Membuat Database mysql dengan PhpMyAdmin

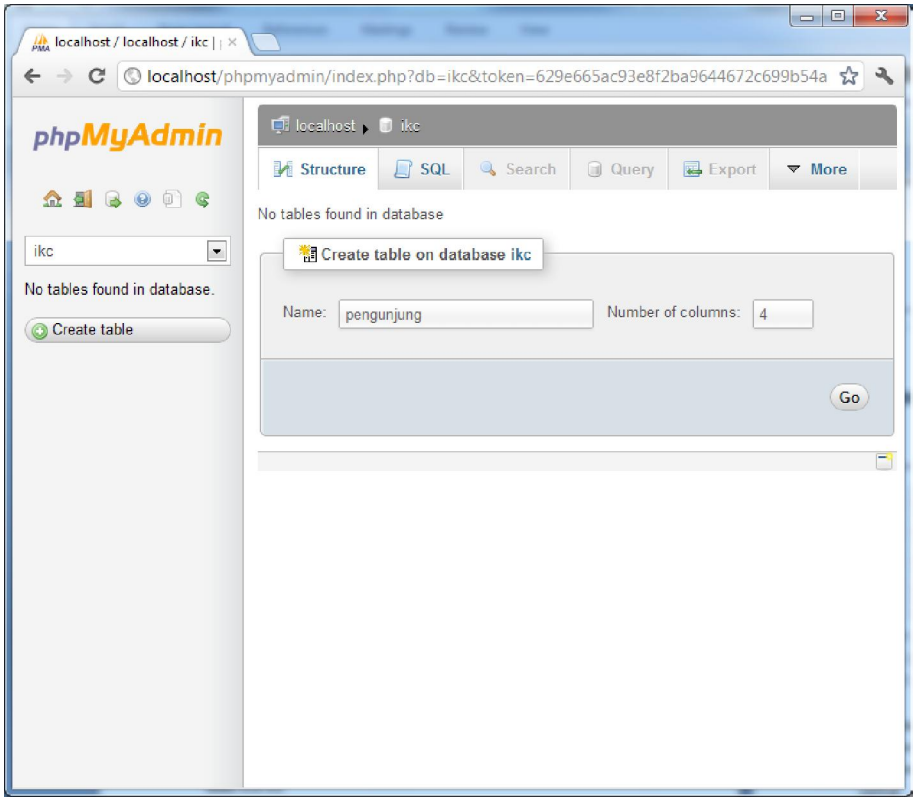
Pertama kita akan membuat database di MySQL dan menampilkannya di browser tentunya masih menggunakan Php. Untuk membuat database di mysql sebaiknya gunakan PhpMyadmin. Jika anda menggunakan phptriad untuk mengakses phpmyadmin, pada browser ketikkan http://localhost/phpmyadmin/.

Pertama kali buat database mysql dengan phpmyadmin. Beri nama databasenya **ikc**.



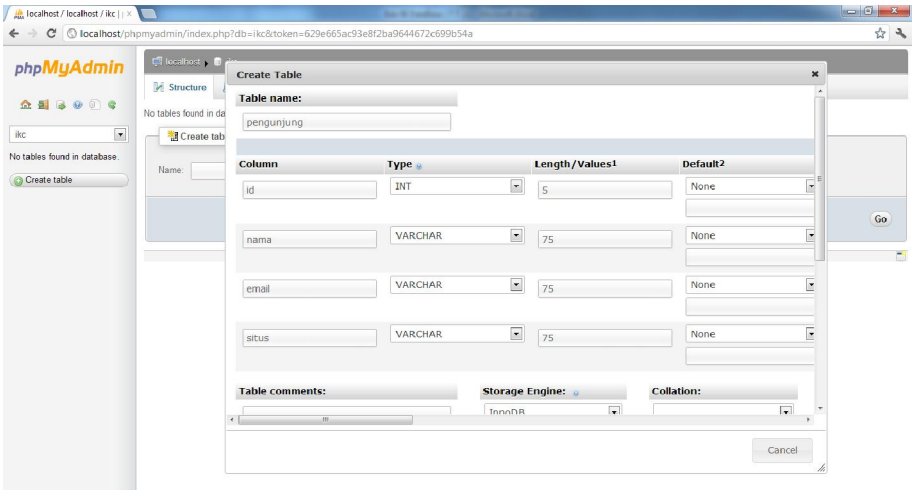
Gambar 6.2 Database Baru

Setelah itu klik ***create.*** Database baru sudah terbentuk, sekarang kita tinggal membuat tabelnya. Contoh pertama kita buat tabel sederhana yaitu tabel pengunjung pada sebuah halaman web.



Gambar 6.3 Tabel Pengunjung

Karena kita akan membuat tabel dengan nama **pengunjung** yang mempunyai field sejumlah 4 yaitu: id, nama, email dan situs. Maka masukkan **pengunjung** pada dan **4** pada **fields.** Klik **Go**.



Gambar 6.4 Membuat field pd database

Selanjutnya ialah mengisi field pada database. *Field* pertama masukkan **id,**

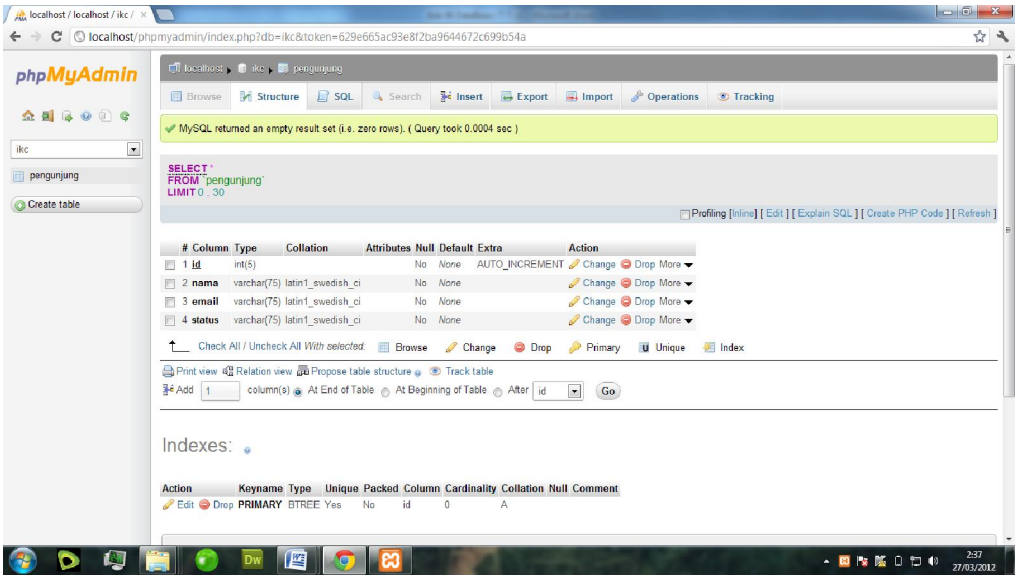
Tabel 6.1 *Field* pada tabel pengunjung

Field	Type	Length	Attribut	Null	Default	Extra	PIU
id	Int	5		Not null		Auto_inecreement	P
Nama	Varchar	75		Not null			
Email	Varchar	75		Not null			
situs	Varchar	75		Not null			

P : Primary I : Index U : Unique

Jangan lupa bahwa php *case sensitif* jadi sebaiknya isikan semua *field* dengan huruf kecil termasuk **id**. Pada gambar 4 **id** di setting *auto_increement* dan *Primary. Auto_increement* artinya setiap pengisian *record* (data) otomatis nilai id akan dinaikkan.

Primary artinya memberi kunci yang unik yang dapat di jadikan pengenal dalam tabel (dalam hal ini tabel pengunjung). Angka lebih baik di jadikan kunci utama dari sebuah tabel. Karena angka itu unik dan penggunaannya pun bisa di urutkan sehingga tidak akan ada duplikasi data. Dengan kata lain hanya dengan **id** kita nantinya akan dapat melakukan *editing, deleteing, update* pada *record.* Jika sudah klik **save**. Nantinya hasil tabel pengunjung seperti Gambar 6.5 berikut ini:



Gambar 6.5 Tabelpengunjung

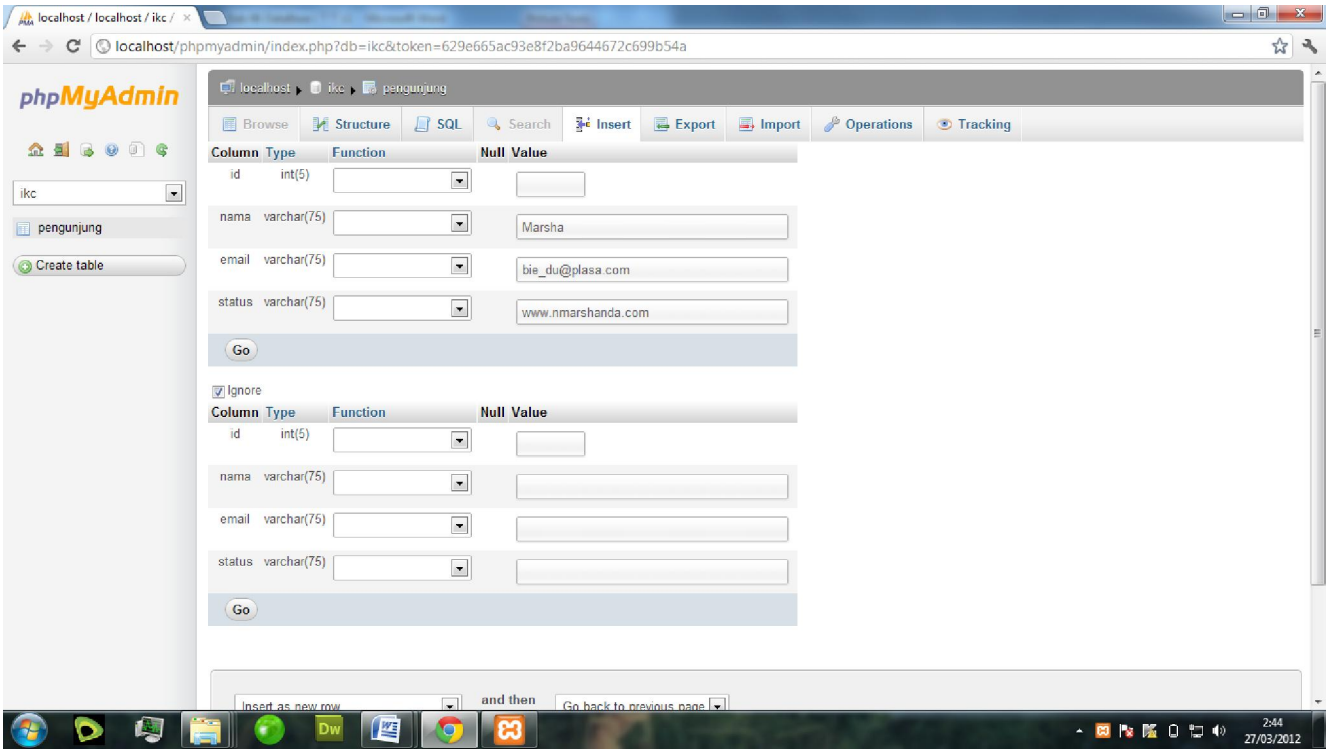
Browse **:** untuk melihat isi *table (record)*

Insert **:** untuk mengisi *table*

Drop **:** Menghapus *field*

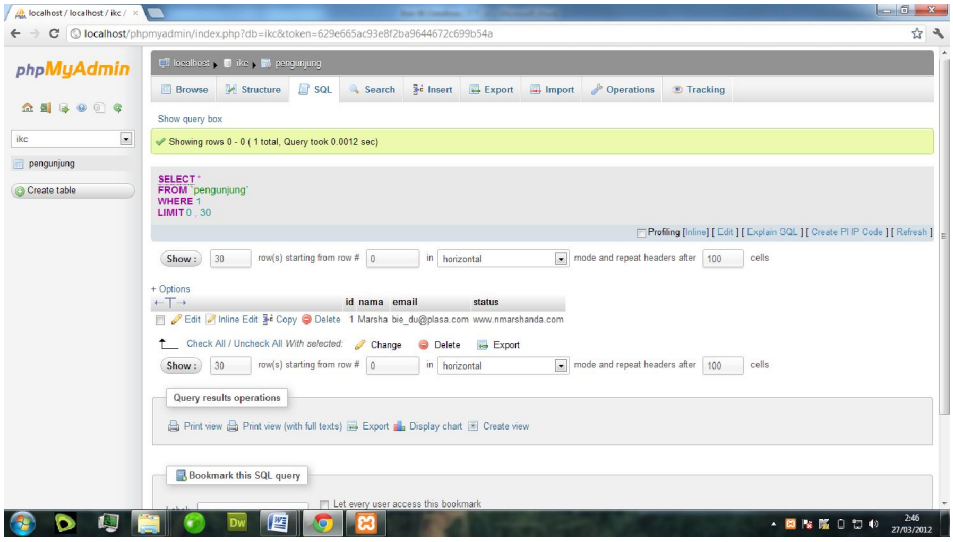
Primary, Index, Unique : Menjadikan *primary, index atau unique*

Isi database tersebut dengan beberapa records, klik **Insert** untuk mengisinya:



Gambar 6.6 Tabel pengunjung

Karena id **auto_increement** maka id tidak perlu diisi. Jika sudah kli Save. Untuk melihat record klik Browse:



Gambar 6.7 Browse table

Apabila anda tidak menggunakan PhpMyadmin untuk administrasi database MySql maka script yang dapat anda ketikkan di mysql atau paste di jendela query:

```
CREATE TABLE `pengunjung` (`id` INT(5) not null
AUTO_INCREMENT, `nama`
VARCHAR(75) not null, `email` VARCHAR(75) not
null, `situs` VARCHAR(75) not null ,
PRIMARY KEY (`id`))
```

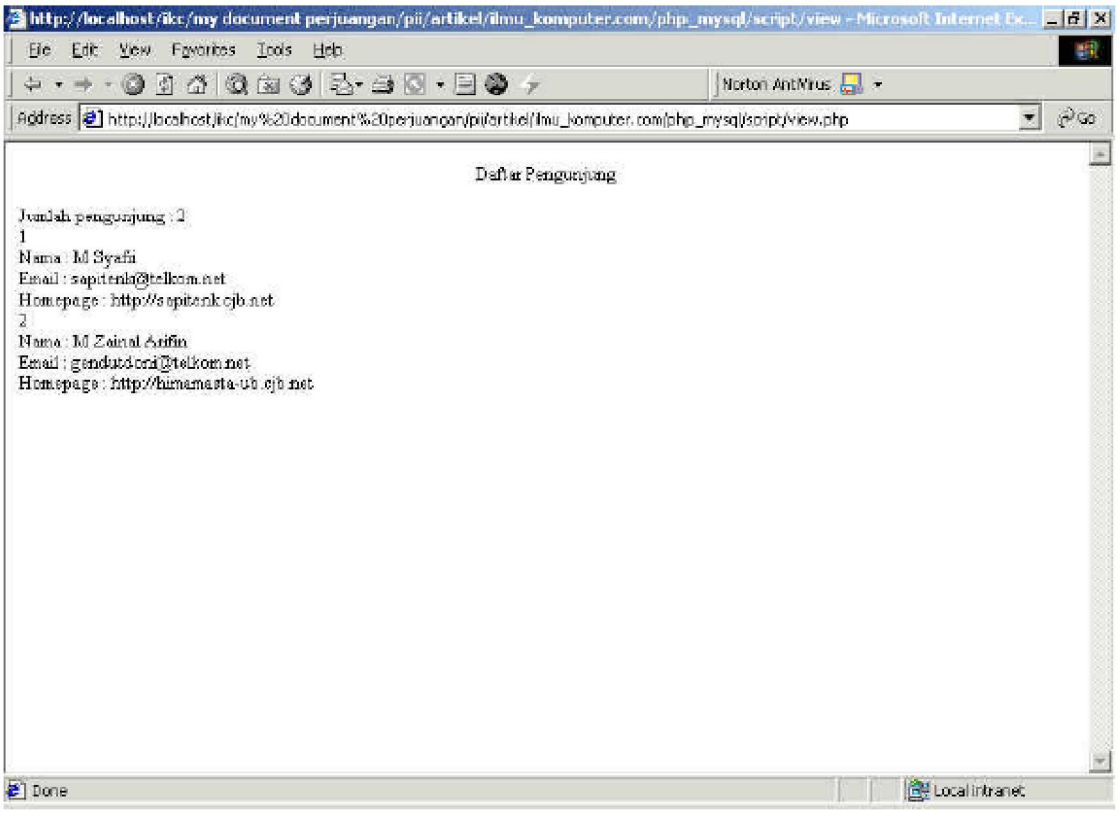
6.3. View

Database sudah terbentuk, sekarang kita akan mulai bagaimana caranya menampilkan isi tabel tersebut pada browser yang tentunya menggunakan Php. Pertama jangan lupa start dulu apache dan mysql anda (jika anda menggunakan PhpTriad).

view.php

```
1  <?
2
3  $host = localhost;// alamat atau host
4  $user = root; // username
5  $pass = ""; // password
6
7  $koneksi = mysql_connect("$host","user",$pass);
8
9  mysql_select_db("ikc",$koneksi);
10
11 $query = mysql_query("select * from pengunjung",$koneksi);
12
13 $jumlah = mysql_num_rows($query);
14
15 echo "<center>Daftar Pengunjung</center>";
16 echo "Jumlah pengunjung : $jumlah";
   while($baris
17 =mysql_fetch_array($query))
18 {
19 echo "<br>";
20 echo $baris[0];
21 echo "<br>";
22 echo "Nama : ";
23 echo $baris[1];
24 echo "<br>";
25 echo "Email : ";
26 echo $baris[2];
27 echo "<br>";
   echo "Homepage :
28 ";
29 echo $baris[3];
30 }
31
32 ?>
```

Hasilnya adalah sebagai berikut:



Gambar 6.8 *Browse table*

Kesimpulan :

Untuk menampilkan database mysql di Browser dengan Php langkah-langkahnya adalah:

- Hubungkan dulu Php dengan mysql (menggunakan fungsi mysql_connect())
- Pilih database (menggunakan fungsi mysql_select_db)
- Query (menggunakan fungsi mysql_query())
- Tampilkan (menggunakan fungsi mysql_fetch_array)

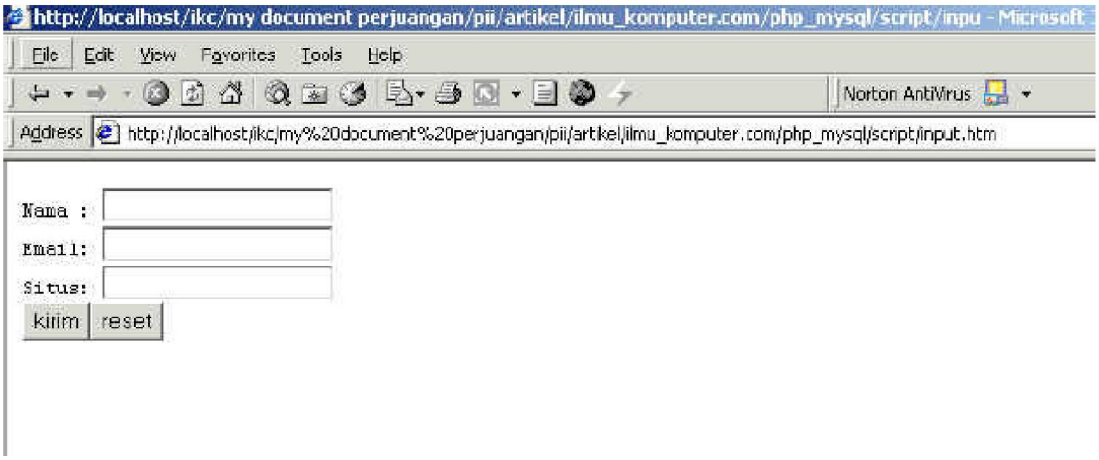
6.4. Insert

File view.php tidak memungkinkan untuk seseorang memasukkan record dikarenakan file tersebut hanya menampilkan record. Untuk dapat membuat sebuah halaman yang dapat memasukkan sebuah inputan maka perlu dibentuk sebuah *form*. Form tersebut memuat variabel-variabel yang dibutuhkan yang kemudian setelah di *submit* akan di proses dan dimasukkan dalam database. Maka dibutuhkanlah dua file yaitu input.htm dan file pemroses input.php

input.htm

```
1
2  <html>
3  <head></head>
4  <body>
5  <form method="post" action="input.php">
6  <pre>
7  Nama : <input type="text" name="nama">
8  Email: <input type="text" name="email">
9  Situs: <input type="text" name="situs">
10 <input type="submit" name="submit" value="kirim"><input type="reset"
   name="reset" value="reset">
11 </pre>
12 </form>
13 </body>
   </html>
```

<input type=”text” **name=”nama”**> **name=”nama”** artinya bahwa apapun yang diinputkan dalam text box tersebut menjadi variable nama.



Gambar 6.9 *Hasil inputan textbox*

input.php

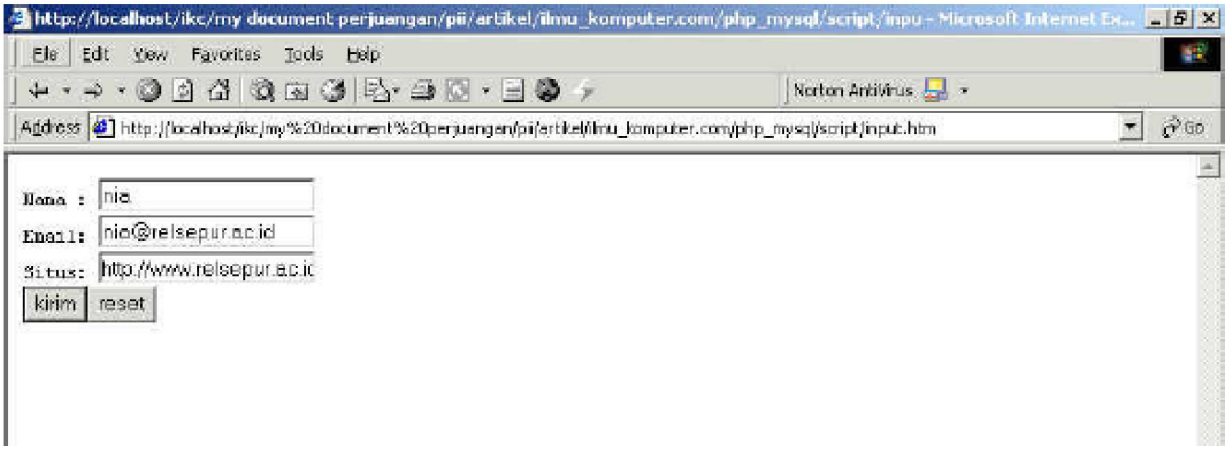
```
1
2
3
4  <?
5  $koneksi = mysql_connect("localhost","root","");
6  mysql_select_db("ikc", $koneksi);
7  mysql_query("insert into pengunjung (nama, email, situs)
8  values ('$nama', '$email', '$situs')",$koneksi);
9  echo "Data telah dimasukkan";
   ?>
```

Pada source code input.htm terdapat 3 variabel yaitu nama, email dan situs. Ketiga variable tersebut nantinya akan diproses dalam file input.php. Cara pemrosesan atau pemasukan data dalam database mysql ialah menggunakan perintah mysql yaitu perintah **insert**.

```
mysql_query("insert into pengunjung (nama, email,
situs)
values ('$nama', '$email', '$situs')",$koneksi);
```

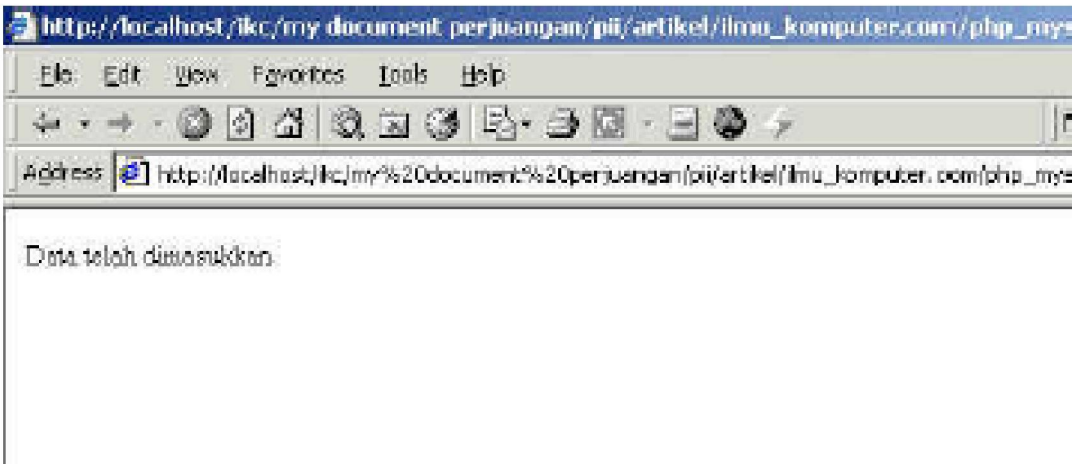
Sintak tersebut memasukkan data yang terdapat pada variable nama, variable email, variable situs untuk di masukkan ke dalam tabel pengunjung dengan urutan field nya nama, email dan situs. **Id** tidak dimasukkan karena sudah **auto_increement**. Memasukkan dalam database harus urut dengan fieldnya. Pada tabel pengunjung urutannya id, nama, email, situs. Maka pada pengisiannya harus di urutkan.

Sekarang akan kita coba memasukkan data. Buka file input.htm di browser:



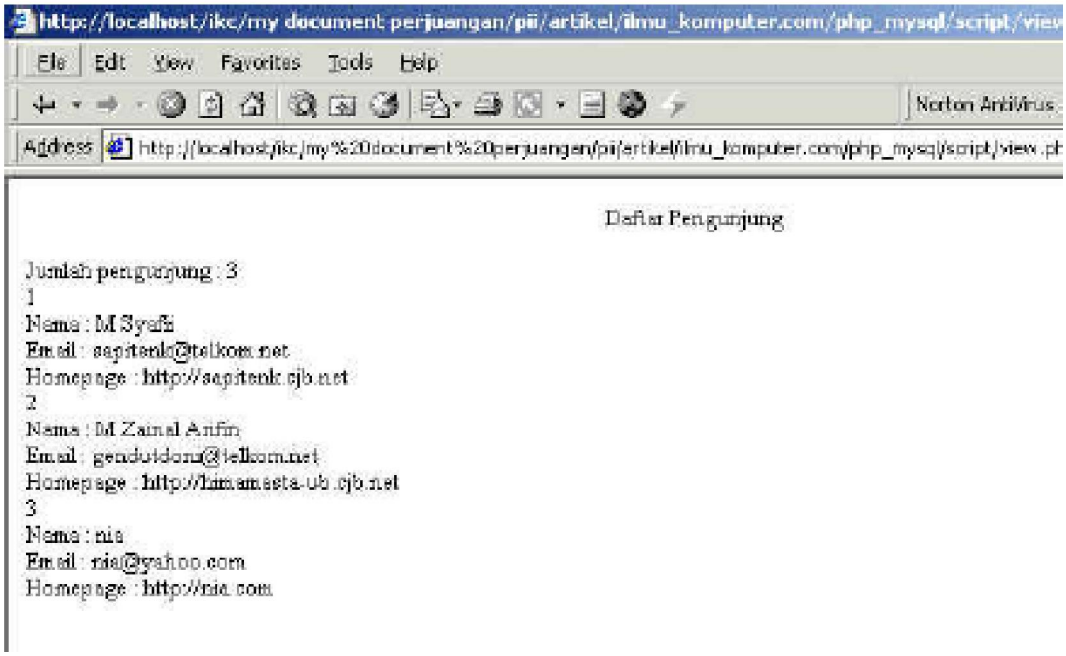
Gambar 6.10 file input.htm di browser

Klik kirim, maka selanjutnya akan diproses di input.php



Gambar 6.11 file input.php

Untuk melihat data buka lagi file view.php



Gambar 6.12 Tambahan record

6.5. Edit

Sebuah database tentunya terdapat revisi-revisi atau perubahan isi tabel atau record. Maka harusnya terdapat sebuah form yang mampu mengedit isi yang terdapat dalam database. Jika pada file input.php kita sudah dapat memasukkan data ke dalam database maka kali ini kita akan mengedit data yang telah di masukkan. Langkah pertama kita harus memodifikasi file view.php agar dapat mempunyai *link* untuk menuju sebuah form peng-editan yang selanjutnya dengan form tersebut akan di proses dan di update= ke dalam database.

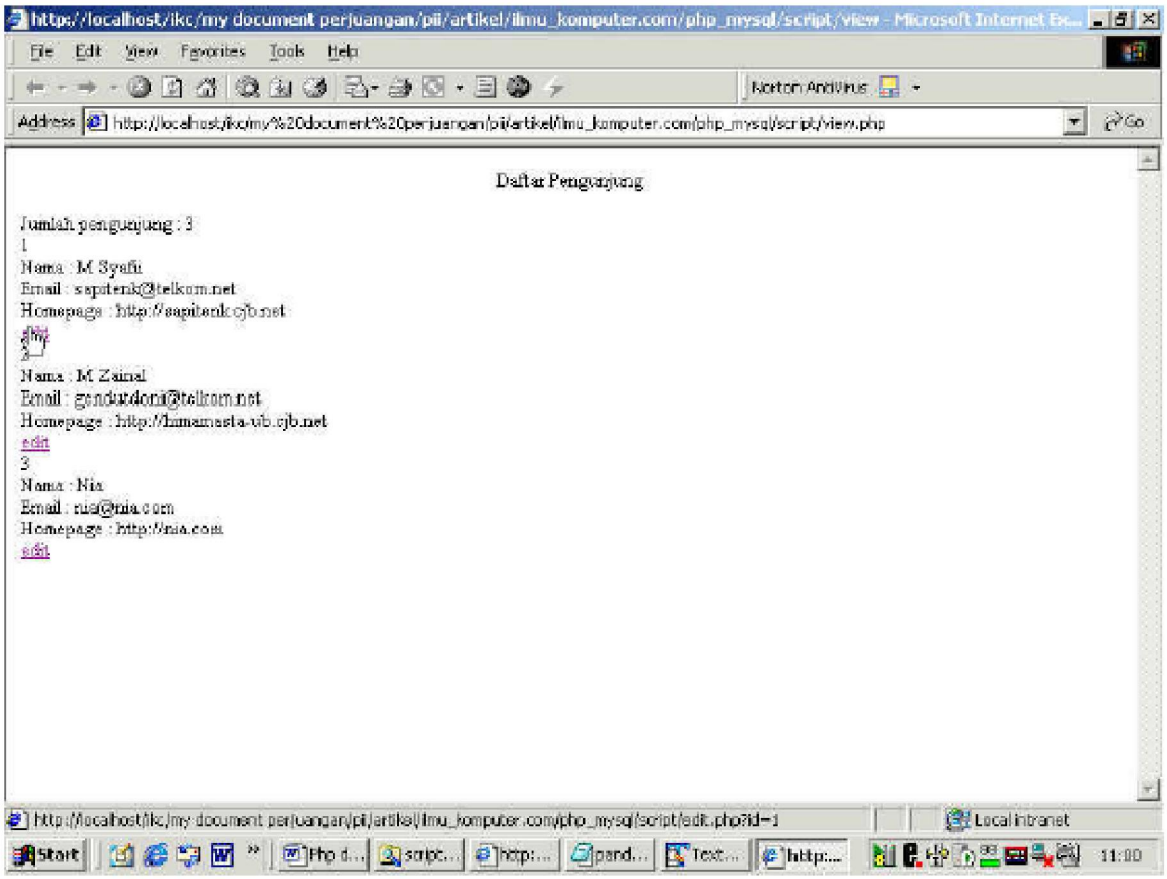
Tabel 6.2 Source code baru view.php

view.php

```
 1  <?
 2
 3
 4  $koneksi = mysql_connect("localhost","root","");
 5
 6  mysql_select_db("ikc",$koneksi);
 7
 8  $query = mysql_query("select * from pengunjung",$koneksi);
 9
10  $jumlah = mysql_num_rows($query);
11
12  echo "<center>Daftar Pengunjung</center>";
```

```
13  echo "<br>";
14  echo "Jumlah pengunjung : $jumlah";
    while($baris =
15  mysql_fetch_array($query))
16  {
17  echo "<br>";
18  echo $baris[0];
19  echo "<br>";
20  echo "Nama : ";
21  echo $baris[1];
22  echo "<br>";
23  echo "Email : ";
24  echo $baris[2];
25  echo "<br>";
    echo "Homepage :
26  ";
27  echo $baris[3];
28  echo "<br><a href=edit.php?id=$baris[0]>edit</a>";
29  }
30
31  ?>
```

Script view.php mendapat tambahan pada baris 28. Dimana dibuat sebuah link yang memanfaatkan variable **id** untuk di proses pada tahap selanjutnya.



Gambar 6.13 View.php

Jika cursor diarahkan pada link **edit** maka link tersebut akan menggunakan **id** dari record untuk dijadikan sebagai variable pemroses di file edit.php.

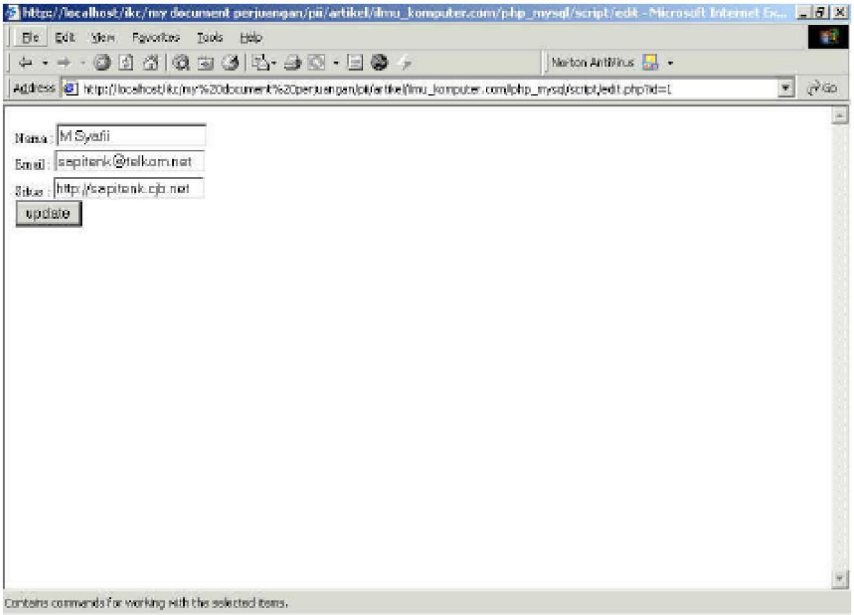
edit.php

```
1  <?
2
3  $koneksi = mysql_connect("localhost","root","");
4
5  mysql_select_db("ikc",$koneksi);
   $query = mysql_query("select * from pengunjung where
6  id='$id'",$koneksi);
   while ($baris =
7  mysql_fetch_array($query))
8  {
9
   echo "<form method=\"post\"
10 action=\"update.php\">";
11 echo "Nama : <input type=\"text\" name=\"nama\" value=\"$baris[1]\">";
12 echo "<br>";
13 echo "Email : <input type=\"text\" name=\"email\" value=\"$baris[2]\">";
14 echo "<br>";
15 echo "Situs : <input type=\"text\" name=\"situs\" value=\"$baris[3]\">";
16 echo "<br>";
17 echo "<input type=\"submit\" name=\"submit\" value=\"update\">";
18 echo "<input type=\"hidden\" name=\"id\" value=\"$baris[0]\">";
19 echo "</form>";
20
21 }
22
23 ?>
```

Penjelasan:

Inti dari edit.php yaitu dia membawa sebuah variable id yang nilainya diambil dair view.php. Variable tersebut ditampung dalam variable **$id** diambil dari edit.php?**id**=1. Pada baris 6 dilakukan query untuk mengambil seluruh record yang id-nya sama dengan $id (dalam contoh id=1) .

Baris 18 digunakan sebagai pengendali proses agar jika form diatas di *submit* maka variable **id** tetap dapat digunakan pada file update.php atau jika ingin sebuah variable dapat di 'tangkap' untuk diproses maka ketikkan perintah seperti pada baris 18 dan yang penting harus tetap di antara tag <form> dan </form>



Gambar 6.14 edit.php?id=1

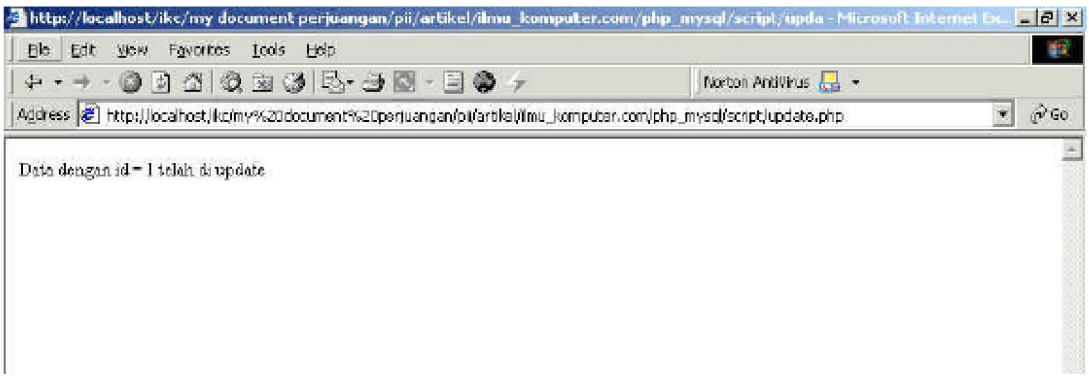
Bisa Anda lihat bahwa dengan modal variable id saja record tersebut dapat di edit. Logika dari edit.php ialah hasil query di tampung dalam text box. Sehingga nilai (value) yang ada dalam text box hasil query. Silahkan merubah nya jika sudah klik **update.**

update.php

```
1  <?
2
3  $koneksi = mysql_connect("localhost","root","");
4
5  mysql_select_db("ikc",$koneksi);
6  $query = mysql_query("update pengunjung set id='$id',
7  nama='$nama', email='$email', situs = '$situs' where id='$id'");
8  echo "Data dengan id = $id telah di update";
9  ?>
```

Penjelasan :

Baris 6 mengupdate record dimana **id** yang ingin diupdate sama dengan variable **id** yang di 'bawa' oleh file update.php. Jika pada file edit.php baris 18 di hapus maka query tidak dapat dilaksanakan karena nilai variable **id** tidak dapat diakses.



Gambar 6.14 *Update* data

Proses *update* telah selesai. Anda bisa meliha isi data pada *file view.php.*

6.6. Searching

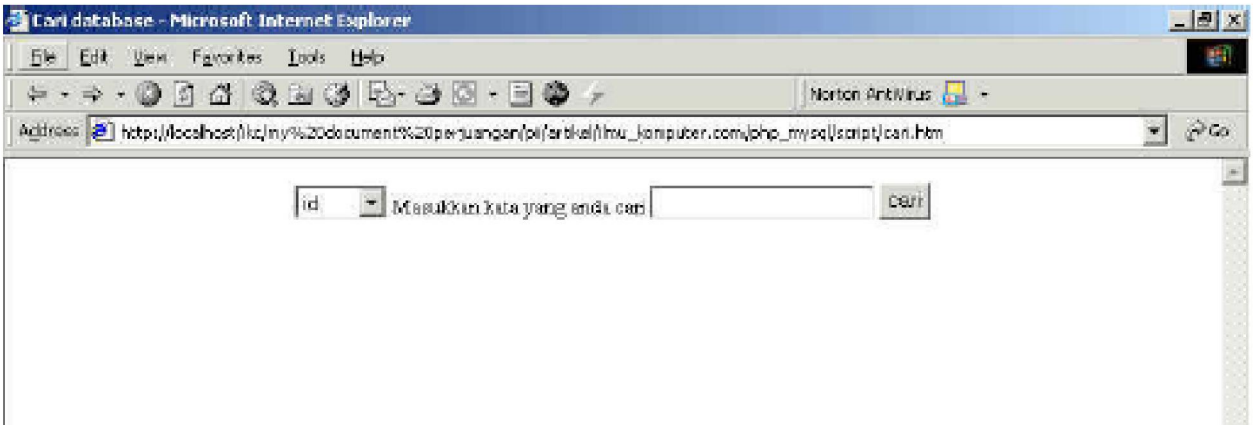
Database yang besar yang tentunya mempunyai record ribuan atau bahkan ratusan ribu yang seharusnya mempunyai fasilitas pencarian data atau searching. Pencarian ini bersifat fleksibel dimana dapat mencari berdasarkan field yang ada dalam tabel. Pada contoh tabel pengunjung, pencarian dapat di *customize*-kan berdasarkan id, nama, email dan situs. Untuk membuat *form* pencarian menggunakan html sedangkan pemrosesan *variable* menggunakan php.

cari.htm

```
1  <html>
2  <head><title>Cari database</title>
3  </head>
4  <body>
5  <center>
   <form name  methode="post"
6  action="cari.                    php">
7  <select name="kolom">
8  <option value="id">id</option>
9  <option value="nama">nama</option>
10 <option value="email">email</option>
11 <option value="situs">situs</option>
12 </select>
   Masukkan kata yang anda cari <input
13 type                             ="text" type ="text" name="cari">
14 <input type ="submit"  value="cari" >
15 </form>
16 </center>
17 </body>
18 </html>
```

Penjelasan :

- Baris 8-11 menyatakan *variable* untuk masing-masing *field* yang terdapat dalam tabel pengunjung.
- Baris 13 menyatakan kata/*keyword* yang di cari.



Gambar 6.15 Cari.htm

Tabel 6.3 Tabel Source code cari.php

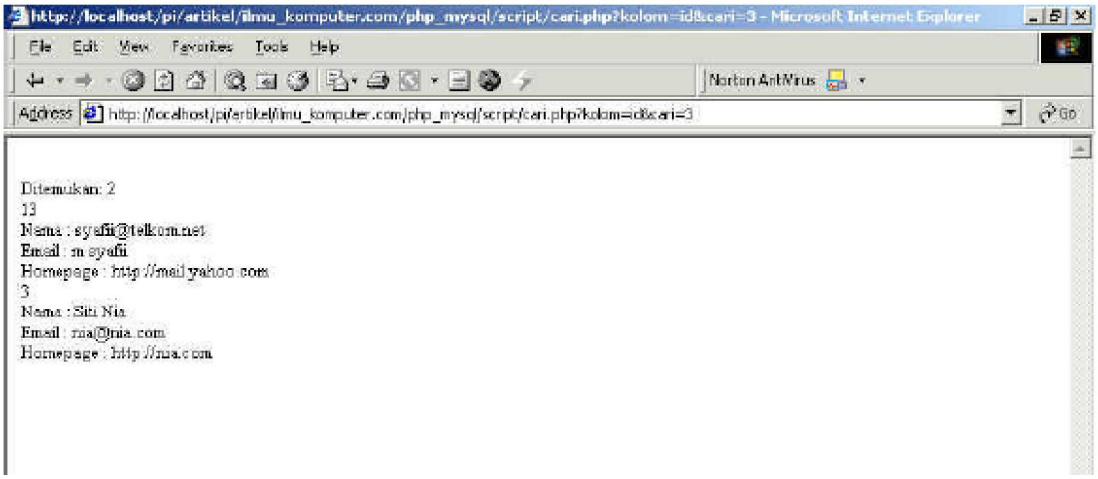
cari.php

```
1  <?
2
3  $koneksi = mysql_connect("localhost","root","") ;
4
5  mysql_select_db("ikc",$koneksi) ;
6
   $query = mysql_query("select * from pengunjung where $kolom like '%$cari%'
7  ",$koneksi);
8
9  $jumlah = mysql_num_rows($query);
10
11
```

```
12  echo "<br>";
13  echo "Ditemukan: $jumlah";
    while($baris =
14  mysql_fetch_array($query))
15  {
16  echo "<br>";
17  echo $baris[0];
18  echo "<br>";
19  echo "Nama : ";
20  echo $baris[1];
21  echo "<br>";
22  echo "Email : ";
23  echo $baris[2];
24  echo "<br>";
    echo "Homepage :
25  ";
26  echo $baris[3];
27
28  }
29
30  ?>
```

Inti dari script.php ialah pada baris 7. Secara umum *query* akan melakukan pencarian kata ($cari) dalam sebuah field ($kolom) jika cocok akan ditampilkan. Penggunakan *like* di maksudkan untuk mengambil kata/*keyword* yang terdapat dalam *field* yang sepenggal katanya atau huruf sama dengan *keyword* yang di masukkan.

Contoh mencari pengunjung dengan nama **syafii** maka jika menggunakan *like* akan ditampilkan data **M Syafii, Syafii Maarif,** dan lain-lain. Sepanjangan kata atau kalimat cocok dengan *keyword* meskipun satu huruf .



Gambar 6.16 pencarian kolom

Pada contoh diatas mencari pada kolom *(field)* **id** dengan *keyword* **3.** Hasilnya ditampilkan id : **13**, **3** .

6.7. Menggunakan ODBC

6.7.1. Apa itu ODBC

ODBC merupakan singkatan dari *Open database Connectivity* yaitu merupakan suatu standart yang dikembangkan oleh Microsoft untuk mengakses database agar lebih mudah dengan fungsi yang bersifat umum. Tentu saja untuk mengakses database pada PHP tidak lagi menggunakan cara-cara seperti pada MS Acces, VBasic ataupun Delphi yang dengan menggunakan komponen tertentu. Untuk mengakses database pada PHP diperlukan perintah SQL *(Structured Query Language)*. Pada program berbasis Windows sumber data ODBC dapat diakses pada program melalui DSN *(Data Source Name)*.

Open Database Connectivity (ODBC) adalah sebuah standar terbuka untuk konektivitas antar mesin basis data. Standar ini menyediakan API yang dapat digunakan untuk menjalankan dan mengkoneksikan sebuah aplikasi dengan sebuah sistem manajemen basis data (SMBD).

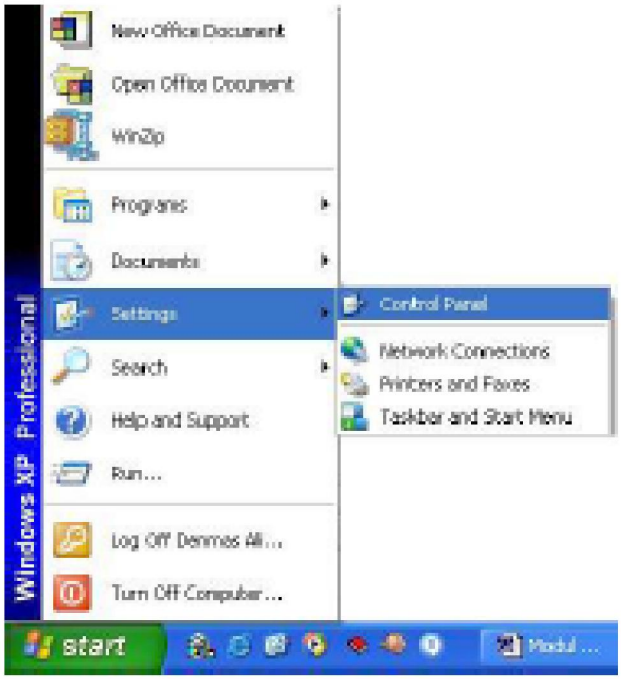
6.7.2. Membuat DSN

Sebelum membuat DSN terlebih dulu harus dibuat database yang akan digunakan dalam aplikasi jaringan ini di program MS Acces. Caranya seperti lihat pada halaman berikut. Dan simpanlah *database* yang anda buat ke dalam directori **C:/apache/Acces/pemasaran.mdb**

Untuk membuat DSN caranya adalah sbb :

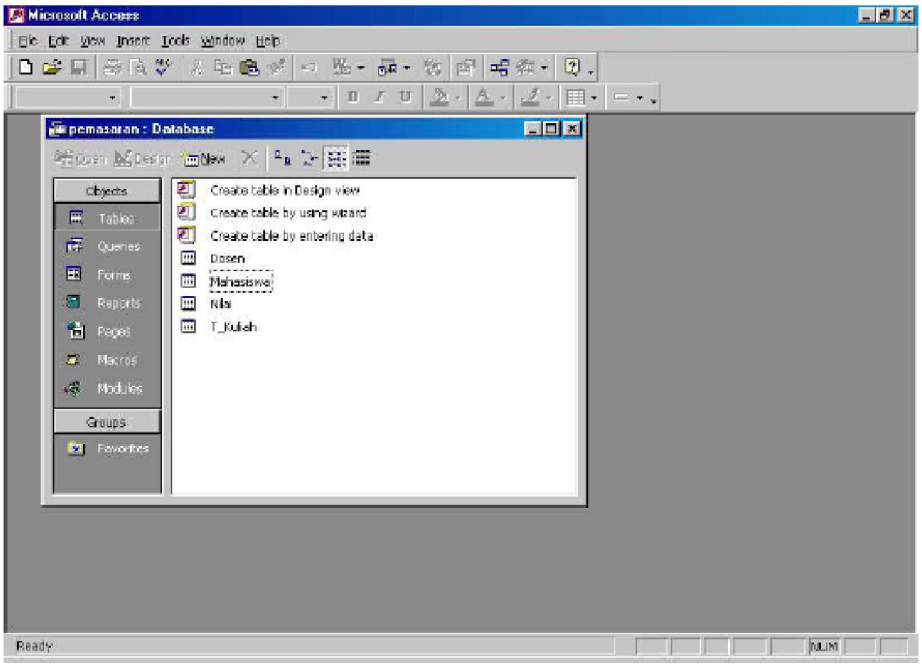
- Pilih *menu Star* > *Setting* > *Control Panel*

Maka akan muncul menu sebagai berikut:



Gambar 6.17 Langkah-langkah pembuatan Database

Membuat *Database* pemasaran dengan Microsoft Acess



Gambar 6.18 Tampilan dari Microsoft Acces

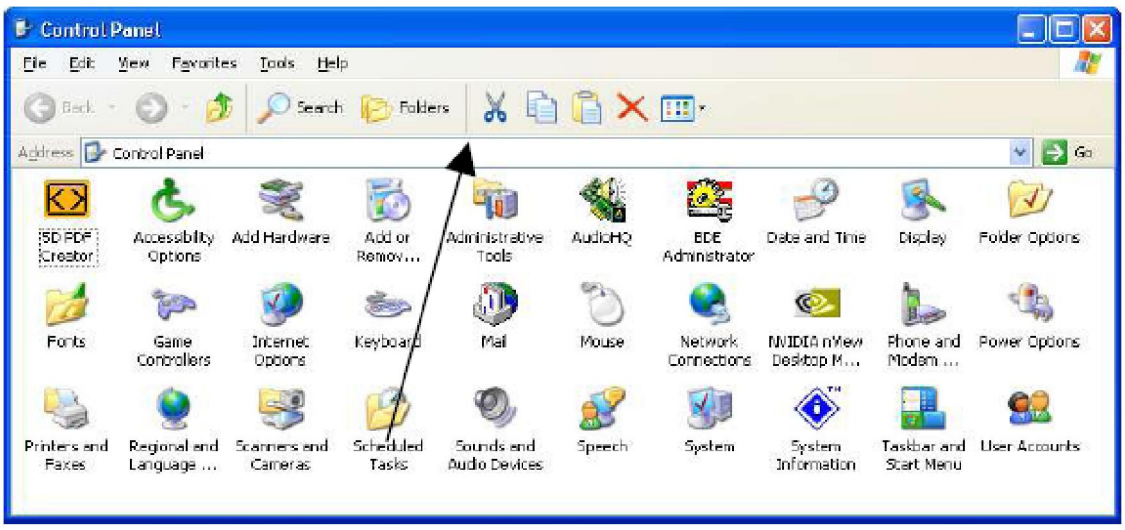
Buatlah *Database* dengan nama pemasaran dalam MS Accces yang terdiri dari 4 tabel seperti pada gambar di atas.

TABEL MAHASISWA
NIM (Number)
Nama (Text/20)
Alamat (Text/50)
Tgl_Lahir (Date/time)
Orang_Tua (Text/20)

TABEL DOSEN
NIP (Number)
Nama (Text/20)
Alamat (Text/50)
Telepon (Text/15)
Golongan (Text/5)

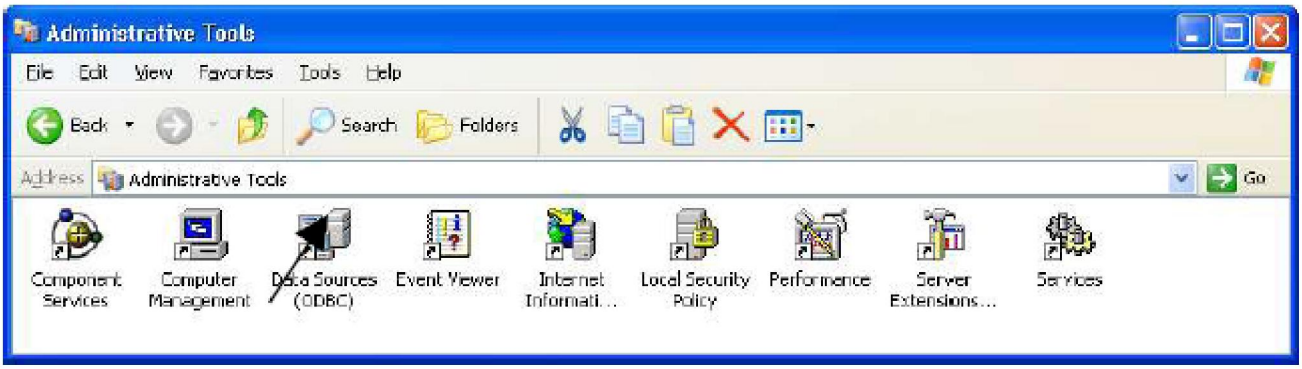
TABEL T_KULIAH
Kode_Kuliah (Number)
Mata_Kuliah (Text/20)
SKS (Number)
Ruang (Text/5)
Dosen (Number)

TABEL NILAI
Kode_Kuliah (Number)
Kuliah (Text/20)
Mahasiswa (Number)
Nilai (Number)



Gambar 6.19 *Interface* dari Control Panel

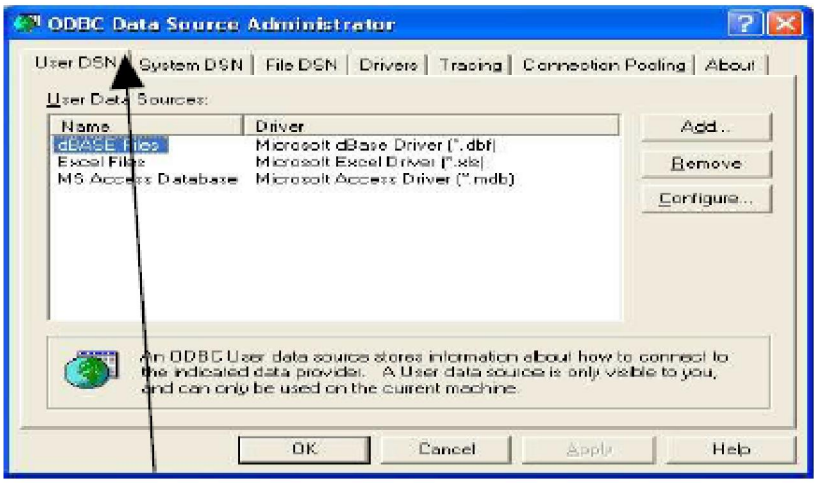
○ Pilih Menu Administrative Tools (Untuk Windows 2000 Prof, XP)



Gambar 6.20 Tampilan menu *Administrative Tools* (Untuk Windows 2000 Prof, XP)

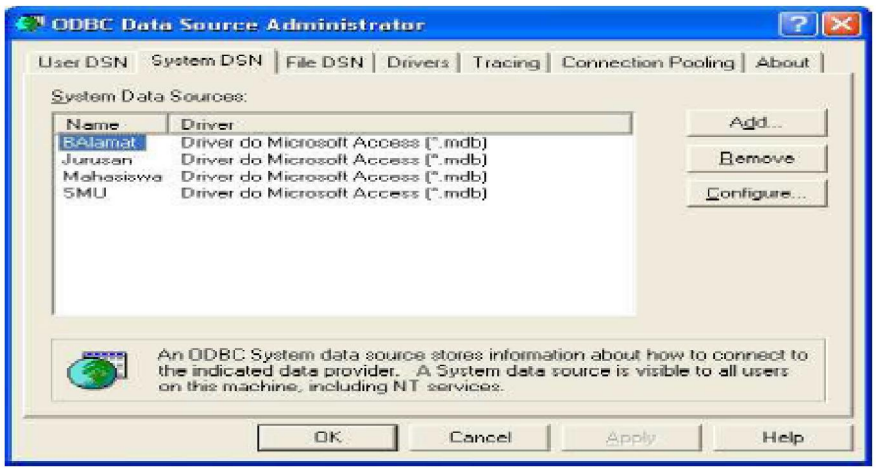
○ Lalu pilih ODBC Data Source

○ Untuk Windows 98 dari Control Panel langsung Pilih Menu ODBC Data



Gambar 6.21 Tampilan menu *ODBC*

○ Pilih Tab System DSN



Gambar 6.22 Tampilan *tab System* DSN

- Pilih Add untuk membuat DSN baru

Gambar 6.23 Tampilan pembuatan DNS baru

- Pilih Driver Do Microsoft Acces (*.mdb)
- Lalu pilih dan tekan tombol Finish maka akan muncul menu sbb :

Gambar 6.24 Tampilan menu ODBC pada Microsoft Acess

- Isilah Nama *Data Source* (Nama Bebas) Nama data *source* inilah yang nantinya digunakan untuk menghubungkan ke *Database*. Contoh Nama Jur_Pemasaran
- Isilah *Description* (bebas) yaitu keterangan Nama DSN
- Lalu Pilih *Database* > Tombol *Select* dan akan muncul tampilan

Gambar 6.25 Tampilan menu *Select Database*

- Pilih Database yang telah anda buat misalkan di sini namanya Pemasaran.mdb yang diletakkan pada directori C:\Apache\Acces\ pemasaran.mdb
- Pilih OK
- Sudah Selesai dan nanti Database Name akan muncul pada menu berikut:

Gambar 6.26 Tampilan menu *ODBC Data Source Administrator*

6.8. Mengakases Database Melalui PHP

PHP telah menyediakan sejumlah fungsi untuk mengakses database melalui ODBC. Dengan fungsi-fungsi ODBC ini maka akses ke database akan menjadi mudah dan cepat. Untuk dapat mengakses database maka harus dimengerti fungsi-fungsi berikut ini :

Dengan fungsi-fungsi ODBC ini maka akses ke Database akan menjadi lebih mudah dan cepat.

Fungsi odbc_connect

Fungsi ini berguna untuk menghubungkan nama sumber data ODBC dengan format pemanggilan sebagai berikut:

```
odbc_connect (nama_sumber_data, nama_pemakai,
nama_password)
```

Fungsi odbc_close

Fungsi ini berguna untuk menutup hubungan dengan sumber data ODBC. Format penulisannya sebagai berikut:

```
odbc_close (pengenal_hubungan) pengenal_hubungan
adalah nilai yang dihasilkan oleh odbc_connect
```

Fungsi odbc_close_all

Fungsi ini berguna untuk menutup semua hubungan dengan sumber data ODBC. Format penulisannya sebagai berikut:

```
odbc_close_all ( )
```

Fungsi odbc_do

Fungsi ini berguna untuk menjalankan perintah sql dengan format penulisan sebagai berikut:

```
odbc_do (pengenal_hubungan, string_query)
```

Fungsi odbc_num_fields

Berguna untuk memperoleh jumlah field

```
odbc_num_fields (pengenal_hasil)
```

Fungsi odbc_num_rows

Berguna untuk memperoleh jumlah record atau baris

```
odbc_num_rows (pengenal_hasil)
```

Fungsi odbc_fields_name

```
odbc_fields_name (pengenal_hasil, no_kolom)
```

Fungsi odbc_ fields_len

Berguna untuk memperoleh panjang kolom

```
odbc_ fields_len (pengenal_hasil, no_kolom)
```

Fungsi odbc_ fields_type

Berguna untuk memperoleh tipe kolom,

```
odbc_ fields_type (pengenal_hasil, no_kolom)
```

Fungsi odbc_ fetch_row

Berguna untuk memperoleh sebuah baris dari suatu himpunan hasil.

```
odbc_ fetch_row (pengenal_hasil, no_baris)
```

Fungsi odbc_ fetch_into

Berguna untuk memperoleh sebuah baris dari suatu himpunan hasil.

```
odbc_ fetch_into (pengenal_hasil, no_baris)
```

Fungsi odbc_ result

Berguna untuk memperoleh nilai dari sebuah kolom pada baris sekarang.

```
 odbc_ result (pengenal_hasil, kolom)
```

Fungsi odbc_ result_all

Berguna untuk menampilkan semua baris dari suatu hasil ke browser.

```
odbc_ result_all (pengenal_hasil, format)
```

Open Database Connectivity (disingkat menjadi ODBC) adalah sebuah standar terbuka untuk konektivitas antar mesin basis data. Standar ini menyediakan API yang dapat digunakan untuk menjalankan dan mengoneksikan sebuah aplikasi dengan sebuah sistem manajemen basis data (SMBD). Para desainer ODBC membuatnya dengan tujuan agar ODBC terbebas dari penggunaan bahasa pemrograman tertentu, sistem manajemen basis data tertentu, dan sistem operasi tertentu. (Sumber https://id.wikipedia.org)

Latihan 1. Membuat Tes Koneksi ke ODBC

Untuk mengetahui apakah hubungan ke database sudah berjalan atau belum dapat dibuat dengan script PHP dengan fungsi odbc_open dan odbc_close sebagai berikut:

```
<HTML>
<TITLE> Tes Koneksi Ke Database Dengan ODBC </
TITLE>
<BODY>
<?php
$pemakai = "";
$password = "";
$dataku = odbc_connect("Jur_Pemasaran",
$pemakai,
$password);
if (! $dataku)
die("Database tak dapat dibuka");
odbc_close($dataku);
print("Alhamdulillah Berhasil Oi");
?>
</BODY>
</HTML>
```

Jika dijalankan dalam Browser Internet Explorer maka hasilnya sebagai berikut:

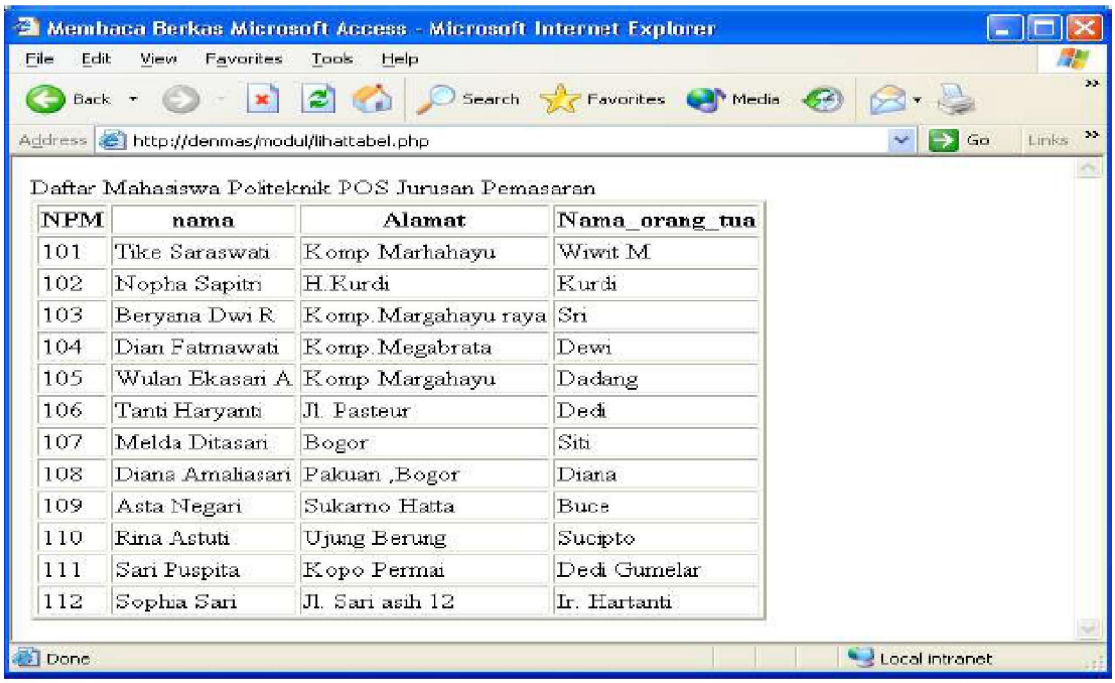
Gambar 6.27 Tampilan dalam *Browser Internet Explorer*

Latihan 2. Melihat Daftar Tabel Mahasiswa

```
<HTML>
<HEAD>
<TITLE>Membaca Tabel Mahasiswa </TITLE>
</HEAD>
<BODY>
<align = "center">
</p>
Daftar Mahasiswa Politeknik POS Jurusan Pemasaran
<?php
$pemakai = "";
```

```
$password = "";
$dataku = odbc_connect("Jur_Pemasaran",
$pemakai,
$password);
if (! $dataku)
die("Database tak dapat dibuka");
$sql = "SELECT Mahasiswa.NPM, Mahasiswa.nama,
Mahasiswa.Alamat,
Mahasiswa.Nama_orang_tua " .
"FROM Mahasiswa " .
"ORDER BY Mahasiswa.NPM";
$hasil = odbc_do($dataku, $sql);
odbc_result_all($hasil, "Border=2");
odbc_close($basis_data);
?>
</BODY>
</HTML>
```

Jika dijalankan dalam *Browser Internet Explorer* maka hasilnya sebagai berikut:



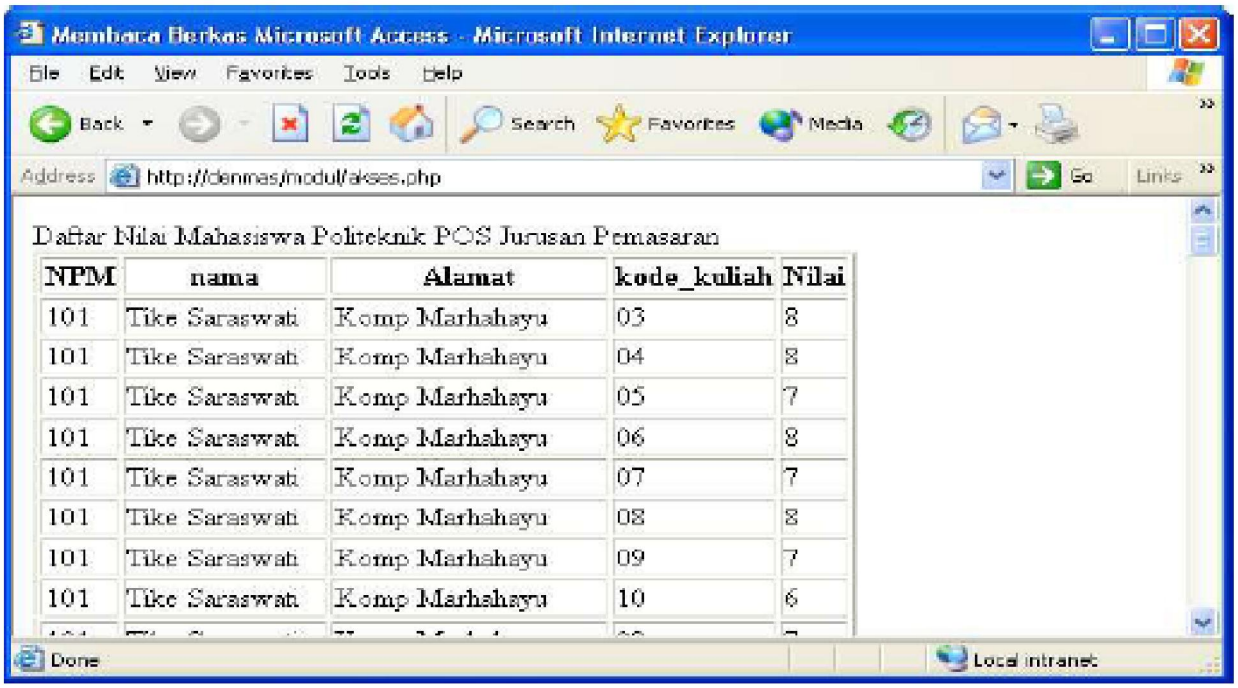
Gambar 6.28 Tampilan dalam *Browser Internet Explorer*

Latihan 3. Melihat Daftar Tabel Mahasiswa Dan Tabel Nilai

```
<HTML>
<HEAD>
<TITLE>Membaca Berkas Microsoft Access</TITLE>
</HEAD>
```

```
<BODY>
Daftar Nilai Mahasiswa Politeknik POS
Jurusan Pemasaran
<?php
$pemakai = "";
$password = "";
$dataku = odbc_connect("Jur_Pemasaran", $pemakai,
$password);
if (! $dataku)
die("Database tak dapat dibuka");
$sql = "SELECT Mahasiswa.NPM, " .
"Mahasiswa.nama, Mahasiswa.Alamat,
Nilai.kode_kuliah,
Nilai.Nilai " .
"FROM Mahasiswa, Nilai " .
"WHERE Mahasiswa.NPM = Nilai.NPM " .
"ORDER BY Mahasiswa.NPM";
$hasil = odbc_do($dataku, $sql);
odbc_result_all($hasil, "Border=2");
odbc_close($dataku);
?>
</BODY>
</HTML>
```

Jika dijalankan dalam Browser Internet Explorer maka hasilnya sebagai berikut:



Gambar 6.29 Tampilan hasil saat dijalankan dalam *Browser Internet Explorer*

Tampilan di atas merupakan gabungan antara tabel Mahasiswa dan tabel Nilai

Latihan 4. Melihat Daftar Tabel Mahasiswa, Tabel Nilai, Tabel Kuliah Dan Dosen

```
<HTML>
<HEAD>
<TITLE>Membaca Berkas Microsoft Access</TITLE>
</HEAD>
<BODY>
Daftar Nilai Mahasiswa Politeknik POS
Jurusan Pemasaran
<?php
$pemakai = "";
$password = "";
$dataku = odbc_connect("Jur_Pemasaran",
$pemakai,
$password);
if (! $dataku)
die("Database tak dapat dibuka");
$sql = "SELECT Mahasiswa.NPM, Mahasiswa.nama, " .
"Nilai.kode_kuliah, T_Kuliah.Nama_Kuliah,
Nilai.Nilai, Dosen.Nama " .
"FROM Mahasiswa, Nilai, T_Kuliah, Dosen " .
"WHERE Mahasiswa.NPM = Nilai.NPM " .
"and T_Kuliah.kode_kuliah = Nilai.kode_kuliah " .
"and Dosen.NIP = T_kuliah.NIP " .
"ORDER BY Mahasiswa.NPM";
$hasil = odbc_do($dataku, $sql);
odbc_result_all($hasil, "Border=2");
odbc_close($dataku);
?>
</BODY>
</HTML>
```

Jika dijalankan dalam *Browser Internet Explorer* maka hasilnya sebagai berikut:

Gambar 6.30 Tampilan hasil saat dijalankan dalam *Browser Internet Explorer*

Latihan 5. Penambahan Data

```
<html>
<head>
<title>
Contoh Penambahan Data Melalui Web
</title>
</head>
<body>
<?
$pemakai = "";
$password = "";
$dataku = odbc_connect("Jur_Pemasaran", $pemakai,
$password);
if (! $dataku)
die("Database tak dapat dibuka");
$sql="insert into Mahasiswa " .
" (NPM, Nama, Alamat, Tgl_Lahir, Nama_Orang_Tua) " .
" VALUES (115, 'Denmas', 'Jl. Sekeloa Utara 62', 11/
27/1975, 'Ir. Ali Akbar')";
if (odbc_do ($basis_data, $sql))
printf(" Data Berhasil Diinputkan ");
else
printf(" Data Tidak Berhasil Diinputkan ");
odbc_close($dataku);
?>
</body>
</html>
```

Jika dijalankan dalam Browser Internet Explorer maka hasilnya sebagai berikut:

Gambar 6.31 Tampilan hasil saat dijalankan

Reynold B. Johnson | Penemu Hardisk Komputer

Reynold B. Johnson

Nama Lengkap:
Reynold Berners Johnson

Tempat tanggal lahir:
Minnesota - Amerika Serikat,
16 Juli 1906

Tempat tinggal: Michigan-
Amerika Serikat
Kebangsaan: Amerika Serikat
Kuliah : University of Minnesota
Pekerjaan: Ilmuwan komputer
Tempat kerja:
IBM

Hasil penemuan:
Penemu Hardisk Komputer

(Sumber: https://www.penemu.co)

Reynold B. Johnson lahir di negara bagian Minnesota, Amerika Serikat, tahun 1906. ia merupakan penemu dari hardisk dan disebut sebagai "father of disk drive". Sesungguhnya, masa kecil dan masa remaja Reynold jauh dari urusan komputer. Di situ pula ia makan bangku sekolahan. Kuliah di University of Minnesota, dia mengambil jurusan administrasi pendidikan. Setelah menuntaskan gelar sarjananya tahun 1929, Johnson mengajar di sebuah sekolah menengah di Ironwood, Michigan. Di sana jadi guru ilmu pengetahuan alam dan matematika. Sekuat tenaga, Pak guru Johnson mencari cara agar matematika yang rumus-rumusnya bikin mumet para murid, bisa lebih mudah dimengerti.

Pernah juga di suatu saat, dua orang siswa tertangkap basah mencuri sebuah radio sekolah. Keduanya dijatuhi hukuman. Selama masa hukuman itu mereka dibimbing Johnson. Sang guru mengubah hukuman itu menjadi sesuatu yang berguna. Saban hari Reynold membimbing keduanya membuat alat eksperimen. Hingga di akhir masa hukuman, dua murid ini sukses menciptakan alat untuk memeriksa lembar ujian yang menggunakan sistem pilihan berganda alias multiple choice. Meski lihai mendidik murid, karir mendidik Johnson gagal total di tengah jalan. Empat tahun mengajar di sekolah, sang guru malah dipecat. Peristiwa inilah yang tak hanya mengubah hidup Pak Guru Johnson, tapi juga mengubah sejarah sistem penyimpan data. Johnson memanfaatkan waktu luangnya untuk mematangkan mesin pemeriksa ujian multiple choice tadi. Hasilnya digadang ke beberapa sejumlah perusahaan, termasuk IBM. Akhirnya, hanya setahun menjadi pengangguran, pada tahun 1934, Johnson diangkat menjadi menjadi engineer di IBM.

Ia ditempatkan di laboratorium IBM di Columbia University dan Endicott, di New York. Tugasnya cuma satu: mengembangkan mesin pemeriksa lembar ujian multiple choice. Mesin ini mampu membaca goresan pensil di lembar jawaban, lalu secara otomatis menghitung dan membeberkan hasil akhir penilaian secara visual. Ide membuat mesin ini memang tidak datang begitu saja. Sejak bocah, Johnson tahu betul tentang sifat listrik dari pensil nomor 2 (semacam pensil 2B), yang sering digunakan oleh murid-murid untuk ujian. Johnson ingusan, suka iseng menghidupkan mesin mobil Ford T milik kakak perempuannya, dengan cara menggosok-gosokan busi dengan pensil tadi. Dari situlah, Johnson mulai menyadari kandungan magnetik dari pensil itu, yang kemudian bisa dideteksi oleh mesin pemeriksa lembar ujian ciptaannya. Mesin yang diberi nama IBM 805 itu, kemudian diproduksi hingga ribuan unit pada 1937. Setidaknya, hingga dekade 1970-an, IBM 805 masih digunakan secara luas.

Episode berikutnya dari karir Johnson, bermula saat para petinggi IBM menyadari kebutuhan terhadap sistem penyimpanan data yang bakal meningkat pesat. Itu sebabnya, tahun 1952, perusahaan itu mengirim Johnson ke San Jose, California untuk membentuk tim yang diberi tugas yang maha berat. Mencari sebuah sistem penyimpanan data yang cepat dan handal. Di tempat barunya, Johnson melanjutkan penelitian yang telah dimulai oleh tim riset IBM di Columbia. Salah satunya adalah mesin penyimpanan Random Access Memory (RAM). Sistem ini memungkinkan pengguna mengakses dan mencari data dengan lebih cepat. Di masa itu, biasanya data disimpan di kartu plong atau pita magnetik (magnetic tape). Cara ini amat njelimet. Memerlukan waktu bermenit-menit mencari data yang diinginkan. Awalnya, Johnson yang dibantu 15 orang stafnya, memusatkan diri pada pengembangan sistem penyimpanan berbentuk silinder alias rotating drum. Pada masa itu, media penyimpanan model begini sedang luas dikembangkan.

Bentuk Hardisk Pertama

Belakangan, tim ini lebih tertarik pada penyimpanan berbentuk piringan cakram yang dikembangkan oleh Jacob Rabinow dari National Bureau Standard. Alasannya sederhana. Di benaknya, penyimpanan cakram lebih mudah dikembangkan, lebih dapat diandalkan, dan juga irit biaya. Itu sebabnya semua tenaga dikerahkan ke proyek ini. Hasilnya luar biasa. Di ujung tahun 1955, mereka sukses membuat penyimpan cakram keras yang diberi nama IBM 350. Cakram keras ini merupakan bagian dari sistem komputer bernama RAMAC 305 (bedakan dengan sistem penyimpannya, yang bernama IBM 305). RAMAC sendiri merupakan kependekan dari Random Access Method of Accounting Control, yaitu sistem yang mampu menyediakan akses data yang cepat bagi penggunanya di masa itu. Tidak seperti cakram keras yang dijumpai di pasaran saat ini, IBM 350 itu jauh lebih bongsor. Besarnya setara dengan dua buah kulkas yang berderet. Bobotnya pun tak tanggung-tanggung, satu ton. Itu karena, ia terdiri 50 buah cakram berdiagonal 24 inci (sekitar 60 cm), yang ditumpuk-tumpuk. Kapasitas penyimpanannya? Sekitar 5 megabyte data.

Bentuk Hardisk Pertama

menyimpan file musik berformat MP3, IBM 350 hanya bisa memuat satu file lagu, yang kurang lebih, berdurasi 5 menit. Coba bandingkan dengan cakram keras teranyar dari Seagate, yang baru diluncurkan pertengahan tahun ini. Dengan ukuran 3,5 inci (sekitar 9 cm), ia sudah mampu memuat data sebesar 1,5 terabyte, atau 314.000 kali lebih banyak dari kapasitas penyimpanan IBM 350. Walau hanya mampu menyimpan 5 megabyte data, tapi kapasitas sebesar itu merupakan lompatan yang fenomenal di masanya. Piringan cakram IBM 350 yang dibuat Johnson, mampu berputar sekitar 1200 putaran per detik (RPM). Ia memiliki dua head (piranti pembaca data) untuk membaca dan menulis data (read/write) dalam hitungan detik atau milidetik. Akhirnya, pada 1956, IBM 350 mulai diperkenalkan sebagai cakram keras komersial pertama di dunia.

Johnson pernah sedikit mengenang masa-masa yang menentukan saat bekerja di IBM. Menurutnya, ketika itu, ia mengambil dua keputusan yang paling tepat dalam hidupnya. Pertama, keputusan untuk mendalami riset di bidang penyimpanan random-access. Yang kedua, saat ia memutuskan untuk berkonsentrasi pada riset penyimpanan data pada media cakram. "Belakangan, saya mengetahui bahwa beberapa orang di negara ini (Amerika Serikat) juga mengembangkan penelitian yang sama dengan konsep yang saya miliki. Hanya saja, mereka melakukan penelitian di berbagai media lain, seperti kabel, gulungan pita, dan lain-lain," ujar Johnson.

Kesuksesan Johnson tak semata-mata karena kejelian visinya. Menurut Jack Kuehler --orang yang mengangkat Johnson menjadi karyawan dan belakangan menjadi President IBM-- memberikan kesaksian tentang sikap hangat yang dimiliki Johnson. "Johnson adalah orang yang selalu mencari ide-ide baru. Saat berada di timnya, kau akan merasa bahwa dia adalah figur seorang ayah, yang akan menjadi tutor dan selalu bersedia membantu. Kau akan merasa bahwa ia bekerja lebih baik, dan akan menantangmu untuk mencapai konsep dan kreativitas yang lebih tinggi dari sebelumnya," ujar Jack. Tentang pribadi ayahnya, salah satu anak Johnson, Philip, berkomentar bahwa Johnson adalah ayah yang lembut, baik hati, dan tak suka pamer. "Dari sekian banyak penemuan yang ia capai, ia tidak pernah berusaha menonjolkan dirinya," kata Philip. Johnson pensiun dari IBM pada tahun 1971, dan mendirikan sebuah perusahaan konsultan Education Engineering Associates, yang berlokasi di Palo Alto. Di perusahaannya itu, Johnson sempat mengembangkan teknologi micronograph yang memungkinkan seseorang untuk mendengarkan rekaman audio melalui sebuah tulisan atau gambar di buku.

Johnson, bekerja sama dengan Sony mengembangkan teknologi penyimpanan video yang dapat memuat data dua kali lipat lebih besar dibandingkan pita video normal saat itu. Penemuan ini kemudian juga menjadi cikal bakal teknologi Video Camera Recorder. Atas berbagai jasanya, pada tahun 1986, Johnson mendapat banyak penghargaan. Termasuk di antaranya penghargaan National Medal of Science dari Presiden Amerika Serikat Ronald Reagan. Selasa, 15 September 1998, Johnson tutup usia di umurnya yang 92 tahun. Dia meninggalkan seorang istri bernama Beatrice, dan dua orang putra, Philip dan David. Juga meninggalkan warisan yang memudahkan Anda menyimpan data, Hard Disk Drive. biografiku.comlah membuat kita pusing tujuh keliling. Sehingga Reynold B. Johnson mencari cara agar rumus-rumus Matematika ini dapat dengan mudah di pahami oleh para murid muridnya.

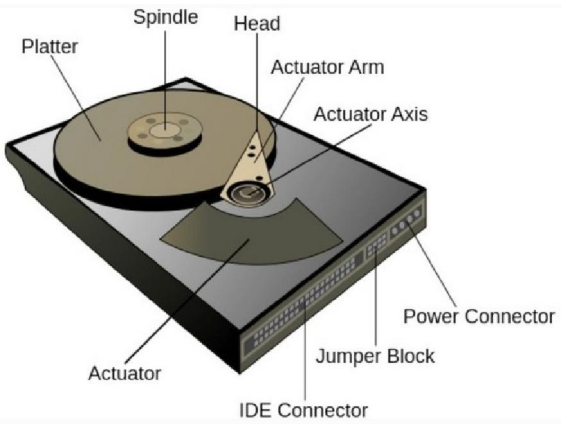
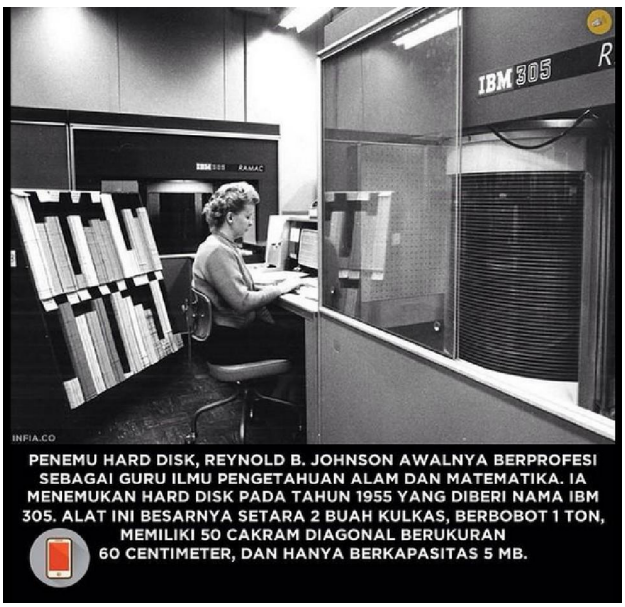


Diagram komponen utama cakram ...Diagram komponen utama cakram keras *(Surachit/Wikimedia Commons)*



(Sumber: http://www.biografiku.com/; https://www.wikidata.org/wiki/Q1297016; http://nationalgeographic.co.id/berita/2014/10/penemu-hard-disk-dulunya-seorang-guru)

Rangkuman

1. Fungsi mysql_connect digunakan untuk membuat hubungan ke databse MySQl yang terdapat pada suatu host
2. Untuk membuat database di mysql sebaiknya gunakan PhpMyadmin. Jika anda menggunakan phptriad untuk mengakses phpmyadmin, pada browser ketikkan http://localhost/phpmyadmin
3. Sebuah database tentunya terdapat revisi-revisi atau perubahan isi tabel atau record. Maka harusnya terdapat sebuah form yang mampu mengedit isi yang terdapat dalam database
4. Database yang besar yang tentunya mempunyai record ribuan atau bahkan ratusan ribu yang seharusnya mempunyai fasilitas pencarian data atau searching

Daftar Istilah

1. MySQL
2. Browse tabel
3. Localhost
4. mysql_query
5. mysql_db_query
6. mysql_affected_rows
7. mysql_fetch_row

Latihan Soal

1. Sebutkan dan jelaskan fungsi – fungsi dalam MySQL yang digunakan untuk mengakses database MySQL menggunakan PHP!
2. Mengapa PHP dijadikan sebagai salah satu alat dalam akses web ?
3. Jelaskan pengertian dari ODBC?
4. Sebutkan dan jelaskan fungsi – fungsi dalam ODBC !
5. Jelaskan satu fungsi yang paling krusial dari ODBC !

KASUS PADA UKM : MEMBANGUN JARINGAN USAHA

Harus disadari bahwa keberadaan usaha mikro, kecil dan menengah (UMKM) sebagian integral dari pembangunan nasional tidak bisa diabaikan. Krisis moneter yang berawal pada pertengahan Juli1997 dan merembet pada Krisis ekonomi, politik dan sosial atau lebih dikenal dengan nama krisis multidimensi telah membawa berkah tersendiri bagi kehidupan UMKM. Kalau di masa Orde Baru kehadiran UKM sering dipinggirkan, kini telah diakui oleh semua pihak, khususnya pemerintah, untuk mendapatkan perhatian khusus. Kebijakan masa lalu yang beroientasi untuk mengejar pertumbuhan ekonomi tanpa memperhatikan pemerataan hasil-hasil pembangunan telah dirasakan keliru. Orientasi pada pertumbuhan tersebut, ternyata tidak berhasil mengurangi jurang kesenjangan antara yang kaya dengan yang miskin, antara wilayah barat dengan wilayah timur atau antara pelaku bisnis kecil - baca pengusaha mikro kecil dan menengah dengan pelaku bisnis besar-baca konglomerat. Kisah masa lalu dan krisis multidimensi yang terjadi selama ini telah membawa perubahan mendasar pada paradigma pembangunan nasional.

Pemusatan ekonomi tidak lagi menjadi prioritas; justru sebaliknya kesempatan yang seluas-luasnya harus diberikan kepada semuanya tanpa harus memandang suku, agama, ras, dan golongan. Ini berarti kesempatan bagi berkembangnya UMKM semakin terbuka. UMKM yang menguasai 99,99% dari seluruh usaha nasional harus diberdayakan agar mampu berperan lebih besar dalam perekonomian nasional. Mereka, yang kini jumlahnya besar tersebut mampu menyerap lebih dari 99% angkatan kerja saat ini harus mampu memberikan kontribusi yang besar juga dalam ekspor dan Produk Domestik Bruto (PDB).

Saat ini mereka masih menyumbang tidak lebih dari 20% dalam ekspor dan 59.36% dalam PDB. Kalau dilihat dari perjalanan negara maju seperti Amerika Serikat, Jepang, Italia dan Taiwan yang sebagian besar ekonominya di dukung oleh UMKM, maka kita harus berupaya mengembangkan UKM agar mampu berkiprah lebih besar dalam perekonomian nasional. Pada negara-negara maju tersebut diatas, peran UMKMnya sudah melebihi peran Usaha Besar. Dalam ekspor, misalnya sudah lebih dari 55% merupakan kontribusi UMKM di negara-negara tersebut. Bahkan di Italia, peran UKM dalam ekspor sudah mencapai lebih dari 75%. Pertanyaan yang mungkin muncul adalah apakah hal ini bisa dilakukan oleh UMKM Indonesia? Untuk menjawab pertanyaan tersebut, kita harus optimis karena kita memiliki potensi sangat besar untuk dikembangkan oleh UMKM.

UMKM harus didorong mengembangkan usahanya dengan berdasarkan sumberdaya lokal dan berorientasi ekspor. Dengan strategi pengembangan usaha berorintasi ekspor dengan menggali dari potensi sumberdaya lokal yang tersedia, khususnya pada sektor agrobisnis/agroindustri termasuk kelautan dan perikanan berarti kita telah mengembangkan "comparative advantage" untuk dikembangkan menjadi "competitive advantage".

(Sumber : http://microentrepreneurshipcommunity.blogspot.com /2012 /04/ membangun jaringan-usaha-bagi-usaha.html)

BAB 7

Pemahaman Tentang Upload File

Tujuan Instruksional Umum

Setelah mempelajari bab ini, Mahasiswa dipastikan dapat:

1. Menjelaskan Langkah – Langkah Melakukan Upload
2. Menjelaskan Cara Membatasi Ukuran File

BAB

7

PEMAHAMAN TENTANG *UPLOAD FILE*

7.1. Melakukan *Upload*

Bila kita membuat aplikasi web yang memungkinkan pemakai memasukkan data dai mana saja, adakalanya kita juga menghendaki agar pemakai bisa mengirim berkas/file yang berada pada tempat ke server. Proses seperti ini dikenal dengan istilah **upload.** Hal ini bisa ditangani dengan menggunakan fasilitas formulir pada HTML, yaitu melalui tag <INPUT> dengan tipe "FILE". Dalam hal ini kita juga perlu menyebutkan atribut ENCTYPE dengan isi berupa "MULTIPART/FORM-DATA" pada <FORM> Sebagai contoh:

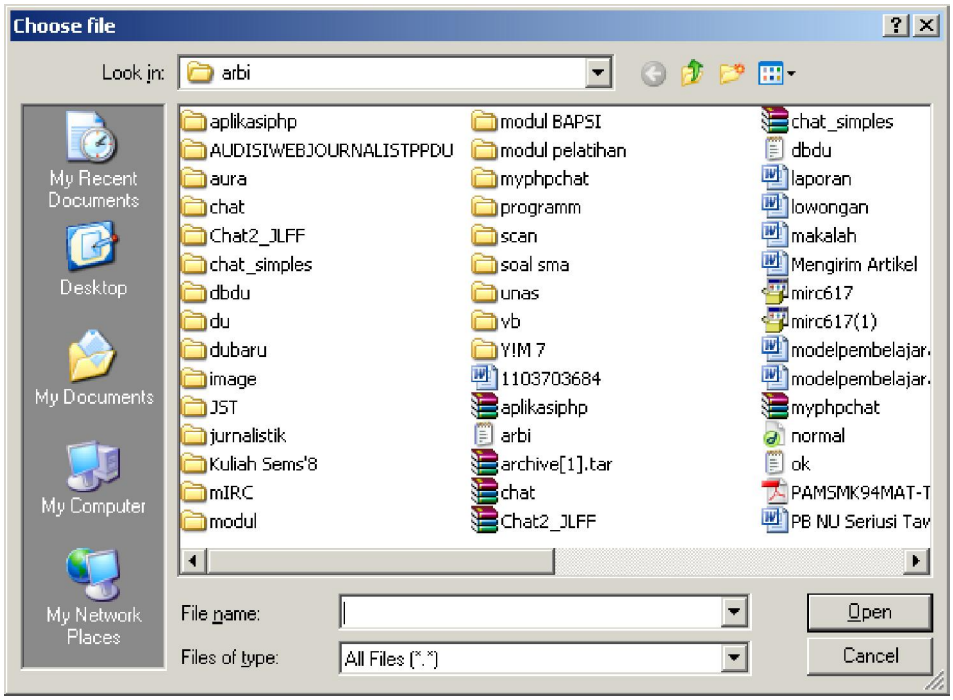
Script **upload.html**

```
<html>
<head>
<title>Upload File</title>
</head>
<body>
<form action="upload.php" method="post"
enctype="multipart/form-data">
Nama File :
<input type="file" name="namafile">
<input type="submit" value="Upload">
</form>
</body>
</html>
```

Perhatikan bahwa tag <FORM> pada script diatas mengandung atribut ENCTYPE yang berisi "MULTIPART/FORM-DATA". Pada tag ini, ACTION (yang menyatakan file akan dijalankan sekiranya tombol Subnit diklik) diisi dengan file upload.php, yang isinya akan segera Anda lihat.

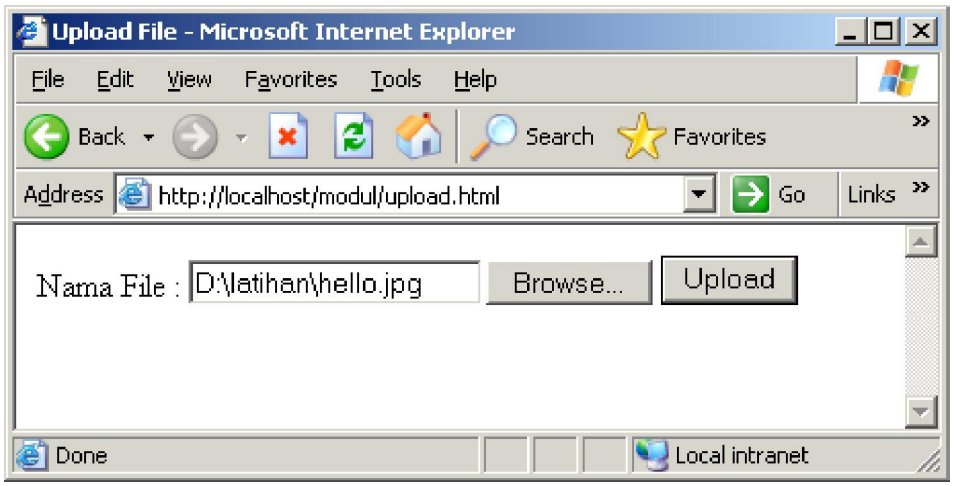
```
<input type="file" name="namafile">
```

Adapun perintah diatas adalah perintah untuk menangani penentuan file oleh pemilih. Bila Anda memanggil file upload.html di atas maka akan memperoleh tampilan sebagai berikut:



Gambar 7.1 Tampilan *Choice File*

Anda bisa segera memilih file yang Anda inginkan. Berikut adalah tampilan setelah sebuh file dipilih



Gambar 7.2 Tampilan *Upload File*

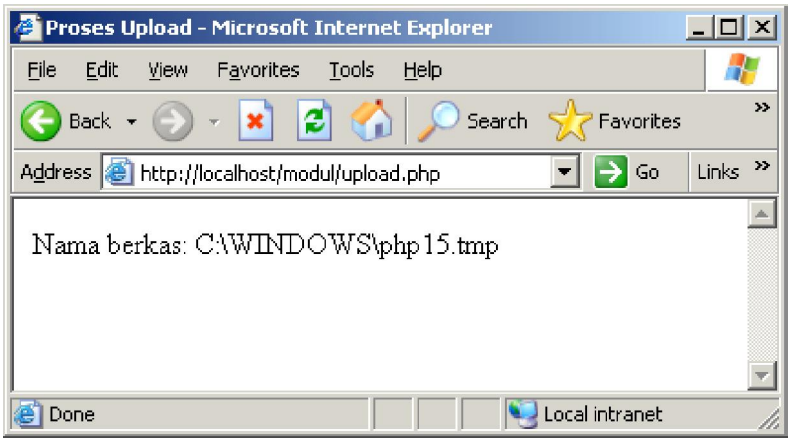
Berikut adalah script upload.php untuk melengkapi script upload.php

```
<HTML>
<HEAD>
<TITLE>Proses Upload</TITLE>
```

```
</HEAD>
<BODY>
<?php
  print("Nama berkas: $namafile<BR>\n");
?>
</BODY>
</HTML>
```

Pada script ini, kode PHP digunakan untuk menampilkan tulisan "Nama Berkas:" diikuti dengan isi variable **namafile.** Sebagaimana diketahui, **namafile** adalah nama variable yang disebutkan pada upload.html untuk menyimpan nama file pada computer klien yang akan di uploadkan ke server.

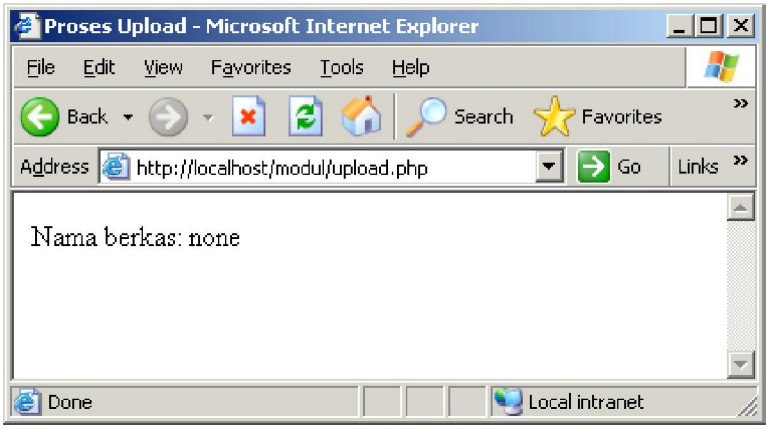
Berikut adalah hasil setelah tombol Upload di-klik:



Gambar 7.3 Tampilan proses *Upload File*

Dalam hal ini PHP akan melakukan penyalinan terhadap file tersebut ke suatu direktori sementara pada server. Direktori ini dapat diatur melalui direktif **upload_tmp_dir** yang terdapat pada file PHP.INI. Nama file yang digunakan untuk menyalin file berbentuk PHP diikuti suatu nilai yang dihasilkan oleh PHP.

Sekiranya file tidak diupload (misalnya tidak ada file yang dipilih ketika pemakai mengklik Upload), akan diperoleh string none pada variable yang disebutkan dalam tag INPUT bertipe FILE. Contohnya dapat dilihat di bawah ini:



Gambar 7.4 Tampilan proses *Upload.php*

7.2. Membatasi Ukuran File

Ukuran file yang dapat di upload bergantung pada direktif upload_max_files yang terdapat pada file PHP.INI. Tentu saja kita bias secara langsung melakukan perubahan pada file tersebut. Namun demikian ada cara lain yang dapat dilakukan pada kode HTML, yaitu dengan menambah baris seperti berikut:

```
<input type=”hidden” name=”max_file_size” value=”1024">
```

Yang terpenting, kode diatas perlu diletakan sebelum INPUT yang bertipe FILE. Sebagai informasi tambahan yang perlu diketahui, 1024 pada VALUE menyatakan ukuran file terbesar yang dapat disalin ke server. Satuannya adalah byte. Kode lengkap yang melibatkan max_file_size dapat dilihat berikut ini:

1024 pada VALUE menyatakan ukuran file terbesar yang dapat disalin ke server. Satuannya adalah byte.

Upload2.html

```
<html>
<head>
<title>Upload File</title>
</head>
<body>
<form action=”upload.php” method=”post”
enctype=”multipart/form-data”>
<input type=”hidden” name=”max_file_size”
value=”1024">
Nama File :
<input type=”file” name=”namafile”>
<input type=”submit” value=”Upload”>
</form>
</body>
</html>
```

Bila ukuran file yang akan diupload lebih dari nilai pada max_file_Size, variable yang disebutkan pada INPUT bertipe FILE akan berisi string none.

7.3. Mengenal Variabel Global Berakhiran _name, _size, _type

Script yang dipanggil oleh formulir yang mengandung INPUT bertipe FILE akan mengenali variable global yang diawali dengan nama varibel yang disebutkan pada tag INPUT bertipe FILE dan diakhiri dengan _name, _size, dan _type.

Dengan mengambil contoh pada skrip upload.php, skrip ini akan mengenal variabel – variabel dengan nama:

- Namafile
- Namafile_name
- Namafile_size

- Namafile_type

Berikut adalah penjelasan untuk keempat variable tersebut:

Tabel 7.1 Tabel variabel

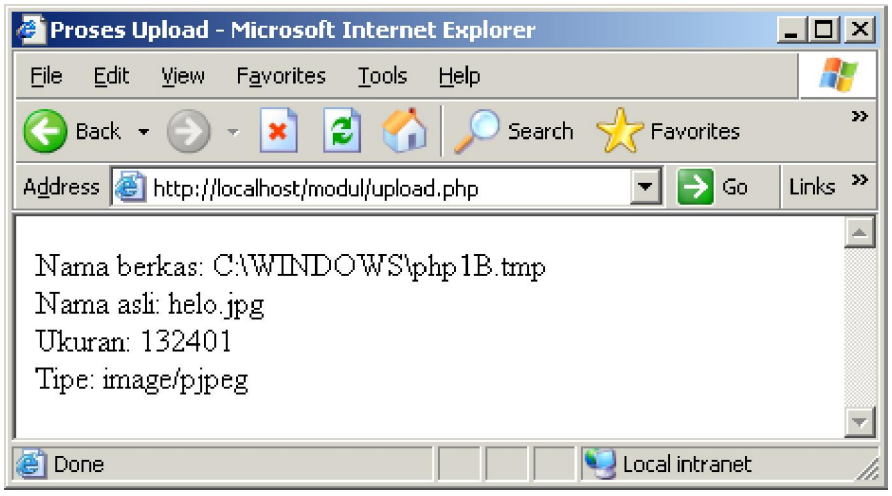
Variabel	Keterangan
Namafile	Namafile plus fpath dari file yang diupload (ada pada server pada direktori yang bersifat sementara)
Namafile_name	Nama file asli
Namafile_size	Ukuran file dalam satuan byte
Namafile_type	Tipe file

Berikut adalah skrip ulpload.php yang telah dilengkapi untuk menangkap informasi yang disebutkan di atas.

Upload3.php

```
<HTML>
<HEAD>
<TITLE>Proses Upload</TITLE>
</HEAD>
<BODY>
<?php
  print("Nama berkas: $namafile<BR>\n");
  print("Nama asli: $namafile_name<BR>\n");
  print("Ukuran: $namafile_size<BR>\n");
  print("Tipe: $namafile_type<BR>\n");
?>
</BODY>
</HTML>
```

Berikut adalah contoh hasilnya:



Gambar 7.5 Tampilan hasil dari proses *Upload.php*

7.4. Menangani Penyalinan Secara Permanen

Sebagaimana telah diterangkan, file yang diupload akan ditaruh pada direktori yang bersifat sementara. Supaya file ini tidak hilang, Anda perlu menyalinnya pada lokasi Anda sendiri. Cara penyalinannya ditunjukan pada skrip berikut:

Penyalinan dilakukan dengan menggunakan fungsi *copy*. Hasil salinan diletakan pada direktori image dibawah direktori web Anda

Upload4,php

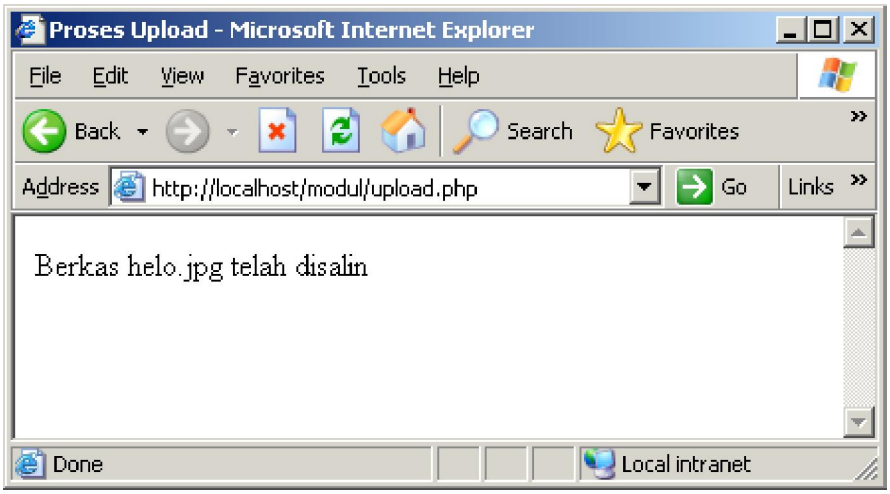
```
<HTML>
<HEAD>
<TITLE>Proses Upload</TITLE>
</HEAD>
<BODY>
<?php
  if ($namafile == "none")
     die ("Tak ada berkas yang di-upload");

  if (copy($namafile, "image/$namafile_name"))
  {
    print("Berkas $namafile_name telah disalin");
    unlink($namafile);
  }
  else
    print("Gagal menyalin $namafile_name");
?>
</BODY>
</HTML>
```

Pada skrip di atas:

```
if ($namafile == "none")
     die ("Tak ada berkas yang di-upload");
```

digunakan untuk menjajaki kemungkinan kalau tidak ada file yang diupload. Penyalinan dilakukan dengan menggunakan fungsi *copy*. Hasil salinan diletakan pada direktori image dibawah direktori web Anda (silahkan buat direktori ini lebih dahulu). Jika file berhasil disalin, maka akan muncul keterangan tentang hal itu dan kemudian file pada direktori sementara dihapus (melalui fungsi *unlink*).

Berikut adalah contoh hasil setelah file diupload.



Gambar 7.6 Tampilan hasil dari proses *Upload.php*

Marissa Ann Mayer | CEO Yahoo

(Sumber: http://1.bp.blogspot.com/-vjiCNMh3lII/UAm-UmcfiII/AAAAAAAACzo/9C7Z7j42dXU/s400/Marissa-Mayer.jpg)

Nama Lengkap:
Marissa Ann Mayer

Tempat dan Tanggal Lahir:
Wausau, Wisconsin, AS 30 Mei 1975
(umur 41)

Tempat tinggal:
San Francisco, Palo Alto

Kebangsaan:
Amerika Serikat
Almamater: Universitas Stanford
Pekerjaan: Presiden & CEO, Yahoo!
Instruktur pemrograman komputer, Universitas Stanford
Tempat kerja: Yahoo!
Anggota dewan dari Cooper–Hewitt, National Design Museum
New York City Ballet
San Francisco Ballet
San Francisco Museum of Modern Art
Walmart
Pasangan: Zachary Bogue (2009–sekarang)

(Sumber: https://id.wikipedia.org/wiki/Marissa_Mayer)

Marissa Ann Mayer dilahirkan pada tanggal 30 Mei 1975 di Wausau, Wisconsin. Ia adalah seorang eksekutif bisnis Amerika dan CEO Yahoo!. Sebelumnya, ia adalah seorang eksekutif dan juru bicara utama bagi Google. Dia adalah CEO termuda dari sebuah perusahaan Fortune 500. Mayer merupakan putri dari Margaret Mayer, seorang guru seni, dan Michael Mayer, seorang insinyur. Setelah lulus dari Sekolah Tinggi West Wausau pada tahun 1993, Mayer dipilih oleh Gubernur Wisconsin Tommy Thompson sebagai salah satu dari delegasi untuk menghadiri Science Camp Pemuda Nasional di West Virginia. ibu Mayer adalah keturunan Finlandia. Mayer lulus dengan pujian dari Stanford University dengan gelar BS dalam sistem simbolis dan M.S. dalam ilmu komputer.

Untuk kedua gelar tersebut, dia mengkhususkan diri dalam kecerdasan buatan. Pada tahun 2009, Illinois Institute of Technology memberikan Mayer sebuah gelar doktor honoris causa sebagai pengakuan atas karyanya di bidang sistem pencarian. Mayer bergabung dengan Google pada tahun 1999 yang jumlah karyawannya hanya 20 dan dia merupakan insinyur perempuan pertama perusahaan. Selama 13 tahun dia dengan perusahaan google, dia adalah seorang insinyur, desainer, manajer produk, dan seorang eksekutif. Di Google, Mayer bertanggung jawab untuk mengawasi peluncuran beberapa produk perusahaan yang paling ikonik, termasuk Gmail, Google Maps, dan iGoogle. Estetika desainnya yang tajam disebut-sebut sebagian menyebabkan keberhasilan dia di Google, dan membantu perusahaan ini tumbuh menjadi salah satu perusahaan teknologi dan Internet terbesar di planet ini. Warisan Mayer paling abadi di Google adalah nuansa minimalis dalam mesin pencariannya. Beda dengan mesin pencari lain yang tampilannya "padat", Google bernuansa minimalis, ruang putih yang cukup luas dengan warna biru cerah, merah, warna kuning dan hijau. Dialah yang paling ngotot agar tampilan mesin pencari Google itu "bersih" - yang tampaknya merupakan refleksi dari gayanya di luar kantor juga.

Mayer memegang peran kunci di Google Search, Google Images, Google News, Google Maps, Google Books, Google Product Search, Google Toolbar, iGoogle dan Gmail. Dia juga mengawasi tata letak terkenal homepage Google, cari tanpa hiasan. Pada tahun-tahun terakhirnya dengan Google, ia adalah Wakil Presiden Lokal, Maps, dan Layanan Lokasi dan, sebelum itu, Wakil Presiden Produk Pencarian dan User Pengalaman. Marissa Mayer boleh dikata saksi hidup perjalanan Google. Ia bergabung dengan perusahaan itu sejak masih berupa perusahaan start-up, belum menjadi titan Internet seperti sekarang. Seiring dengan naiknya Eric Schmidt sebagai CEO, Mayer menjadi salah satu eksekutif Google yang paling sering terlihat di muka publik dalam satu dekade terakhir. Ia kerap berbicara dalam konferensi teknologi, dan terlihat paling "bersinar" di antara insinyur muda bidang IT yang senang ber-jeans ria.

Biografi Marissa Mayer - CEO YahooSebelum bergabung dengan Google, Mayer bekerja di lab riset UBS (Ubilab) di Zurich, Swiss, dan SRI International di Menlo Park, California. Pada tanggal 16 Juli 2012, Mayer diangkat Presiden dan CEO Yahoo!, efektif pada hari berikutnya. Dia juga anggota dewan direktur perusahaan. Mayer adalah anggota dewan direktur dari Cooper-Hewitt, National Design Museum, New York City Ballet, San Francisco Ballet, San Francisco Museum of Modern Art dan Walmart. Mayer terpilih dalam daftar tahunan majalah Fortune tentang 50 Wanita Paling Berpengaruh dalam Bisnis pada tahun 2008, 2009, 2010 dan 2011. Dia menempati peringkat , masing-masing 50, 44, 42 dan 38. Pada tahun 2008, pada usia 33, dia adalah wanita termuda yang pernah terdaftar. Mayer terpilih sebagai salah satu dari Perempuan Glamour Magazine of the Year tahun 2009. Mayer menikah investor real estat komersial Zachary Bogue pada tanggal 12 Desember 2009. Pada hari yang sama Yahoo! mengumumkan mempekerjakan dia, Mayer mengungkapkan bahwa dia dan suaminya mengharapkan bayi laki-laki, karena 7 Oktober 2012.

Dikutip dari Slashgear, wanita berambut pirang tersebut akan memperoleh gaji sebesar 1 juta USD per tahunnya. Namun tentu saja gaji tersebut masih belum ditambah dengan bonus-bonus lain. Besaran bonus yang diterima oleh Mayer kemungkinan berkisar antara 2 juta USD hingga 4 juta USD. Sementara itu, menurut Wall Street Journal, Mayer yang kini tengah hamil tersebut akan memperoleh gaji sebesar 5.4 juta USD dalam satu tahun. Dan, tak menutup kemungkinan jika besaran gaji yang diterima oleh Mayer pada tahun depan akan meningkat hingga 20 juta USD jika performanya. Pundi-pundi uang yang diterima oleh Mayer dari Yahoo masih belum berhenti di situ. Yahoo juga membayar uang sebesar 14 juta USD sebagai kompensasi karena meninggalkan Google. Dia pun juga bakal menerima biaya retensi senilai 30 juta USD yang akan diberikan dalam jangka lima tahun. Dan, uang ini belum termasuk uang yang diterima oleh Mayer berkat dedikasinya kepada Google. Tentu saja uang tersebut juga bukan jumlah yang kecil.

Yahoo boleh dibilang sangat beruntung mendapatkan Marissa Mayer sebagai CEO baru mereka. Wanita berambut pirang ini adalah sosok yang dikenal cerdas dengan segudang prestasi selama berkecimpung di Google. Pengalamannya yang mentereng

tak bisa dipungkiri menjadi pertimbangan Yahoo memilih wanita berparas cantik ini memimpin perusahaan mereka. Dikutip dari Business Insider, Selasa (17/7/2012), setidaknya ada 4 alasan Marissa pantas dipilih, seperti diungkapkan narasumber yang dekat dengan jajaran direksi Yahoo.

Keberhasilan Marissa dalam menggapai impiannya tentunya tak bisa lepas dari beberapa aspek diantaranya:

1. **Fokus**

   Marissa dikenal sangat fokus pada produk dan pengalaman pengguna. Dia diyakini sangat mampu menggiring perusahaan kembali ke 'akar' produk dan konsumennya.

2. **Mentor yang baik**

   Di usianya yang ke-37, Marissa sudah mengenyam banyak pengalaman dalam membimbing orang-orang di timnya. Rekan-rekannya di Google menyebut Marissa adalah mentor yang baik, terutama bagi orang-orang baru untuk menggali bakat mereka. Di Google, dia menciptakan program khusus untuk melatih para Product Manager mengenai kepemimpinan eksekutif. Banyak dari mereka, kini menjadi manager di layanan terkenal seperti Uber, Dropbox dan Polyvore.

3. **Bertangan dingin**

   Sebagian besar orang mungkin mempertanyakan keputusannya hijrah ke Yahoo yang saat ini sedang dalam kondisi kurang baik. Namun Marissa yang bertangan dingin sepertinya malah tertantang mengubah keadaan tersebut. Didukung penuh jajaran direksi Yahoo untuk menjabat sebagai CEO, Marissa dipandang bisa dengan 'men-stok ulang' Yahoo dengan orang-orang penuh bakat.

4. **Populer**

   Di dunia teknologi, namanya memang sudah dikenal dengan sederet prestasi. Tak heran, kabar dipinangnya Marissa oleh Yahoo menjadi kejutan besar bagi publik. Lulusan Stanford University ini beberapa kali masuk daftar 50 Most Powerful Women in Business versi majalah Fortune, bahkan menjadi wanita termuda yang masuk dalam daftar tersebut.

Itu sebabnya, Co-founder Yahoo David Filo yang masih bekerja di sana sekaligus pemegang saham terbesar pun tak ragu memberi 'cap' persetujuan jabatan CEO bagi Marissa.[4] Fuegi and Francis 2003 pp. 19, 25.

(Sumber : http://www.biografiku.com/2012/07/biografi-marissa-mayer-ceo-yahoo.html)

Rangkuman

1. Bila kita membuat aplikasi web yang memungkinkan pemakai memasukan data dai mana saja, adakalanya kita juga menghendaki agar pemakai bisa mengirim berkas/file yang berada pada tempat ke server
2. Ukuran file yang dapat di upload bergantung pada direktif upload_max_files yang terdapat pada file PHP.INI. Tentu saja kita bias secara langsung melakukan perubahan pada file tersebut
3. Sebagaimana telah diterangkan, file yang diupload akan ditaruh pada direktori yang bersifat sementara. Supaya file ini tidak hilang, Anda perlu menyalinnya pada lokasi Anda sendiri

Daftar Istilah

1. DHTML *[Dynamic HTML]*
2. DNS *[Domain Name System* (Layanan)]
3. Nama Domain
4. Frames
5. FTP *[File Transfer Protocol]*
6. Tautan *[Hyperlink]*
7. Meta Tag
8. RSS *[Really Simple Syndication]*
9. SSI *[Server-Side Includes]*
10. *Script*
11. ASP.NET
12. *Bandwidth*
13. *JavaScript*
14. *Scripting* Bahasa AJAX *[Asynchronous JavaScript dan XML]*
15. ASP *[Active Server Pages]*
16. URI *[Uniform Resource Identifier]*
17. URL *[Uniform Resource Locator]*
18. W3C *[World Wide Web Consortium]*
19. XHTML *[Extensible HyperText Markup Language]*
20. XML *[Extensible Markup Language]*

Latihan Soal

1. Jelaskan kegunaan Upload!
2. Jelaskan perintah untuk membatasi ukuran file!
3. Dapatkah anda menjelaskan, hambatan yang mungkin dapat muncul ketika proses upload sedang dilakukan ?
4. Dalam sebuah proses kerja, mengapa upload menjadi sesuatu yang penting ? berikan contoh dan ilustrasi pada jawaban anda!!
5. Dalam proses pengamanan konten, mengapa kadang kita perlu mengamankan konten yang di upload pada lokasi kita sendiri ??

KONSEP JARINGAN USAHA: SEBUAH STUDI KASUS

Beberapa perusahaan yang sukses membangun jaringan usaha:

Masalah

1. Asia Pacific Design Group. Jaringan usaha ini dibentuk oleh 11 perusahaan yang bergerak dalam jasa konsultasi untuk seluruh aspek bangunan dan konstruksi
2. ATVC. ATVC adalah Automotive Trim and Upholstrey Contractors Network. Dimana 8 perusahaan membentuk jaringan usaha untuk memenangkan kontrak-kontrak tender pemasangan rel untuk Trim, dimana yang selama ini selalu dimenangkan oleh Bridgestone dari Jepang atau Amerika Utara.
3. Daplar. Daplar merupakan jaringan kerjasama dari 4 perusahaan pembuat kabinet dan penyambungan yang memproduksi kitchen set dan kamar mandi. Jaringan usaha ini mampu memproduksi rangka-rangka rumah yang "knocked down".
4. Ambulances to Asia. Jaringan ini menyediakan paket ambulances siap pakai untuk layanan medis guna memenuhi permintaan di pasaran Asia.
5. Oz Electronics Manufacturing. Jaringan tersebut adalah kerjasama usaha antara 3 (tiga) perusahaan

elektronik kecil dimana menghadapi masalah yang sama yaitu biaya komponen perusahaan tersebut yang tinggi yang dibutuhkan. Tanpa jaringan ini tampaknya mereka tidak mungkin bisa bertahan dalam menghadapi biaya tinggi tersebut. Selain konsepsi jaringan usaha tersebut diatas, menurut pendapat C. Richard Hatch (2000) dalam makalah yang disampaikan pada lokakarya di Manila yaitu "The ADB/OECD Workshop on SME Financing in Asia" mengemukakan beberapa konsep jaringan usaha, yang secara umum terdapat 4 type jaringan yang berbeda, antara lain.

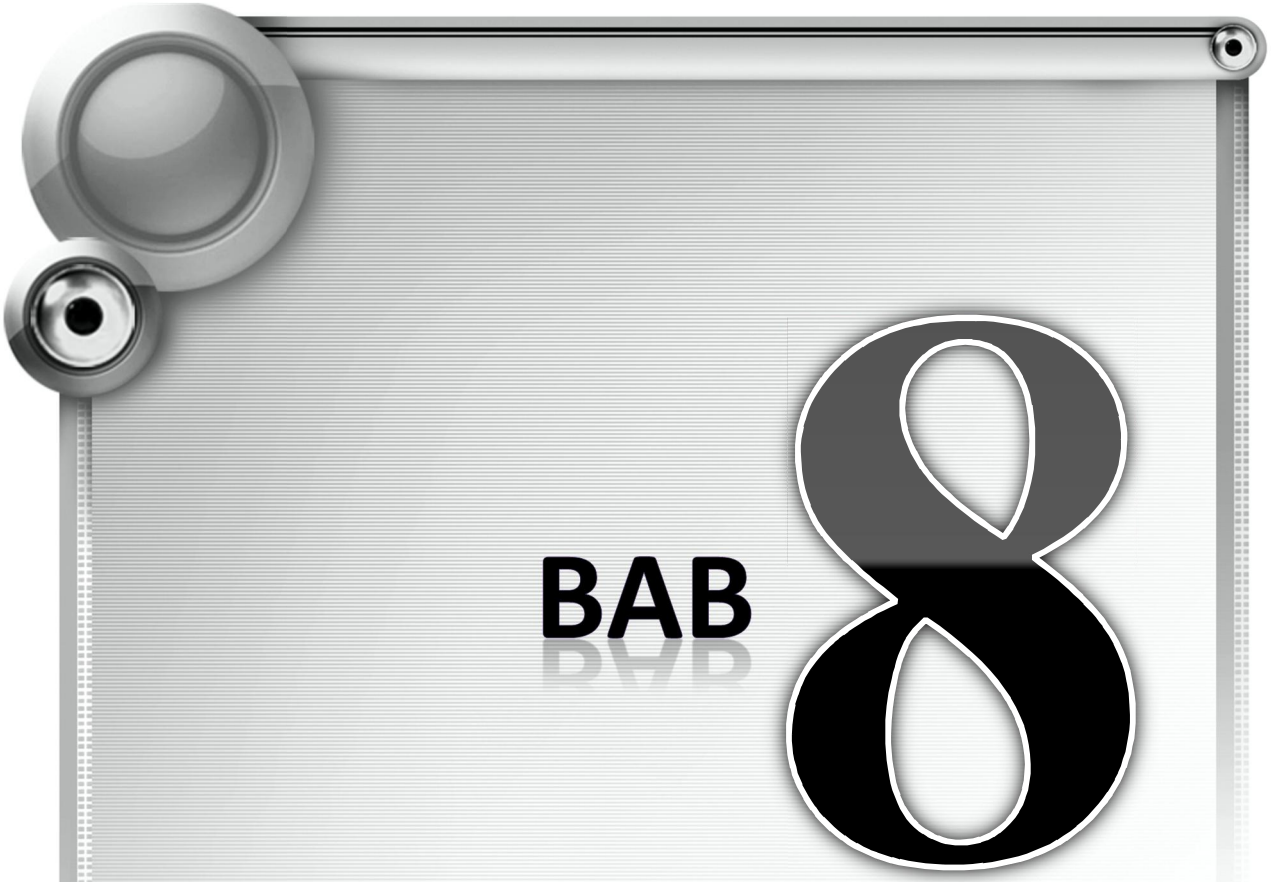
Jaringan yang berkonfigurasikan kembali agar rantai perusahaan bermakna, untuk membuat *scope* bisnis yang ekonomis. Beberapa contoh jaringan usaha ini adalah di Denmark, 11 pembuat pakaian jadi membentuk Christian Dior Line (CD-Line) untuk menggali kemampuan mereka dalam memenuhi perlengkapan untuk memproduksi jas, baju, assesoris dan retsleting. Sehingga perusahaan-perusahaan yang membentuk jaringan tersebut mampu membangun "image clothing" bagi perusahaan besar Eropa.

Jaringan yang meningkatkan efisiensi internal, memampukan UMKM akan menyadari pentingnya skala ekonomi. Salah satu contoh yang berhasil mengenai hal ini adalah proyek perusahaan penghasil alat-alat pertanian di bagian Utara Argentina dekat Propinsi Mato Grasso di Brasilia. Jaringan tersebut dibentuk oleh karena adanya kondisi lahan yang mampu, mereka ingin mengembangkan pertanian yang efektif.

Mereka mengembangkan konsep pemasaran bersama dan percobaan rekayasa desain untuk mengurangi integrasi vertical agar proses produksi lebih efisien. Jaringan ini secara perlahan berhasil menjadi spesialisasi manufaktur yang besar, meningkatkan sub-kontrak dan harga yang kompetitif. Jaringan untuk mengembangkan pengelolaan yang ekonomis.

Secara teori ekonomi, usaha besar akan memperoleh keuntungan dengan pengelolaan yang ekonomis: adanya kemampuan menyebarkan biaya tinggi dari administrasi yang canggih, pembiayaan, dan aktivitas pemasaran melalui volume transaksi bisnis yang besar. Sementara itu produksi barang dan jasa yang didesentralisasi dapat menjadi efisien, manajer sekaligus pemilik yang terisolasi akan sama efektifnya dengan team pengelolaan yang tersebar.

(sumber : http://microentrepreneurshipcommunity.blogspot.com/2012/04/membangun-jaringan-usaha-bagi-usaha.html)



BAB 8

Pemahaman Tentang Cookies, Session, User

Tujuan Instruksional Umum

Setelah mempelajari bab ini, Mahasiswa dipastikan dapat:

1. Menjelaskan Kegunaan dari *Cookies*
2. Menjelaskan Kegunaan dari *Session*
3. Menjelaskan Aplikasi *Manjemen User*

BAB

8

PEMAHAMAN TENTANG *COOKIES, SESSION, USER*

8.1. Menangani *Cookies* dan *Session*

8.1.1. Penggunaan *Cookie*

Namun yang harus diperhatikan bahwa nilai TRUE atau FALSE dari fungsi ini tidak mencerminkan apakah cookie telah berhasil diterima oleh user atau tidak.

Cookie adalah mekanisme penyimpanan data pada browser *(client)* dimana data tersebut mengandung suatu informasi.

Kemampuan PHP dalam mengakses cookie disediakan dalam suatu fungsi bernama setcookie(). Karena cookie merupakan bagian dari HTTP header, maka fungsi setcookie() harus dijalankan sebelum ada output apapun pada browser. Jika ternyata telah ada output yang dikeluarkan sebelum fungsi setcookie(), maka fungsi ini akan menghasilkan nilai FALSE dan cookie tidak akan terkirim. Namun yang harus diperhatikan bahwa nilai TRUE atau FALSE dari fungsi ini tidak mencerminkan apakah cookie telah berhasil diterima oleh user atau tidak.

OK, sekarang kita bahas lebih dalam fungsi setcookie(). Sintaks penggunaan fungsi setcookie() adalah sebagai berikut:

```
setcookie
(nama[,nilai[,kadaluwarsa[,path[,domain[,secure]]]]]);
```

Arti dari argumen-argumen tersebut adalah sebagai berikut:

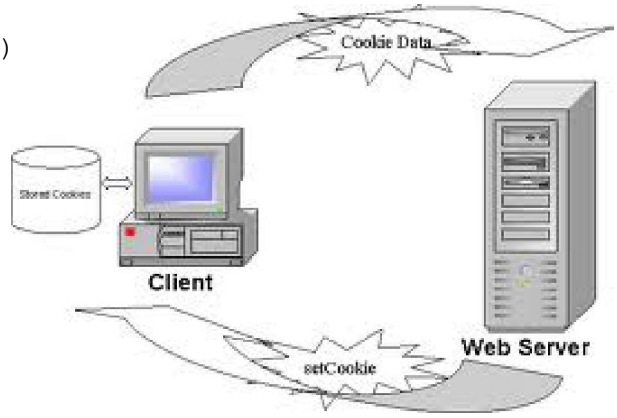
- Nama: merupakan nama dari cookie. Jika nama cookie tersebut telah ada di browser, maka cookie lama akan dihapus dan diganti dengan cookie yang baru.

- Nilai: merupakan informasi yang akan disimpan dalam cookie.
- Kadaluwarsa: merupakan tanggal kadaluwarsa dari sebuah cookie. Jika melewati tanggal tersebut, maka informasi cookie akan dihapus. Nilai dari argumen ini menggunakan tipe Unix time integer yang dapat dihasilkan dari fungsi time().
- Path dan domain: PHP manual tidak memberikan deskripsi mengenai kedua argumen ini, tetapi diduga kedua argumen ini memberikan informasi mengenai path dan domain dari situs yang dikunjungi oleh user.
- Secure: menentukan apakah cookie akan dikirim melalui protokol HTTPS.

Selain argumen nama, argumen-argumen tersebut bersifat opsional. Jika harus dituliskan tetapi tidak mengandung isi apapun, argumen nilai, path, dan domain dapat dituliskan dengan string kosong ("") sedangkan argumen kadaluwarsa dan secure karena berupa integer dapat dituliskan dengan angka 0. Berikut adalah contoh sederhana dari penggunaan cookie:

Seperti halnya *session, cookies* juga merupakan sebuah konsep penyimpanan informasi *user.* Hanya saja, jika session tempat penyimpanan berada di server sedangkan *cookies* berada di client.

```
<?php
if (!isset($_COOKIE['test'])) {
      setcookie("test","Isi test cookie")
      echo "<HTML>";
      echo "<HEAD>";
      echo "<TITLE> Welcome </TITLE>";
      echo "</HEAD>";
      echo "<BODY>";
      echo "This is your first visit";
      echo "</BODY>";
      echo "</HTML>";
} else {
      echo "<HTML>";
      echo "<HEAD>";
      echo "<TITLE> Welcome </TITLE>";
      echo "</HEAD>";
      echo "<BODY>";
      echo "You have visited here before <BR>";
      echo "This is your cookie's value: <BR>";
      echo $_COOKIE['test'];
      echo "</BODY>";
      echo "</HTML>";
}
?>
```



Gambar Cara Kerja Cookies
(sumber: http://www.transiskom.com/)

Simpanlah skrip ini dengan nama cookie.php. Jika file cookie.php tersebut kita jalankan pada browser, maka mula-mula PHP akan memeriksa apakah variabel $_COOKIE['test'] telah diset. Jika belum, maka server akan mengeksekusi skrip pada blok yang pertama (sebelum else). Dengan demikian pada pemanggilan file cookie.php yang pertama kali akan muncul tulisan *"This is your first visit"* (bandingkan dengan gambar 1). Sebelum itu PHP akan mengirim cookie bernama test yang isinya adalah *"Isi test cookie"*.

Oleh karena itu, konsep cookies sebaiknya jangan digunakan untuk menyimpan informasi login user seperti username dan password dan sebagainya.

Lalu pada kunjungan yang kedua dan seterusnya, cookie telah tersimpan pada browser dan otomatis variabel $_COOKIE['test'] telah diset sehingga dengan demikian yang dieksekusi oleh server adalah blok skrip setelah else. Jadi pada kunjungan yang kedua dan seterusnya, browser akan memberikan output seperti gambar 2.

Jika Anda menginginkan cookie tersebut memiliki kadaluwarsa, maka contoh penggunaannya adalah sebagai berikut:

```
setcookie("test","Isi test cookie", time() + 3600);
```

Cookie tersebut akan memiliki kadaluwarsa 1 jam setelah diset. Ingat, 1 jam adalah sama dengan 3600 detik.

Cookie yang telah diset juga dapat dihapus. Penghapusan cookie juga menggunakan fungsi setcookie(). Caranya adalah dengan mengatur waktu kadaluwarsa cookie dengan suatu nilai di masa lampau. Contoh:

```
setcookie("test","Isi test cookie", time() - 3600);
```

Penggunaan cookie ini memiliki kelemahan yaitu apabila user melakukan setting pada browsernya agar tidak dapat menerima cookie. Jika hal ini terjadi, maka eksekusi terhadap file cookie.php tidak akan pernah mencapai keadaan seperti pada gambar 2, melainkan akan terus seperti gambar 1.

Karena hal ini, maka apabila Anda akan mengembangkan aplikasi web menggunakan cookie, maka sebaiknya di halaman pertama dari aplikasi tersebut diberi suatu petunjuk agar user mengaktifkan penerimaan cookie pada browsernya.

Selain *user* bisa melihat informasi yang disimpan, user juga bisa bisa men-*disable* atau menon-aktifkan *cookies* itu sendiri. Jika *cookies didisable* maka program yang memanfaatkan *cook-ies* tentunya tidak akan berjalan dengan baik. *Cookies* sendiri biasanya dipakai dalam aplikasi *shooping cart*. Biasa digunakan untuk menyimpan sementara produk-produk yang dipilih oleh pengunjung pada saat berbelanja.

8.1.2. Penanganan Session

Session memiliki fungsi yang mirip dengan cookies, yaitu suatu teknik penyimpanan data. Bedanya data akan disimpan di sisi server. Session berguna untuk pembuatan password atau authentication, e-commerce, sistem informasi dan sebagainya. Untuk memulai session:

```
session_start();
```

Untuk mengisi variabel global session:

```
$_SESSION["Nama_var"] = "isi session"; // cara 1
$HTTP_SESSION_VARS["var1"] = "isi session"; // cara
2
```

Untuk mengakhiri session :

```
session_destroy();
```

Variabel session dapat diisi dengan data-data seperti data authentikasi dan sebagainya. Perlu diketahui variabel global session berbentuk array sehingga anda dapat membuat variabel session lebih dari satu dan untuk mengakses tiap array dapat menggunakan indeksnya yaitu *"Nama_var"*. Contoh penggunaan :

```
<?
session_start();
?>

<HTML>
<HEAD>
<TITLE> Proses </TITLE>
</HEAD>

<BODY>
<?
$_SESSION['nama'] = $_POST['nama'];
$_SESSION['umur'] = $_POST['umur'];
$_SESSION['email'] = $_POST['email'];
$_SESSION['waktu'] = date("h:i");
?>

<H1> Hallo
<?=$_SESSION['nama']?>
</H1>
<H2> Selamat Datang Di Situs Kami </H2>
Umur Anda saat ini adalah
<?=$_SESSION['umur']?>
tahun <BR>
Alamat email Anda adalah
<?=$_SESSION['email']?> <BR>
Anda login pada jam
<?=$_SESSION['waktu']?>
<BR>
<A HREF="next.php">Klik di sini</A>
untuk menuju ke halaman berikut.
</BODY>
</HTML>
```

Session adalah salah satu fasilitas yang digunakan untuk menyimpan data sementara ke dalam variabel (variabel session) sehinga data tadi dapat di akses oleh client selama variabel session tadi tidak di kosongkan atau dihilangkan. Nilai variabel di dalam session di simpan di sisi server (web server).

Setelah itu buatlah sebuah file lagi bernama next.php yang isinya sebagai berikut:

```
<?
session_start();
?>
<HTML>
<HEAD>
<TITLE> Halaman Dua </TITLE>
</HEAD>
<BODY>
```

Jadi session relatif lebih aman digunakan untuk menyimpan variabel nilai yang bersifat rahasia seperti username dan password pada saat login. Kegunaan session diantaranya untuk menyimpan informasi login yang berlaku hanya dalam satu sesi dan menyimpan catatan order barang dalam sistem e-commerce/ transaksi online

```
<H1> Hallo
<?=$_SESSION['nama']?>
</H1>
<H2> Anda memasuki halaman kedua </H2>
Berikut adalah informasi yang Anda
bawa dari halaman pertama: <BR>
<HR>
Umur Anda saat ini adalah
<?=$_SESSION['umur']?>
tahun <BR>
Alamat email Anda adalah
<?=$_SESSION['email']?>
<BR>
Anda login pada jam
<?=$_SESSION['waktu']?>
<HR>
<?
$_SESSION=array();
session_destroy();
?>
</BODY>
</HTML>
```

Jalankan file javaform.html, lalu isikan data-data pada form tersebut. Setelah Anda mengklik tombol submit, maka browser akan menuju ke file proses.php. Pada file proses.php akan terdapat sebuah link untuk menuju ke halaman next.php. Jika link ini Anda klik, maka browser akan menuju ke halaman next.php dan semua nilai variabel yang didaftarkan ke dalam session, yaitu variabel $_SESSION['nama'], $_SESSION['umur'], $_SESSION['email'], dan $_SESSION['waktu'] akan dipertahankan pada halaman next.php. Jika variabel pada file proses.php tidak dimasukkan ke dalam session, maka nilainya tidak akan muncul pada halaman next.php. Perhatikan bahwa fungsi session_start() harus dituliskan pada bagian skrip paling awal sebelum ada output apapun yang diberikan kepada browser.

8.2. Enkripsi Data

Enkripsi merupakan proses pengkodean data sehingga maknanya menjadi jelas. Proses ini sering dilakukan untuk meningkatkan keamanan data yang dikirim. Kebalikan dari enkripi adalah dekripsi, yaitu membalikan pesan terenkripsi kembali ke bentuk normal. Berkaitan dengan hal ini, PHP menyediakan beberapa algoritma untuk melakukan enkripsi data, antara lain dengan menggunakan algoritma Message-Digest, DES serta SHA1. adapun algoritma – algoritma itu sudah dikemas dalam bentuk fungsi, sehingga tingal kita gunakan saja.

Ketika kita mengulas tentang enkripsi data, maka tidak akan terlepas dari beberapa istilah berikut:

- Plaintext, merupakan bentuk awal dari pesan
- Chipertext, merupakan bentuk terenkripsi dari data
- Kriptografi, yaitu cara penggunaan enkripsi untuk menyembunyikan teks.
- Kriptoanalis, yaitu orang yang mempelajari enkripsi dan pesan terenkripsi dengan tujuan menemukan makna yang tersembunyi
- Kriptografer, memiliki tujuan sama seperti kriptoanalis, bedanya adalah kriptografer bekerja untuk tujuan legal sedang kriptoanalis cenderung secara illegal.

MD5 (Message-Digest algortihm 5) ialah fungsi hash kriptografik yang digunakan secara luas dengan hash value 128-bit. Melakukan enkripsi terhadap password sangat diperlukan untuk keamanan dari tangan-tangan jahil para hacker. Meski sekarang sudah dapat di-decrypt

(Sumber: md5decryption.com)

8.2.1. Enkripsi dengan Message – Digest

Fungsi MD5 merupakan mekanisme pengacakan string menggunakan algoritma Message – Digest, dan akan mengembalikan hash atau chipertext. Secara normal, hash yang dihasilkan adalah bilangan heksadesimal 32 karakter. Sintaks md5() selengkapnya sebagai berikut :

```
string md5 (string str [,bool raw_output])
contoh :
<?php
/* md5.php */

$str = 'rahasia';

echo "<p> Hasil kalkulasi tanpa parameter opsional <br>";
echo md5($str);
// Hasil: ac43724f16e9241d990427ab7c8f4228

echo "<p> Hasil kalkulasi dengan parameter opsional <br>";
echo md5($str, TRUE);
// Hasil: ¬CrOé$™'«|□B(

?>
```

Selain melakukan enkripsi string, tersedia fungsi khusus ketika Anda ingin melakukan enkripsi file, yaitu md5_file()

```
$myfile = "myfile.txt";
echo "Nama File $myfile";

md5_myfile = md5_file($myfile);
echo "<p>Enkripsi Nama file : $md5_myfile";
```

8.2.2. Enkripsi dengan Crypt.

Fungsi crypt merupakan salah satu alternative lain yang digunakan untuk enkripsi data. Fungsi ini akan mengembalikan data terenkripsi dengan berbasis pada algoritma DES (Data Enkription Standard). Sintaks crypt selengkapnya sebagai berikut:

```
String crypt (string str [,string salt])
```

Contoh :

```
<?php
/* crypt.php */

$str = "rahasia";

echo "<p> Hasil kalkulasi tanpa parameter opsional
<br>";
echo crypt($str);
// Hasil: data enkripsi random

echo "<p> Hasil kalkulasi dengan parameter opsional
<br>";
echo crypt($str, "RAHASIA");
// Hasil: RA3V2kL6SzqsA

?>
```

Enkripsi dengan Algoritma SHA1

Jika Anda kurang tertarik dengan dua algoritma enkripsi di atas, masih ada alternatif lain yang ditawarkan PHP bagi Anda untuk melakukan enkripsi, yaitu menggunakan sha1(). Fungsi ini dikembangkan dengan berbasis pada US Secure Hash Algorithm 1 (SGA1). Sintaknya adalah sebagai berikut:

```
String sha1 ( string str [,bool raw_output])
```

Enkripsi dengan Algoritma SHA1: Fungsi ini memberikan hasil hash dalam bentuk heksadesimal dengan panjang karakter sebanyak 40. Dan akan mengembalikan hasil dengan panjang 20 karakter dalam bentuk binary apabila diberi nilai true pada parameternya.

Contoh:

```
<?php
/* sha1.php */

$str = 'rahasia';

echo "<p> Hasil kalkulasi tanpa parameter opsional
<br>";
echo sha1($str);
// Hasil: 829b36babd21be519fa5f9353daf5dbdb796993e

echo "<p> Hasil kalkulasi dengan parameter opsional
<br>";
echo sha1($str, TRUE);
// Hasil: ,›6°½!¾QŸ¥ù5=¯]½·–™>

?>
```

8.3. Aplikasi Manajemen User

Agar dapat mengelola data user secara mudah, kita menggunakan table sebagai penyimpan data. Adapun kita memerlukan dua buah table untuk menyimpan data user dan menyimpan log user.

```
CREATE TABLE IF NOT EXISTS user (
  nama varchar(20) NOT NULL DEFAULT '' ,
  password varchar(32) NOT NULL DEFAULT '' ,
  email varchar(25) NOT NULL DEFAULT '' ,
  level varchar(5) NOT NULL DEFAULT '' ,
  PRIMARY KEY (nama),
  UNIQUE KEY Idx_email (email)
);

CREATE TABLE IF NOT EXISTS useractive (
  user varchar(20) NOT NULL DEFAULT '' ,
  level varchar(12) NOT NULL DEFAULT ''
);
```

Dari sruktur table user, yang perlu diperhatikan adalah filed password. Mengingat data password akan kita enkripsi menggunakan md5(). Oleh karena itu panjang field yang digunakan adalah 32 karakter. Jika Anda memberikan nilai kurang dari 32, maka tidak semua chipertext akan tersimpan, dan akhirnya pencocokan data tidak akan pernah sama.

Untuk memudahkan ketika mencoba aplikasi, tidak ada salahnya jika masukan data user awal dengan nama admin seperti berikut:

```
INSERT INTO user
VALUES("admin","md5('admin',"admin@indosql.net","admin");
```

Selanjutnya Anda akan melihat password admin yang telah terenkripsi, sebenarnya perintah diatas sama juga ketika Anda memberikan perintah berikut:

```
INSERT INTO user
VALUES("admin","21232f297a57a5a743894a0e4a801fc3","admin@indosql.net","admin");
```

Pertama siapkan file khusus unruk melakukan koneksi.

```
<?php
/* config.inc */

$host = "localhost";  //Nama host
$user = "root";       //Nama user database
$pass = "";           //Password user database
$db   = "login";      //Nama Database

$conn = @mysql_connect($host, $user, $pass);
@mysql_select_db($db);
```

```
if (!$conn) {
    printf("<title>Ooops,.... Connect Failed</
title>");
    exit();
}
?>
```

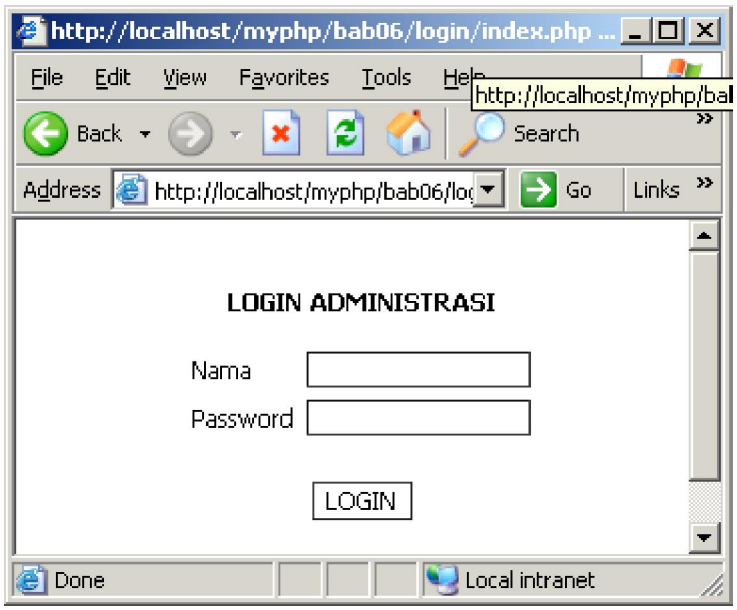
Sebelum melangkah ke pembuatan aplkasi lebih lanjut, perlu Anda ketahui bahwa sebisa mungkin kita melakukan enkripsi terhadap nilai variable yang akan dikirim. Adapun untuk memudahkan dalam mencocokan chipertext, kita membuat fungsi tersendiri bernama cek_curr().

```
<?
/* Cek_Curr.php */

function Cek_Curr($enk) {
    $res = mysql_query("SELECT nama FROM user");
    $i = 0;
    while ($row = mysql_fetch_array($res)) {
       $nama[$i]  = $row[0];
       $i++;
    }

    for ($j = 0; $j <= $i; $j++) {
       $user = md5($nama[$j]);
       if ($user == $enk) {
   $id=$j;
       }
    }
    $user = $nama[$id];
return $user;
}

function Cek_Online() {
    $sql = "SELECT level, count(user) as jml
            FROM useractive GROUP BY level";
    $res = mysql_query($sql);
    $cek = mysql_num_rows($res);
    if ($cek != 0) {
        $str = "Ada ";
        while ($row = mysql_fetch_array($res)) {
           $user = $row[0];
           $jmlh = $row[1];
           $str.="<b>[$jmlh $user]</b> ";
        }
    $str.="sedang Online";
    }
    echo "$str";
}
?>
```

Sekarang kita menyediakan halaman index yang akan digunakan untuk login user sekaligus memeriksa jika cookie telah diset.

```
<?php
/* index.php */

if(!isset($_COOKIE[“my_cookie”])){

   echo '<html><head>'
        .'<link rel=stylesheet type=text/css href=mycss.css>'
        .'<body><center><br><b>LOGIN ADMINISTRASI</b><br><br>'
        .'<table border=0>'
        .'<form action=”redirect.php” method=”post”>'
        .'<tr><td>Nama </td><td>'
        .'<input type=text name=”username” size=18></td></tr>'
        .'<tr><td>Password </td><td>'
        .'<input type=”password” name=”password” size=18></td></tr>'
        .'<tr><td> </td></tr>'
        .'<tr><td align=”center” colspan=2>'
        .'<input type=”submit” value=”LOGIN”></td></tr>'
        .'</form>'
        .'</table>'
        .'</center><br><br><br>';

} else {
  echo “<b> Anda Sudah Login..</b><br>
       [<a href=\”main_menu.php\”>Lanjutkan</a>] 
       [<a href=\”logout.php\”>Log Out</a>]”;
}
?>
```

Tampilan halaman index. Php



Gambar 8.1 Tampilan *logim administrasi*

File program utama yang digunakan untuk authentikasi user adalah redirect.php, dan harus Anda letakan pada bagian paling atas di setiap halaman web yang sifatnya private

```
<?php
/* redirect.php */

header("Last-Modified: " . gmdate("D, d M Y H:i:s")
. " GMT");
header("Cache-Control: no-store, no-cache, must-
revalidate");
header("Cache-Control: post-check=0, pre-check=0",
false);

global $username, $password;

$login_page = "index.php";
$page_ok = "main_menu.php";

$username = $_POST["username"];
$password = $_POST["password"];

if($username == "" && $password == ""  &&
   !isset($_COOKIE["my_cookie"])) {
   include($login_page);
   exit();
}
if ($username != "" || $password != ""  &&
    !isset($_COOKIE["my_cookie"])) {
    if (preg_match ("/[^a-zA-Z0-9]/",
$username.$password)) {
include($login_page);
exit();
    }
}

if (!isset($_COOKIE["my_cookie"]) ) {
    require_once "config.inc";
    $sql = "SELECT nama, password, level FROM user";
    $res = mysql_query($sql);

    $user_exists = false;
    $user = $_POST["username"];
    $md5pass = md5($_POST["password"]);

    $i = 0;
    while ($row = mysql_fetch_array($res)) {
       $nama[$i]  = $row[0];
       $pass[$i]  = $row[1];
       $level[$i] = $row[2];
       if (($nama[$i] == $user) && ($pass[$i] ==
```

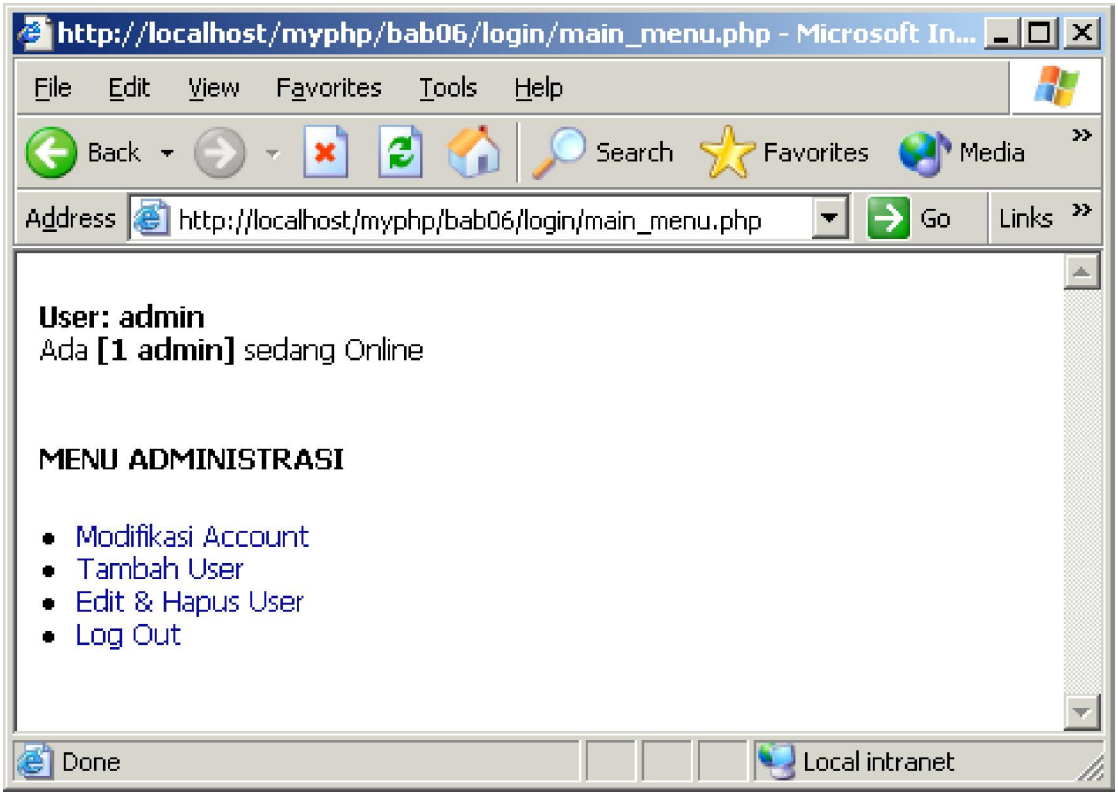
```
$md5pass)) {
         $user_id = $i;
         $user_exists = true;
        }
        $i++;
    }

    if (!$user_exists){
        echo "Nama User atau Password Tidak Cocok";
     include($login_page);
     exit();
    }

    /* enkripsi nilai cookie */
    $val = md5($nama[$user_id]);
    setcookie ("my_cookie", $val, 0);
    setcookie ("level", $level[$user_id], 0);
    mysql_query("INSERT INTO useractive
          VALUES('$nama[$user_id]', '$level[$user_id]')");
    header("Location: $page_ok");
    exit();
}
?>
```

Dalam program redirect.php telah didefinisikan bahwa halaman main_menu.php merupakan halaman yang diperuntukan bagi user setelah login diterima. Berbekal dari nilai variable level, kita membedakan user sevagai admin dan user. Apabila admin yang mengakses, maka semua link akan diaktifkan. Lain halnya ketika level user yang mengakses, dimana link untuk modifikasi user serta penambahanuser menjadi tidak aktif.

```
<?php
/* main_menu.php */
require_once "redirect.php";
require_once "config.inc";
?>

<html>
<link rel=stylesheet type=text/css href=mycss.css>
<body>

<?
if (isset($_COOKIE['level']) &&
   ($_COOKIE['my_cookie'])) {

    require_once "Cek_Curr.php";
    $id = Cek_Curr($_COOKIE[my_cookie]);
    echo "<b>User: $id</b> <br>";
    Cek_Online();
    $u = $_COOKIE["my_cookie"];
    echo "<p><br><b> MENU ADMINISTRASI </b><p>
     <li> <a href='edit_acc.php?u=$u'> Modifikasi Account </a>";
```

```
    if ($_COOKIE[level] == 'admin') {
        echo "<li><a href='in_user.php'> Tambah User
</a>
          <li><a href='edit_user.php'> Edit & Hapus
User </a>";
    } else {
        echo "<font color='#333333'><li> Tambah User
          <li> Edit & Hapus User </font>";
    }
    echo "<li>  <a href='logout.php?u=$u'>Log Out</
a>
          <br><br></body></html>";

} else {
    echo "Selamat.., Anda melihat halaman yang
diproteksi";
}
?>
</body>
</html>
```

Tampilan main_menu.php



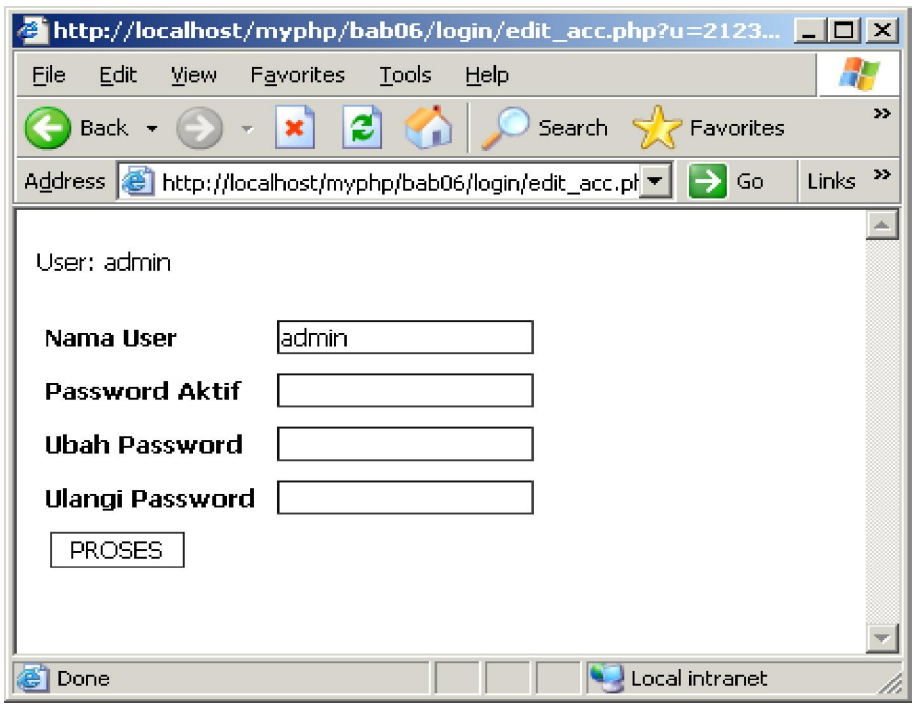
Gambar 8.2 Tampilan *login File*

Halaman main menu yang diakses level admin

```
<?php
/* edit_acc.php */

require_once("redirect.php");
require_once "config.inc";

echo "<html><head>
    <link rel=stylesheet type=text/css href=mycss.css>
    </head><body>";

if (isset($_POST["proses"]) && ($_POST[curr_pass])) {
    $sql = "UPDATE user SET nama = '$_POST[nama]' ";
    if (!empty($_POST[curr_pass])) {
        if (md5($_POST[curr_pass]) != ($_POST[pass])) {
            echo "<b>Password Aktif Tidak Cocok...!!</b>";
            exit();
        } elseif ($_POST[pass1] != $_POST[pass2]) {
            echo "<b>Password Baru Tidak Cocok...!!</b>";
            exit();
        } else {
            $sql.= ",password=md5('$_POST[pass2]') ";
        }
    }
    $id = $_POST["id"];
    $sql.= "WHERE nama = '$id'";

    $res = mysql_query($sql);
    if ($res) {
        $us = md5($id);
        echo "<p><b>Account Anda Sudah dimodifikasi..</b>";
        echo "<br><a href='logout.php?u=$us'> Log Out</a>";
    }
} else {

require_once "Cek_Curr.php";
$id = Cek_Curr($_GET[u]);

$sql = "SELECT * FROM user where nama='$id'";
$res = mysql_query($sql);
$jml = mysql_num_rows($res);
if ($jml != 0) {
  echo "<p>User: $id <p>";
  echo "<form action=\"$PHP_SELF\" method=\"POST\">";
  echo "<table border=0 cellpadding=4 cellspacing=0>";

  while ($row = mysql_fetch_row($res)) {
     $pass = $row[1];
     $nm = $row[0];
     echo "<tr><td><b>Nama User</b>
```

```
        </td><td><input type='text' name='nama'
                        value='$nm'></td></tr>";
      echo "<tr><td><b>Password Aktif</b>
        </td><td><input type='password'
  name='curr_pass'></td></tr>";
      echo "<tr><td><b>Ubah Password</b>
        </td><td><input type='password' name='pass1'>
        </td></tr>";
      echo "<tr><td><b>Ulangi Password</b>
        </td><td><input type='password' name='pass2'>
        </td></tr>";
  }
  echo "<input type='hidden' name='id'
value='$nm'>";
  echo "<input type='hidden' name='pass'
value='$pass'>";
  echo "<tr><td colspan=2><input type='submit'
        name='proses' value='PROSES'> </td></tr>";
  echo "</table>";
  echo "</form>";
}
echo "</body></html>";

}

?>
```

Tampilan edit account oleh level admin (yang bisa menjalankan file ini adalah level admin)



Gambar 8.3 Tampilan *user admin*

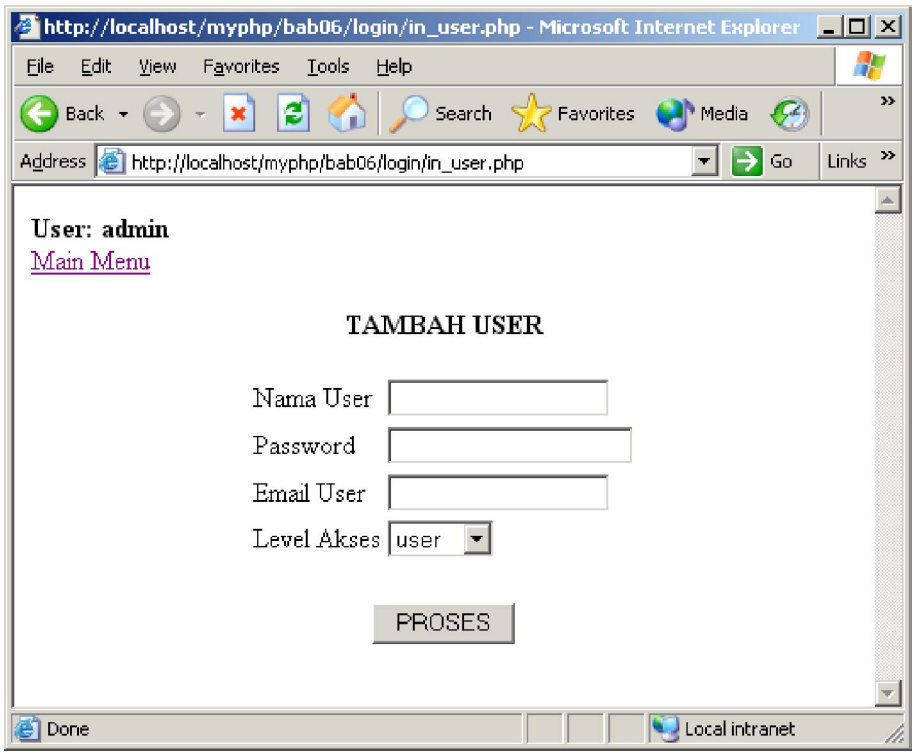
Seperti telah disinggung sebelumnya, ada perbedaan antara *level user* dengan level admin dalam mengakses halaman menu, mungkin apalah artinya menyembunyikan link jika tidak ditindaklanjuti. Sebab user yang cerdik mungkin akan mencoba dengan langsung mengetik nama file program yang tidak diperbolehkan diaksesnya.

Untuk itu kita perlu memproteksi dan memastikan bahwa user memiliki level admin. Karena sudah diset di cookie maka kita tinggal memriksa apakah nilai variable level benar – benar admin.

```
<?php
/* in_user.php */

require_once "redirect.php";
require_once "Cek_Curr.php";
require_once "config.inc";

$id = Cek_Curr($_COOKIE[my_cookie]);
echo "<b>User: $id</b> <br>";
echo "<a href='main_menu.php'> Main Menu </a>";

if (isset($_COOKIE['level']) && ($_COOKIE['my_cookie']) ) {

    if ($_COOKIE['level'] == "admin") {
    echo "<center><br><b>TAMBAH USER</b><br><br>
       <table border=0>
       <form action='$PHP_SELF' method='post'>
       <tr><td> Nama User </td>
       <td><input type=text name='nama' size=18 ></td></tr>
       <tr><td> Password </td>
       <td><input type='password' name='password'></td></tr>
       <tr><td> Email User </td>
       <td><input type='text' name='email' size=18></td></tr>
       <tr><td> Level Akses </td>
       <td><select name='level'>
             <option value='admin'> admin
             <option value='user' selected> user
       </select></td></tr>
       <tr><td> </td></tr>
       <tr><td align='center' colspan=2>
       <input type='submit' name='add' value='PROSES'>
       </td></tr>
       </form>
       </table>
     </center><p>";
    } else {
       echo "<p> <b>Anda Tidak Berhak Menambah User</b>";
    }

    if (isset($_POST["add"]) && ($_POST[nama]) &&
       ($_POST[password]) ) {
```

```
        $sql = "INSERT INTO user VALUES
               ('$_POST[nama]',
md5('$_POST[password]'),
                '$_POST[email]', '$_POST[level]') ";
        $res = mysql_query($sql);

        if ($res) {
    echo "<b> Data Penambahan User Baru </b>";
            echo "<p> Nama: " .$_POST[nama]. "<br>";
            echo " Password: " .$_POST[password].
"<br>";
            echo " Email: " .$_POST[email]. "<br>";
            echo " Level: " .$_POST[level]. "<br>";
        } else {
            echo "Gagal Menambah User...!!<br>";
        }
    }
}
?>
```

Tampilan in_user oleh user admin



Gambar 8.4 Tampilan *form tambah user*

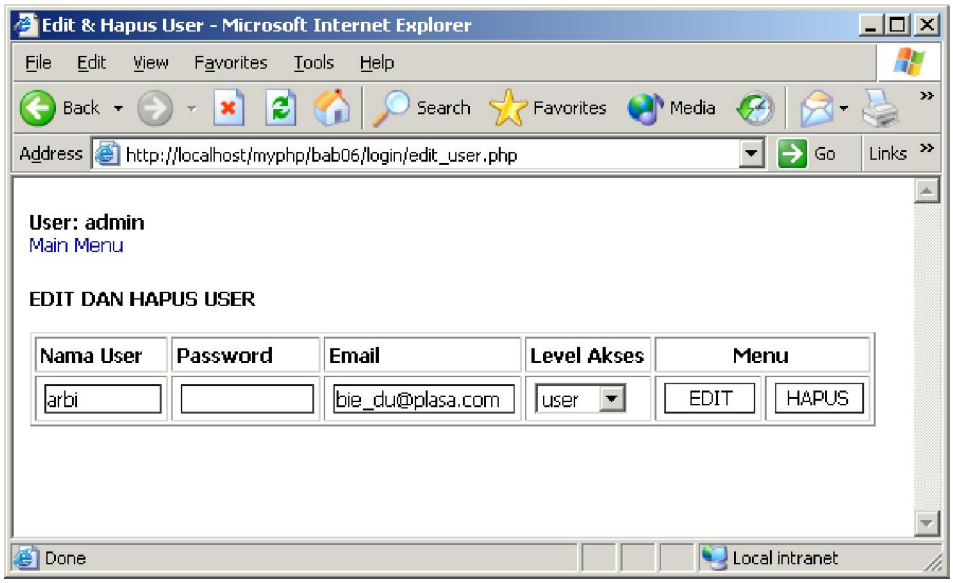
Sama halnya ketika membuat halaman edit data, pastikan bahwa user memiliki level admin. Pada intinya halaman ini kita gunakan untuk melakukan modifikasi seluruh user, dan hanya admin yang berhak mengaksesnya.

```
<?php
/* edit_user.php */
```

```
require_once "redirect.php";
require_once "Cek_Curr.php";
require_once "config.inc";

$curr_id = Cek_Curr($_COOKIE[my_cookie]);
echo "<b>User: $curr_id</b> <br>";
echo "<a href='main_menu.php'> Main Menu </a>";

if (isset($_COOKIE['level']) && ($_COOKIE['my_cookie']) ) {

   if ($_COOKIE['level'] == "admin") {
       $sql = "SELECT * FROM user WHERE level != 'admin'";
       $res = mysql_query($sql);

       echo "<html><head>"
           ."<title>Edit & Hapus User</title>"
           ."<link rel=stylesheet type=text/css href=mycss.css>"
           ."</head><body>";

       echo "<p><b>EDIT DAN HAPUS USER</b><br><br>";

       echo "<table border=1 cellpadding=3>";
       echo "<td><b> Nama User </b></td>
          <td><b> Password </b></td>
          <td><b> Email </b></td>
          <td><b> Level Akses </b></td>";
       echo "<td colspan=2 align=center><b> Menu </b>
             </td></tr>";

       $i=0;
       while ($row = mysql_fetch_row($res)) {
           $id = $row[0];
        echo "<form action=\"$PHP_SELF\" method='POST'>";
        echo"<input type='hidden' name=\"id\" value=\"$id\">";
        echo "<tr><td> <input type=text name='nama'
                              value='$id' size=10> </td>";
        echo "<td> <input type='password'
                          name='password' size=12> </td>";
        echo "<td> <input type=text name='email'
                          value='$row[2]' size=18> </td>";
           echo "<td><select name='lev'>
                 <option value='user'> user
                 <option value='admin'> admin
                 </select></td>";
        echo "<td> <input type='submit' name='edit'
                          value='  EDIT '> </td></form>";
```

```
                // Hapus user aktif
    echo "<form action='$PHP_SELF' method='POST'>";
    echo"<input type='hidden' name=\"id\"
value=\"$id\">";
    echo "<td> <input type='submit' name='hapus'
                        value='HAPUS'> </td></form>";
    echo "</tr>";
    $i++;
        }
        echo "</table>";

    } else {
         echo "<p> <b>Anda Tidak Berhak Memodifikasi
User</b>";
    }

    if ($HTTP_POST_VARS["edit"]) {
        $id = $_POST["id"];
        $sql_ed = "UPDATE user SET
nama='$_POST[nama]', ";
        if (!empty($_POST[password])) {
            $sql_ed.=
"password=md5('$_POST[password]'), ";
        }
        $sql_ed.= "email='$_POST[email]',
                    level='$_POST[lev]' WHERE
nama='$id'";

        $edit = mysql_query($sql_ed);
        if (!$edit) {
            echo "<p><b>Gagal Modifikasi User
<u>$id</u> </b>";
        }

    } elseif ($HTTP_POST_VARS["hapus"]) {
        $id = $_POST["id"];
        $sql_del = "DELETE FROM user WHERE
nama='$id'";

        $del = mysql_query($sql_del);
        if (!$del) {
            echo "<p><b>Gagal Menghapus User <u>$id</
u></b>";
        }
    }
}
?>
```

Tampilan edit user :



Gambar 8.5 Tampilan *form edit dan hapus user*

Akhirnya kita sampai pada program penutup, yaitu halaman untuk memproses *user logout.*

```
<?php
/* logout.php */

require_once "config.inc";

/* Menghapus cookies aktif */
setcookie ("my_cookie", "", 0, "", "", 0);
setcookie ("level", "", 0, "", "", 0);

require_once "Cek_Curr.php";
$id = Cek_Curr($_GET[u]);

/* Hapus log di tabel useractive */
mysql_query("DELETE FROM useractive WHERE user='$id'");

/* Menuju ke halaman login */
header("Location: index.php");
exit();
?>
```

Perlu diingat bahwa logout tidak hanya sekadar menghapus *cookie,* tapi juga menghapus log dalam *table useractive.* Jadi ketika user *logout* maka data keaktifan *user* juga akan terhapus.

ANDREW E. RUBIN | Penemu Android

Rubin di Google Developer Day, Jepang, 2008

Nama Lengkap:
Andrew E. Rubin

Tempat dan Tanggal Lahir:
New Bedford, Amerika Serikat pada 22 Juni 1946

Kebangsaan:
Amerika Serikat

Almamater:
Horace Greeley High School di Chappaqua, New York (1977 - 1981).
Utica College, Utica, New York, Bachelor of Science dalam ilmu komputer (1981 - 1986)

Pekerjaan:
Android Inc., pendiri bersama, 2003 - 2005.

Google, 2005–sekarang. Wakil Presiden Senior Android

Hasil Penemuan:
Penemu sistem surat elektronik

(Sumber: https://id.wikipedia.org/wiki/Ray_Tomlinson)

Andy Rubin lahir pada tanggal 22 Juni 1946 di New Bedford, Amerika Serikat. Dia adalah pengembang dari OS Android. Sejak kecil, Rubin telah terbiasa melihat banyak gadget baru. Ini karena ayahnya, seorang psikolog yang terjun ke bisnis direct marketing, yang akan menyimpan produk elektronik yang dijual di room Rubin. Dia sangat tertarik dengan segala hal yang berbau robot. Di Carl Zeiss AG, dimana dia pertama kali bekerja setelah lulus kuliah, Rubin ditempatkan di divisi robotika, komunikasi digital antara perangkat jaringan dan pengukuran manufaktur. Berikut dari Carl Zeiss, ia pernah bekerja di bidang robot di sebuah perusahaan di Swiss.

Karir Rubin di bidang robotika nampaknya semakin jernih, namun hidupnya berubah karena liburan di Cayman Island pada tahun 1989. Saat mengunjungi pulau tropis di Jamaika, Rubin secara tidak sengaja bertemu dengan seorang pria bernama Bill Caswell. Pria ini sedang tidur di pantai, didorong dari sebuah pondok setelah berkelahi dengan pacarnya. Andy menawarkan kepadanya tempat berlindung, dan seperti hutangnya, Casswell menawarkan pekerjaan itu.

Sungguh kebetulan bahwa pria itu sedang bekerja di Apple. Di Apple, Rubin sempat bersenang-senang. Saat itu, Apple masih dalam kondisi baik berkat komputer Macintosh. Budaya Apple ditransmisikan ke Rubin sendiri. Di sana ia sempat melakukan kenakalan, seperti memprogram ulang sistem telepon sehingga ia bisa berpura-pura menjadi CEO, John Sculley. Lelucon seperti itu mungkin akan seperti Steve Jobs, pria yang suka bercanda lewat telepon, tapi saat itu adalah masa Apple tanpa Jobs.

Dari manufaktur, Rubin pindah ke riset di Apple. Kemudian, pada tahun 1990, Apple melakukan *spin off* untuk membentuk sebuah perusahaan bernama *General Magic* dan Rubin berpartisipasi di dalamnya. ***General Magic*** berfokus pada pengembangan perangkat genggam dan komunikasi. Insinyur workaholic, termasuk Rubin, tentu saja, telah mengembangkan sebuah perangkat lunak bernama Magic Cap. Sayangnya, Magic Cap tidak diterima dari handset dan perusahaan telekomunikasi. Magic Cap yang menerapkan beberapa hanya melakukannya secara singkat. Magic Cap akhirnya hancur.

Beberapa pengembang di General Magic, bersama beberapa veteran Apple, kemudian mendirikan Artemis Research. Perusahaan sedang mengembangkan sesuatu yang disebut WebTV, sebuah upaya awal untuk menggabungkan Internet dengan televisi. Rubin bergabung dengan Artemis untuk mengembangkan WebTV. Saat Microsoft membeli Artemis, pada 1997, Rubin bergabung dengan perusahaan raksasa tersebut. Rubin episode khasnya kembali seru di Microsoft. Rubin membangun robot yang dilengkapi kamera untuk bekerja pada rekan-rekannya. Hebatnya, robot itu terhubung ke internet dan dalam satu kejadian dibobol oleh pesta di luar Microsoft. Pada tahun 1999, Rubin keluar dari WebTV (dan memang, dia bukan lagi karyawan Microsoft). Dia kemudian menyewa sebuah toko di Palo Alto, California, dan menelepon toko itu sebagai laboratorium.

Di tempat itu diisi dengan berbagai koleksi robot mainan Rubin, lahir dari sebuah ide untuk sebuah produk baru. Bersama beberapa rekannya, Rubin kemudian mendirikan Danger Inc Success yang diraih melalui perangkat bernama Bahaya Sidekick. Awalnya, perangkat itu bernama Danger Hiptop, namun pasarnya dikenal dengan sebutan T-Mobile Sidekick. "*Kami ingin membuat perangkat, seukuran cokelat, dengan harga di bawah $ 10 dan bisa digunakan untuk memindai objek dan mendapatkan informasi tentangnya dari Internet. Kemudian, tambahkan radio dan pemancar, jadilah Sidekick,*" *Rubin mengatakan tentang Sidekick.*

Saat ini, Sidekick sudah terlihat aus, namun pada masanya, Sidekick merupakan objek yang bertentangan dengan konsep teknologi yang melampaui waktu. Perangkat, menurut Rubin, akses data dengan kemampuan ponsel. Saat muncul di pasaran, Sidekick harus menghadapi kenyataan bahwa pasar PDA sedang hilang. Namun, Rubin menegaskan bahwa Sidekick bukan PDA. "*Seharusnya, orang bertanya apakah ini bukan PDA atau ponsel Mereka harus bertanya, apakah ini platform untuk pengembang pihak ketiga? Ini bukan hal baru. Ini adalah pertama kalinya ponsel digunakan sebagai platform untuk pengembang pihak ketiga., "Kata Rubin.*

Nah, apa kata Rubin tadi tidak biasa lagi. Tengok saja Apple dengan jutaan aplikasi pihak ketiga yang hadir di iPhone. Hal lain yang membuat Bahaya, yang pada saat itu belum terpikirkan, adalah jembatan antara pembuat handset dengan penyedia jaringan. Bahaya memutuskan untuk berbagi keuntungan dengan layanan Sidekick T-Mobile. Dengan demikian, Bahaya tidak mengandalkan penjualan handset sebagai satu-satunya sumber pendapatan, tapi juga layanan. Hal ini membuat produsen perangkat (Bahaya) memiliki tujuan yang sama dengan penjual (operator telekomunikasi T-Mobile).

bisnis ponsel dengan lebih agresif lagi. Nilai yang ditawarkan pun tak tanggung-tanggung. Menurut laporan Microsoft yang beredar membeli Bahaya dengan harga $ 500 juta. Namun, pembelian Bahaya oleh Microsoft tidak membawa hasil bunga. Eksekutif dibiarkan dari Danger yang didirikan oleh Microsoft ke divisi Komunikasi Bisnis, Hiburan dan Perangkat Mobile. Kemudian, mereka diminta mengembangkan ponsel yang dikenal dengan Project Pink. Sasarannya, ponsel ini semestinya menjadi saingan iPhone dan BlackBerry. Menurut Computerworld, Project Pink adalah penyakit klasik di sebuah perusahaan besar. Karena proyek ini cukup bergengsi, ia diperebutkan oleh beberapa pihak. Dan yang lebih parah lagi, perkembangan semakin menyimpang dari yang diinginkan. Misalnya, ponsel pada awalnya akan dikembangkan berdasarkan Java, namun kemudian diminta untuk menggunakan sistem operasi Microsoft.

Sayangnya, Windows Phone 7 yang seharusnya digunakan untuk Project Pink, belum siap. Alhasil, saat diluncurkan, ponsel yang akhirnya bernama Microsoft Kin ini menggunakan sistem operasi Windows untuk ponsel yang "sudah tua". Sebuah sambutan yang dingin setiap pasar membuat Kin akhirnya ditutup, hanya beberapa bulan sejak diluncurkan. Nasib layanan Sidekick, yang diwarisi Microsoft dari Bahaya, dan tidak terlalu bagus. Dalam satu kejadian, yang masih belum diketahui persis penyebabnya, pelanggan Sidekick mendadak kehilangan semua datanya. Satu hal yang perlu diperhatikan, semua data di Sidekick tersimpan 'di cloud' (dalam hal ini berjalan di server Microsoft dan bisa diakses melalui internet). Nah, saat server sedang down, semua data pengguna Sidekick hilang.

Pada awal 2002, Rubin memberi kuliah di Stanford tentang pengembangan Sidekick. Sebab, meski penjualan Sidekick di pasar tidak meledak, perangkat itu dinilai cukup bagus dalam hal rekayasa. Sebuah kebetulan bahwa Larry Page dan Sergey Brin, pendiri Google, juga hadir dalam ceramah ini. Setelah kuliah, lihat Rubin ke Page Sidekick dari dekat. Ternyata, tampilan halaman, perangkat tersebut menggunakan mesin pencari Google. "Keren," kata Page. Ini adalah titik awal bagi Page untuk gagasan bahwa dalam beberapa tahun kemudian akan terwujud, sebuah ponsel Google. Kira-kira dua tahun kemudian, Rubin telah meninggalkan Bahaya dan mencoba melakukan hal baru. Ini termasuk mencoba memasuki bisnis kamera digital sebelum ia mendirikan Android.

Mengandung Android Rubin saat menjadi wirausahawan-in-residence dengan perusahaan modal ventura Redpoint Ventures pada 2004. "Android berawal dari ide sederhana, menyediakan platform mobile yang tangguh dan terbuka sehingga bisa mendorong inovasi dan manfaat pelanggan semakin pesat," kata Rubin. . Pada bulan Juli 2005, 22 bulan setelah Android berdiri, perusahaan tersebut ditelan oleh Google raksasa. Rubin juga memilih untuk bergabung dengan Google. Saat membeli Android Inc., Google tidak menyebutkan secara detail berapa harga yang anda bayarkan dan apa yang harus dilakukan dengan perusahaan. Padahal, Google mengatakan itu pembelian sebagai akuisisi sumber daya manusia dan teknologi saja. Selain Andy Rubin, Google memang membuat banyak orang cemerlang dari Android. Ini termasuk Andy McFadden (pengembang WebTV dengan Rubin, dan juga pengembang Moxi Digital); Richard Miner (mantan wakil presiden di perusahaan telekomunikasi Orange) dan Chris White (pendiri Android dan perancang tampilan dan antarmuka WebTV).

Bersama dengan Google, Android diberi daya ekstra. Perusahaan dari Mountain View, California yang kemudian membentuk Open Handset Alliance untuk mengembangkan alat untuk Android. "*Google tidak bisa melakukan semuanya, dan kami tidak membutuhkannya. Makanya kami membentuk Open Handset Alliance dengan lebih dari 34 mitra,*" kata Rubin.

Perangkat Android yang hadir dipasaran tidak dibuat oleh Google. Pejuang kelas berat Android termasuk Motorola, Samsung, dan HTC masing-masing memasarkan ponsel andalan Android mereka ke pasar. "*Lepas saja perangkat lunaknya saja tidak cukup,*" *Rubin menjelaskan,* "*Anda membutuhkan handset yang dikembanglan untuk perangkat lunak dan penyedia jaringan yang ingin memasarkannya.*"

Di A.S., Motorola Droid jadi satu senjata melawan Verizon Wireless AT & T dengan iPhone-nya. Sementara Nexus One, ponsel Android besutan Google besutan HTC, hadir tanpa "ikatan resmi" pada satu operator tertentu. Kehadiran Android nampaknya mencoba mengguncang dominasi pasar ponsel di A.S .. Di Indonesia, Android juga nampak siap menjadi primadona setelah tampil dalam keributan di Indonesia Celullar Show 2010. "*Saya tahu akan ada FUD (ketakutan, ketidakpastian, keraguan) .Namun, kita telah melihat beberapa pesaing mengikuti apa yang kita lakukan. Jadi sepertinya kita berada di jalur yang benar,*" kata Rubin.

(Sumber: http://www.ib-article.com/2012/08/biography-of-andy-rubin-inventor-of.html)

Rangkuman

1. Fungsi md5 merupakan mekanisme pengacakan string menggunakan algoritma Message – Digest, dan akan mengembalikan hash atau chipertext
2. Session memiliki fungsi yang mirip dengan cookies, yaitu suatu teknik penyimpanan data. Bedanya data akan disimpan di sisi server
3. Enkripsi merupakan proses pengkodean data sehingga maknanya menjadi jelas. Proses ini sering dilakukan untuk meningkatkan keamanan data yang dikirim
4. Dalam program redirect.php telah didefinisikan bahwa halaman main_menu.php merupakan halaman yang diperuntukan bagi user setelah login diterima

Daftar Istilah

1. Plaintext
2. Chipertext
3. Kriptografi
4. Kriptoanalis
5. Kriptografer

Latihan Soal

1. Jelaskan pengertian Cookie dan Session?
2. Apa pentingnya Cookie dan Session ?
3. Jelaskan pengertian dari Enkripsi data?
4. Apa kegunaan Enkripsi data ?
5. Jelaskan pengertian dari Enkripsi data dengan Message – Digest?
6. Jelaskan pengertian dari Enkripsi data dengan Crypt?

KASUS : KEAMANAN INTERNET BAGI KEGIATAN ONLINE UKM

Keamanan Internet adalah saat user menggunakan internet, user megharapkan kerahasiaan dan integritas data. Selain itu juga, menjaga agar komputer ytidak terkena internet liar dimana mengakibatkan jenis ancaman yang dapat menyebabkan komputer untuk berfungsi secara ridak benar. Salah satu kasus dari keamanan internet yang lemah adalah bocornya LAN perusahaan yang menggunakan wireless LAN IEEE 802.11b. Dengan menggunakan sebuah notebook yang dilengkapi dengan perangkat IEEE 802.11b seorang peneliti sambil berjalan menunjukkan LAN dan data-data dari beberapa perusahaan yang bocor ke jalan di depan kantor.

Penggunaan firewall untuk membatasi akses ke kantor dari Internet akan sia-sia jika pintu belakang (backdoor) wireless LAN bisa digunakan oleh cracker untuk masuk ke sistem.

Program untuk memecahkan wireless LAN ini mulai banyak tersedia di Internet, seperti misalnya Airsnort, Netstumbler 1, WEPcrack, dan lain-lain. Penggunaan Firewall adalah salah satu cara yang dianjurkan untuk menjaga kejahatan internet. Firewall adalah sebuah sistem atau perangkat yang mengizinkan lalu lintas jaringan yang dianggap aman untuk dilintasi dan mencegah lalu lintas jaringan yang tidak aman. Berikut aplikasi firewall:

1. ***Ashampoo Firewall Free***
   Firewall ini sangat mudah digunakan, tidak mengharuskan kita untuk memiliki pengetahuan teknis. Hanya sedikit menggunakan memory dan sumber daya komputer serta berkemampuan memberikan perlindungan extra pada komputer.
2. ***Comodo Firewall***
   Firewall yang bekerja pada standar tindakan. Berkemampuan memeriksa daftar panjang lebih dari dua juta aplikasi, mudah digunakan dan mampu mengenali aman tidaknya file yang akan memasuki sistem.
3. ***Zone Alarm Free Firewall***
   Merupakan salah satu firewall yang paling banyak digunakan. Menawarkan perlindungan terhadap spyware dan memberikan peringatan dini ketika akan mendownload program jahat, sangat baik dalam pemantauan lalu lintas inbound dan outbound.
4. ***PC Tools Free Firewall***
   Secara berkesinambungan memberi perlindungan untuk memblokir program berbahaya yang melewati firewall. Penggunaan mudah untuk dan memiliki aplikasi untuk mencegah pengguna lain untuk mendapatkan akses ke komputer melalui internet.
5. ***Sygate Personal Firewall***
   Memberikan perlindungan dari gangguan kode berbahaya, ulah hacker, trojan horses, dan serangan DOS. Fitur baru termasuk deep-packet inspection, anti-application-hijacking, a log dampener, dan enhanced logging.

BAB 9

Pemahaman Tentang Grafik dan Paging

Tujuan Instruksional Umum

Setelah mempelajari bab ini, Mahasiswa dipastikan dapat:

1. Menjelaskan cara memanipulasi grafik
2. Membuat *paging* secara statis
3. Membuat *paging* secara dinamis
4. Menjelaskan langkah–langkah pembuatan web yang baik

BAB

9

PEMAHAMAN TENTANG GRAFIK DAN *PAGING*

9.1. Pendahuluan

Dukungan tampilan data dalam bentuk grafik mampu menyajikan tampilan halaman web menjadi menarik. Begitu juga ketika aplikasi web yang Anda buat sangat berkautan dengan pengolahan data semacam statistic, sepertinya merupakan suatu keharusan untuk membuat grafik. Dengan demikian, pengunjung akan semakin mudah untuk memahami data yang ditampilkan.

Pada dasarnya PHP menyediakan kumpulan fungsi tersendiri yamg memungkinkan Anda untuk membuat dan melakukan manipulasi grafik atau file gambar. Adapun dukungan terhadap format gambar juga semakin luas, sehingga memberikan keleluasaan lebih bagi kita dalam memasukan grafik ke halaman web.

Pembahasan lain yang juga akan dijelaskan adalah mengenai paging halaman web. Paging atau pemberian halaman web sangat diperlukan ketika memiliki data dalam bentuk record yang cukup banyak jumlahnya, dimana masalah yang sering muncul adalah ketika semakin banyaknya record yang ditampilkan. Adapun menampilkan semua record pada satu halaman tentu dianggap kurang begitu praktis, sehingga diperlukan paging. Hal ini akan Anda temui ketika Anda membuat aplikasi e-news, buku tamu, dan sebagainya.

9.2. Manipulasi Grafik

Tanpa menggunakan PHP, sebenarnya halaman web yang kita buat juga dapat disisipi dengan grafik dari file gambar. Akan tetapi, disini bukan seperti

itu yang kita maksudkan dengan menggunakan grafik. Lebih dari sekadar menyisipkan gambar. Kita akan menggunakan kumpulan fungsi image yang telah tersedia untuk membuat serta melakukan manipulasi grafik atau gambar.

Secara umum ada empat tahap dasar untuk membuat gambar menggunakan fungsi image.

1. MEMBUAT CANVAS GAMBAR

Untuk memulai pembuatan atau manipulasi gambar, kita perlu membuat identifier gambar terlebih dahulu. Cara pertama menggunakan fungsi ***imagecreate($lebar, $tinggi)***. Fungsi ini akan mengembalikan identifier untuk gambar baru. Cara kedua adalah membaca file gambar yang sudah ada untk kemudian kita manipulasi yaitu dengan menggunakn fungsi ***imagecreatefromgif()***, ***imagecreatefromjpeg()***, ***imagecreatefrompng(),*** dan fungsi lainnya sesuai format gambar yang digunakan.

2. MENGGAMBAR BENTUK ATAU TEKS PADA KANVAS.

Pertama kita alokasikan warna gambar melalui fungsi ***imagecolorrallocate().*** Misalnya untuk mengalokasikan warna hitam, perintahnya seperti berikut:

```
$hitam = imagecolorrallocate($im, 0, 0, 0);
```

Setelah alokasi warna, kita dapat membuat bentuk gambar atau teks yang diinginkan. Contoh grafiks teks dengan warna hitam.

```
Imagestring($im, 10, 60, 90, "Teks Horizontal",
$hitam);
```

3. MENAMPILKAN HASIL GAMBAR

Untuk gambar yang akan dihasilkan, misalnya dengan format gif, Anda perlu mengatur informasi header melalui fungsi ***header().***

```
Header ("Content-type: image/gif");
```

Selanjutnya memanggil tampilan gambar untuk dikelaurkan hasilnya. Pengiriman gambar ini menggunakan fungsi yang bergantung dari format gambar. Misalnya seperti berikut:

```
Imagegif($im)
```

4. MEMBERSIHKAN RESOURCE

Kita perlu menghapus identifier setelah tidak digunakan lagi, caranya adalah dengan menggunakn fungsi ***imagedestroy()*** setelah gambar ditampilkan.

Contoh pembuatan dan manipulasi grafiks teks pada listing program berikut:

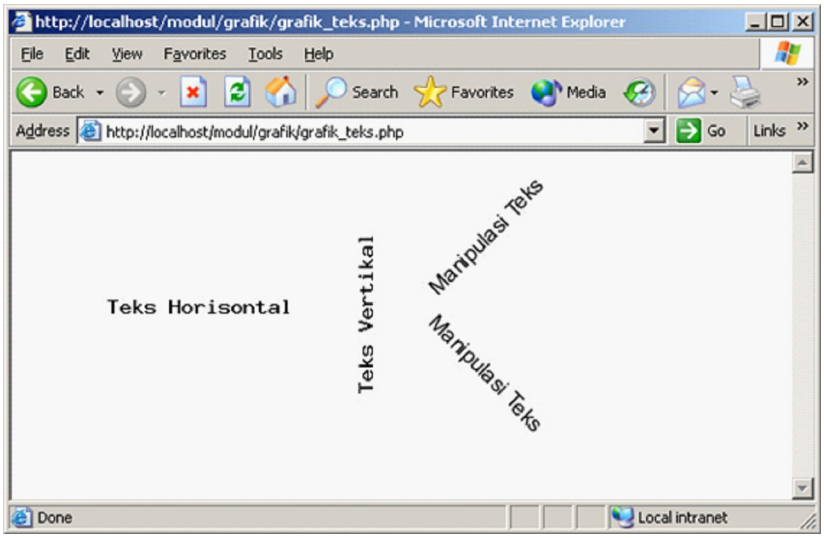
```
<?php
/* grafik_teks.php */
$im = ImageCreate(400,200);
$putih = ImageColorAllocate($im, 255, 255, 255);
$hitam = ImageColorAllocate($im, 0, 0, 0);

imagestring($im, 10, 60, 90, "Teks Horisontal", $hitam);
```

Secara umum terdapat 4 (empat) tahap dasar untuk membuat gambar menggunakan fungsi *image.*

1. Membuat canvas gambar
2. Menggambar bentuk atau teks pada kanvas.
3. Menampilkan hasil gambar
4. Membersihkan *resource*

```
// Membuat teks vertikal
imagestringUp($im, 10, 240, 160, "Teks Vertikal",
$hitam);
// Mengatur skala teks
imagettftext($im, 12, 45, 300, 90, $hitam,
"arial.ttf", "Manipulasi Teks");
imagettftext($im, 12, -45, 293, 110, $hitam,
"arial.ttf", "Manipulasi Teks");

header('Content-Type: image/jpeg');
imagejpeg($im);
imagedestroy($im);
?>
```

Disini mengunakan fungsi **imagettftext** yang akan menghasilkan teks ke gambar dengan fnt True Type (ttf). Bentuk font dapat ditentukan sendiri, misalnya arial.ttf. hasil listing program tersebut diatas adalah:



Gambar 9.1 Tampilan monitor mengunakan fungsi ***imagettftext***

9.3. Paging Halaman

9.3.1. *Paging* Halaman Secara Statis

Pertama persiapkan tabel yang akan digunakan:

```
CREATE TABLE BukuTamu (
id INTEGER(6) NOT NULL,
nama VARCHAR(30) NOT NULL,
email VARCHAR(25) NOT NULL,
komentar VARCHAR(220),
PRIMARY KEY (id)
);
```

Seperti yang sudah umum dilakukan, untuk mempermudahkan dalam melakukan koneksi database, buat file koneksi.inc. File ini nantinya juga diperlukan pada paging secara dinamis. Jadi akan praktis jika Anda membuat sendiri.

```
<?
// koneksi.inc

$host = "localhost";
$user = "root";
$pass = "";
$db   = "modul";

mysql_connect($host, $user, $pass);
mysql_select_db($db);
?>
```

9.3.1.1. Paging Statis Cara Pertama

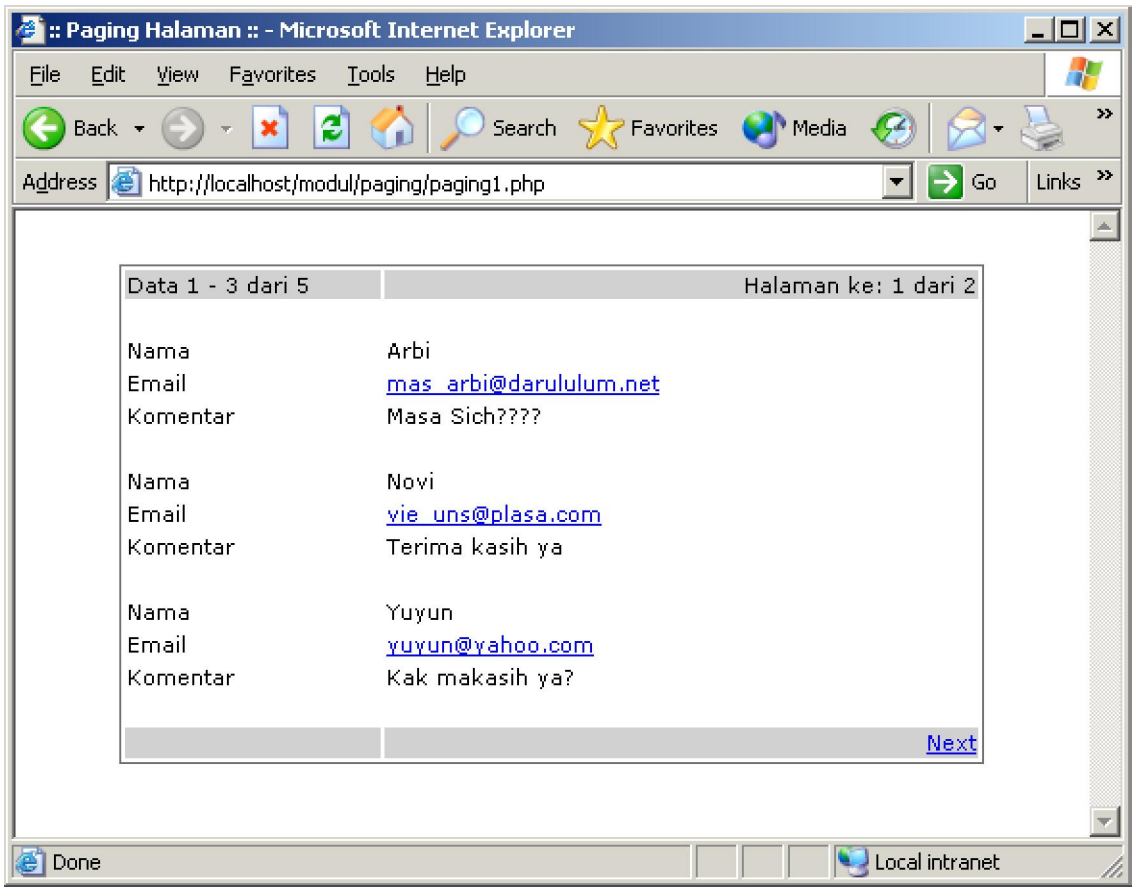
Pada prinsipnya pengguna hanya diperkenankan melakukan navigasi secara berurutan. Jadi ketika Anda sudah membuka halaman ketiga dan ingin kembali ke halaman pertama, maka harus terlebih dahulu melewati halaman kedua. Adapun listing programnya selengkapnya sebagai berikut:

```
<?php
/* paging1.php */

require_once "koneksi.inc";

$flname = basename($PHP_SELF);

// Pesan jika data tidak ada
$dbkosong = '<font color=red>
             <b>Ooops.... Data Tidak Ditemukan</b></font>';

$res = mysql_query("SELECT * FROM bukutamu ORDER BY id");
$jml = @mysql_num_rows($res);
  if ($jml == 0) {
      echo $dbkosong;
      exit;
  }

  // Jumlah baris yang ditampilkan
  $display = 3;

  // Inisialisasi variabel page
  if (isset($_GET["page"])) {
     $page  = $_GET["page"];
  } else {
     $page = 1;
  }

  // Mengatur data dalam array
  while ($tampil_data = mysql_fetch_array($res)) {
    $arrdata[] = $tampil_data;
  }
```

```
  // Hitung awal halaman
  $start = ($page * $display) - $display;
  $view  = array_slice($arrdata, $start, $dis-
play);
  $end   = $display * $page;

  // Menghitung total halaman
  if (($jml%$display) == 0) {
       $total = (int)($jml/$display);
  } else {
       $total = ((int)($jml/$display)+1);
  }

  // Kalkulasi data/row dan halaman
  echo "<table width='450' style='border:1pt solid
#666666;'>";
  echo "<tr><td width='30%' bgcolor='#CCCCCC'>
       Data ".($start+1)." - ".$end." dari $jml </
td>
       <td align=right width='70%'
bgcolor='#CCCCCC'>
       Halaman ke: $page dari $total</td></tr>
       <tr><td> </td></tr>";

  foreach ($view as $row) {
  echo "<tr><td>Nama</td><td>$row[1]</td></tr>";
  echo "<tr><td>Email</td><td><a
href='mailto:$row[2]'>"
       ."$row[2]</a></td></tr>";
  echo "<tr><td>Komentar</td><td>$row[3]</td></
tr>";
  echo "<tr><td> </td></tr>";
  }

  if ($start != 0) {
      $awal = $page-1;
      $prev = "<a
href='$flname?page=$awal'>Previous</a>";
  } else {
      $prev="";
  }

  if (($end < $jml)) {
      $berikut = $page+1;
      $next = "<a href='$flname?page=$berikut'>
Next </a>";
  } else {
      $next="";
  }
```

```
// Mengatur navigasi halaman
echo "<tr><td bgcolor='#CCCCCC' height=16> $prev</td>
     <td bgcolor='#CCCCCC' align=right
         width=40% height=16>$next</td></tr>";
echo "</table>";
?>
```

Hasilnya adalah sebagai berikut:



Gambar 9.2 Tampilan *paging halaman*

9.3.1.2. Paging Statis Cara Kedua

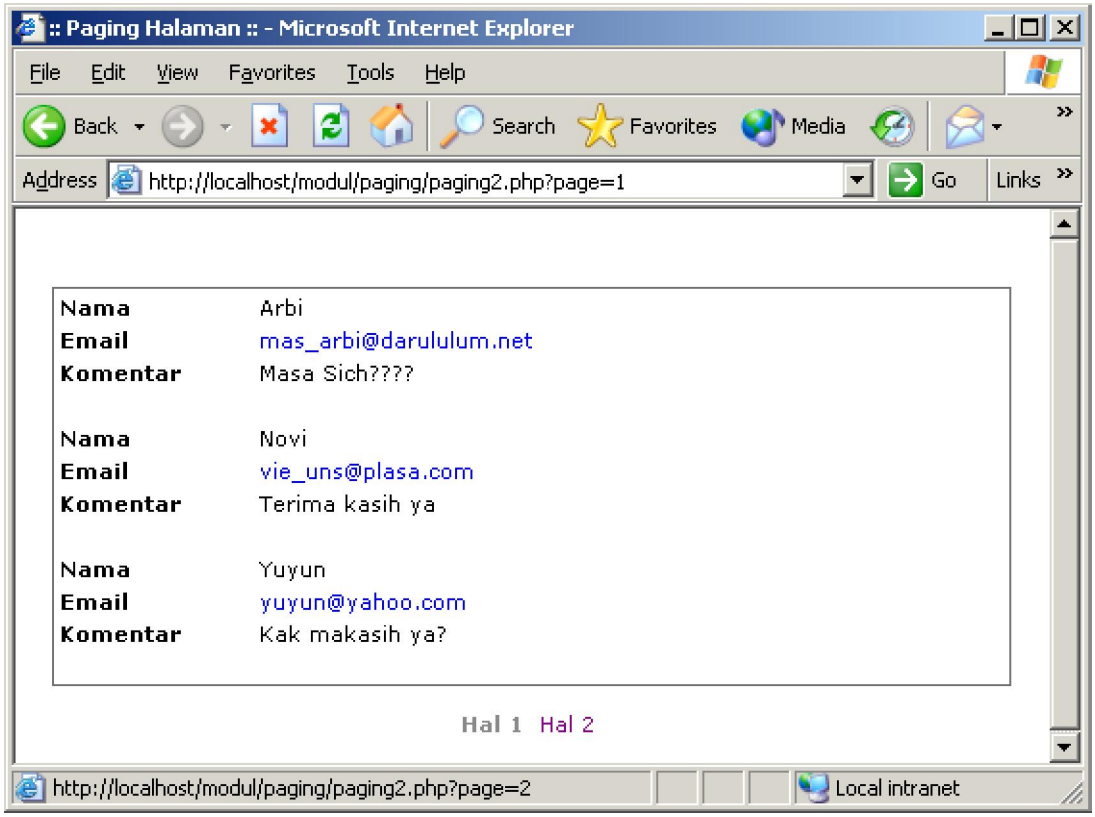
Pada paging cara kedua ini, yang ditekankan adalah keleluasan pengguna ketika melakukan navigasi. Untuk pembuatan paging dengan cara kedua ini, seluruh halaman dikumpulkan dalam bentuk link. Hal ini memungkinkan kita melakukan navigasi secara bebas atau tidak berurutan. Adapun ketika link bertepatan dengan halaman aktif yang ditampilkan, maka link di *disable.*

```
<?php
/* paging2.php */

require("koneksi.inc");

$flname = basename($PHP_SELF);
```

```
$dbkosong = '<font color=red>
             <b>Ooops.... Data Tidak Ditemukan</b></font>';

$res = mysql_query("SELECT * FROM bukutamu ORDER BY id");

$jml = @mysql_num_rows($res);
if ($jml==0) {
    echo $dbkosong;
    exit;
}

  // Batas baris per halaman
  $batas = 3;

  if (($jml%$batas) == 0) {
       $jmlpage=(int)($jml/$batas);
  } else {
     $jmlpage=((int)$jml/$batas)+1;
  }

  // Inisialisasi variabel page
  if (isset($_GET["page"])) {
     $page  = $_GET["page"];
  } else {
     $page = 1;
  }

  if ($page > $jmlpage) {
      $page=$jmlpage;
  }

  while ($rows = mysql_fetch_array($res)) {
      $arrdata[]=$rows;
  }

  $end  =($page*$batas)-1;
  $start=$end-($batas-1);
  if ($end>$jml) {
      $end=$jml-1;
  }


       for ($i=$start; $i<=$end; $i++) {
            $arr[] = $arrdata[$i];
       }

echo "<center><table width=500 style='border:1pt solid #666666;'>";

foreach ($arr as $row) {
    echo "<tr><td><b>Nama</b></td><td>$row[1]</td></tr>";
```

```
    echo "<tr><td width=100><b>Email</b></td><td>
         <a href='mailto:$row[2]'>$row[2]</a></td></tr>";
    echo "<tr><td><b>Komentar</b></td><td>$row[3]</td></tr>";
    echo "<tr><td> </td></tr>";
}
echo "</table> <br>";

// Memeriksa halaman aktif
for ($n=1; $n<=$jmlpage; $n++) {
     if ($n != $page) {
         echo " <a href='$flname?page=$n'>
               Hal $n</a> ";
     } else {
         echo "<font color='gray'><b>Hal $n</b></font>";
     }
}
?>
```

Hasilnya adalah sebagai berikut:



Gambar 9.3 Tampilan *paging2.php*

9.3.2. Paging Halaman secara Dinamis

Secara umum pembuatan paging yang telah kita lakukan sebelumnya mensyaratkan pendefinisan jumlah baris tiap halaman. Sebagai akibatnya, nilai variabel tersebut yang akan selalu digunakan ketika mem-*paging* halaman selanjutnya. Oleh karena itulah cara seperti itu kita anggap cara statis.

Adapun yang dimaksud cara dinamis adalah mengenai pengaturan batas paging. Dimana Anda dapat mengatur jumlah baris tiap halaman secara langsung pada aplikasi sekaligus melakukan paging.

```
<?php
/* paging_dinamis.php */

require_once "koneksi.inc";

echo "<form action=\"$PHP_SELF\" method=\"GET\">";
echo "<b>Batas Baris per Halaman :</b>
      <select name='batas'>
        <option value='3'>3
        <option value='5'>5
        <option value='10'>10
        <option value='10'>15
      </select> ";
echo "<input type=submit value='Terapkan'>";
echo "</form>";

$flname=basename($PHP_SELF);

$res = mysql_query("SELECT * FROM bukutamu ORDER BY id");

$jml = @mysql_num_rows($res);
if ($jml == 0) {
    echo "<font color=red>
          <b>Ooops.... Data Tidak Ditemukan</b></font>";
    exit;
}

// Inisialisasi nilai default batas per halaman
if (isset($_GET["batas"])) {
    $batas  = $_GET["batas"];
} else {
    $batas = 3;
}

if (($jml % $batas) == 0) {
    $jmlpage=(int)($jml/$batas);
} else {
    $jmlpage=((int)$jml/$batas)+1;
}

// Inisialisasi variabel page
if (isset($_GET["page"])) {
    $page  = $_GET["page"];
} else {
   $page = 1;
}
```

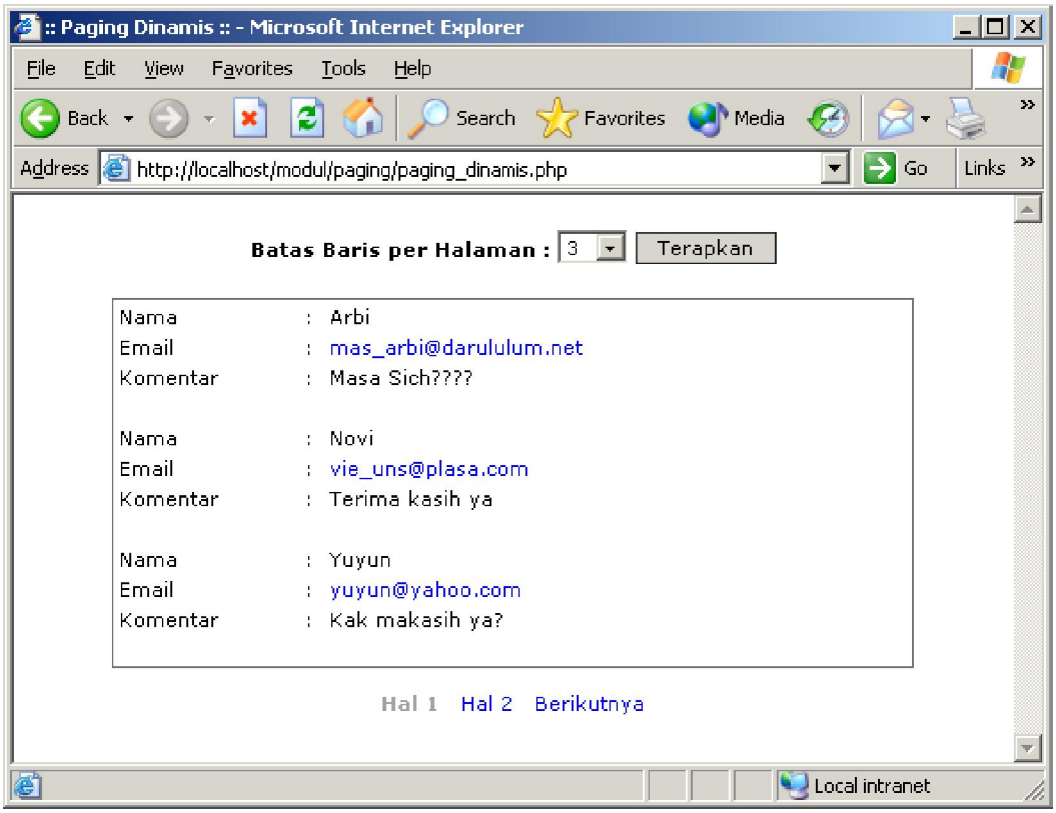
```
if ($page>$jmlpage) {
    $page = $jmlpage;
}

while ($rows = mysql_fetch_array($res)) {
    $arrdata[] = $rows;
}

$end  = ($page*$batas)-1;
$start= $end-($batas-1);
if ($end > $jml) {
    $end = $jml-1;
}

for ($i=$start; $i<=$end; $i++) {
     $arr[] = $arrdata[$i];
}
echo "<table width=450 style='border:1pt solid #666666;'>";
foreach ($arr as $row) {
    echo "<tr><td width=100>Nama</td>
              <td width=10>:</td><td>$row[1]</td></tr>";
    echo "<tr><td>Email</td><td>:</td><td>
          <a href='mailto:$row[2]'>$row[2]</a></td></tr>";
    echo "<tr><td>Komentar</td>
              <td>:</td><td>$row[3]</td></tr>";
    echo "<tr><td> </td></tr>";
}

echo "</table> <br>";

// Mengatur navigasi halaman
for ($n=1; $n<=$jmlpage; $n++) {
  $b = $page + 1;
  if ($n != $page) {
      echo " <a href='$flname?page=$n&batas=$batas'>
            Hal $n</a> ";
  } else {
      echo "<font color='#999999'><b>Hal $n </b></font>";
  }
}
// Navigasi halaman berikutnya
if (($n != $page) && ($n > $b)) {
    echo " <a href='$flname?page=$b&batas=$batas'>
          Berikutnya</a>";
}
?>
```

Sebagai nilai awal pengaturan halaman, kita memberikan batas tiga halaman. Nilai ini digunakan jika pengunjung tidak melakukan pengaturan sendiri, langkah ini diambil dengan memeriksa apakah variabel batas sudah diset apa belum.

```
// Inisialisasi nilai default batas per halaman
if (isset($_GET["batas"])) {
    $batas  = $_GET["batas"];
} else {
    $batas = 3;
}
```

Apabila ternyata variabel sudah diset, maka tentu saja nilai variabel default tidak digunakan. Sebagai gantinya, nilai dari pengunjung itulah yang akan berlaku.

Hasil paging dinamis adalah sebagai berikut:



Gambar 9.4 Tampilan *paging dinamis*

Dennis MacAlistair Ritchie | Penulis Bahasa Pemrograman C

Dennis MacAlistair Ritchie

Nama Lengkap:
Dennis MacAlistair Ritchie
Tempat/Tanggal Lahir:
Bronxville, New York, Amerika Serikat
9 September 1941
Meninggal dan dimakamkan:
12 Oktober 2011 (umur 70)
di Murray Hill, New Jersey, Amerika Serikat
Bidang Keilmuwan:
Ilmu komputer
Institusi: Lucent Technologies Bell Labs
Alma mater: Harvard University
Dikenal karena:
ALTRAN
B
BCPL
C
Multics
Unix
Penghargaan:
Turing Award
National Medal of Technology

(Sumber: https://id.wikipedia.org/wiki/Dennis_Ritchie)

Bagi Anda yang sudah lama berkecimpung diranah pemrograman pasti tidak asing lagi dengan sosok yang satu ini, Dennis M. Ritchie. Pria kelahiran 9 September 1941 ini adalah seorang ilmuwan komputer yang dikenal atas sokongannya membuat bahasa pemrograman C (dibaca: si) dan berbagai sistem operasi seperti Unix yang beliau kembangkan bersama dengan Ken Thompson. Beliau terpilih menjadi anggota National Academy of Engineering atas jasanya dalam "Development of the C Programming Language and for co-development of the Unix Operating System".

Dennis MacAlistair Ritchie (lahir 9 September 1941 – meninggal 12 Oktober 2011 pada umur 70 tahun) adalah seorang ilmuan komputer yang dikenal atas sumbangsihnya membuat bahasa pemrograman C dan berbaga sistem operasi seperti Multics dan Unix. Ritchie menerima penghargaan Turing Award pada tahun 1983 dan National Medal of Technology 1998 pada 21 April 1999. Ia menjabat sebagai kepala Departemen Riset Perangkat Lunak sistem di Lucent Technologies saat pensiun pada tahun 2007.

Dennis Ritchie ini dikenal masyarakat (dibaca: programmer) sebagai pembuat bahasa C? Penemuan bahasa C dan pengembangan sistem operasi Unix menjadikan beliau dan Ken Thompson sebagai pionir utama didunia komputasi modern. Bahasa C hingga kini masih tetap eksis digunakan secara luas dalam program aplikasi, sistem operasi, embedded system,dan pengaruhnya dalam bahasa pemrograman modern lainnya. Sistem Operasi Unix juga telah memberikan pengaruh besar dalam dunia komputasi modern.

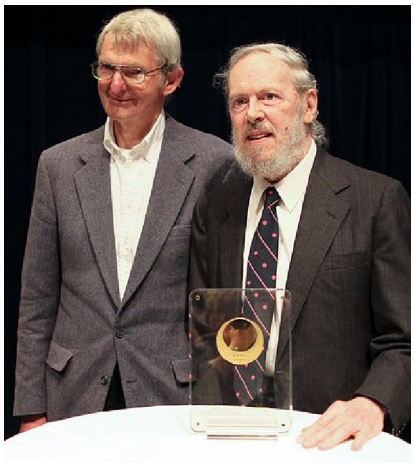
Mungkin bagi pemula di pemrograman belum/baru mengetahui bahwa kemunculan bahasa C ini melahirkan berbagai teknologi baik itu software maupun hardware. Dari bahasa pemrograman – sistem operasi – hingga perangkat smartphone. Lebih jelasnya, mari simak history dari bahasa C, ditunjukan pada gambar dibawah.Tahun 1972 – bahasa C diciptakan oleh Dennis Ritchie untuk sistem operasi Unix. Yang luar biasa, bahasa C mempengaruhi bahasa populer lainnya, terutama C++ (dibaca: si plus plus.) yang merupakan extensi dari C.

Tahun 1990-an – muncul bahasa pemrograman yang disebut Java, bahasa ini banyak mengadopsi sintaks dari C dan C++. Tahun 1965 – Unix, sistem operasi komputer diciptakan yang dilakukan AT&T, GE dan MIT. Kemudian pada tahun 1973 Unix ditulis ulang oleh Ken Thompson menggunakan bahasa C yang baru dikembangkan oleh Dennis Ritchie. Tahun 1991 – Linux diperkenalkan oleh penciptanya yang bernama Linus Torvalds. Dan Linux ini merupakan turunan dari sistem operasi Unix.

Tahun 2001 – Macintosh X dikenalkan oleh Apple, sistem operasi yang didasarkan dari NeXTstep yang menggabungkan stabilitas, dan kehandalan dan keamanan yang dimiliki Unix. Kemudian, menyebar lagi ke berbagai perangkat keras. Seperti iPhone dan Android yang didasari dari kernel Linux.

Latar Belakang

Ritchie lahir di Bronxville, New York. Pendidikan sarjana diraihnya dari Harvard di bidang Fisika dan Matematika terapan. Pada tahun 1967, Ritchie mulai bekerja di Bell Labs' Computing Sciences Research Center, kemudian di Lucent Technologies hingga ia pensiun pada tahun 2007. Pada tahun 1968, Ritchie mendapatkan gelar doktor Ph.D. dari Harvard. Ritchie (bersama Ken Thompson) menerima penghargaan Turing Award pada tahun 1983 dan National Medal of Technology 1998 pada 21 April 1999. Pada tahun 1988, Ritchie terpilih menjadi anggota National Academy of Engineering atas jasanya dalam "development of the 'C' programming language and for co-development of the UNIX operating system.



Dennis Ritchie di acara Japan Prize Foundation bersama mantan rekan Douglas (Doug) McIlroy pada 19 Mei 2011 di kantor pusat Bell Labs, Murray Hill, New Jersey.
(Sumber: http://www.flickr.com/photos/dpanyikdale/5740011186)

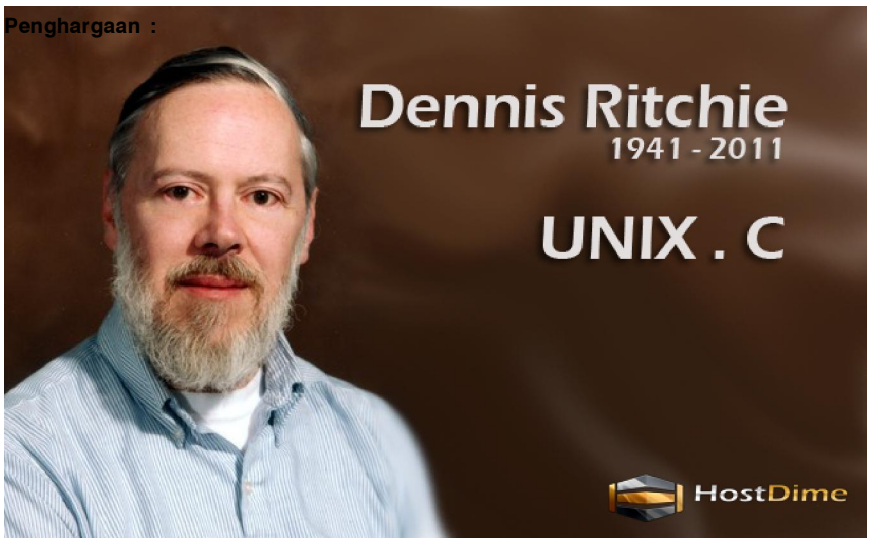
Bahasa C dan Unix

Ritchie dikenal sebagai pembuat bahasa pemrograman C, pengembang utama sistem

operasi, dan sebagai salah satu dari penulis (bersama Brian Kernighan) buku "kamus bahasa C," *The C Programming Language.*

Penemuan bahasa C dan pengembangan Unix menjadikan Ricthie dan Ken Thompson sebagai pionir utama dunia komputasi modern. Bahasa C hingga kini masih digunakan secara luas dalam program aplikasi, sistem operasi, pengembangan sistem benam, dan pengaruhnya dalam bahasa pemrograman modern lain. Sedangkan Unix juga telah memberikan pengaruh besar, membangun konsep dan prinsip dalam dunia komputasi masa kini.

Penghargaan :



(Sumber: http://blog.hostdime.com.br/wp-content/uploads/2011/10/dennis_ritchie-hostdime.png)

Turing Award

Pada tahun 1983, Ritchie bersama Ken Thompson menerima penghargaan Turing Award atas jasa mereka dalam pengembangan teori sistem operasi dan khususnya atas implementasi sistem operasi Unix. Pada kesempatan tersebut, Ritchie memberikan kuliah yang berjudul "Reflections on Software Research".

IEEE Richard W. Hamming Medal

Pada tahun 1990, Ritchi dan Ken Thompson menerima IEEE Richard W Hamming Medal dari Institute of Electrical and Electronics Engineers (IEEE), "sebagai penggagas sistem operasi UNIX dan bahasa pemrograman C".

National Medal of Technology

Pada 21 April 1992, Ritchie dan Ken Thompson menerima National Medal of Technology 1998 dari Presiden Bill Clinton atas jasa mereka dalam menemukan sistem operasi UNIX dan bahasa pemrograman C yang telah menyebabkan pengaruh besar bagi perangkat keras komputer, perangkat lunak, dan sistem jaringan serta merangsang pertumbuhan industri sehingga membawa Amerika memimpin di Era Informasi.

Japan Prize

Pada tahun 2011, Dennis Ritchie bersama Ken Thompson menerima penghargaan Japan Prize for Information and Communications atas jasa mereka sebagai pionir dalam pengembangan sistem operasi Unix.

Kematian

Dennis Ritchie ditemukan meninggal dunia pada 12 Oktober 2011 dalam usia 70 tahun, di rumah yang ditinggalinya sendiri, di Berkeley Heights, New Jersey. Berita kematiannya diketahui pertama kali dari seorang mantan koleganya, Rob Pike. Penyebab dan waktu kematian belum diketahui secara pasti. Ritchie diketahui berada dalam kondisi kesehatan yang lemah sejak beberapa tahun lalu setelah menjalani pengobatan kanker prostat dan penyakit jantung. Kematiannya hanya berselang satu minggu setelah kematian Steve Jobs, namun tidak terlalu banyak menjadi topik pemberitaan media massa.Seorang ahli sejarah komputer, Paul Ceruzzi mengatakan pengaruh keduanya dalam perkembangan dunia teknologi komputer adalah sebanding.

Rangkuman

1. Tanpa menggunakan PHP, sebenarnya halaman web yang kita buat juga dapat disisipi dengan grafik dari file gambar. Akan tetapi, disini bukan seperti itu yang kita maksudkan dengan menggunakan grafik
2. Dukungan tampilan data dalam bentuk grafik mampu menyajikan tampilan halaman web menjadi menarik. Begitu juga ketika aplikasi web yang Anda buat sangat bertautan dengan pengolahan data
3. Secara umum pembuatan paging yang telah kita lakukan sebelumnya mensyaratkan pendefinisan jumlah baris tiap halaman. Sebagai akibatnya, nilai variabel tersebut yang akan selalu digunakan ketika mem-paging halaman
4. Pada proses Paging, pengguna hanya diperkenankan melakukan navigasi secara berurutan. Jadi ketika Anda sudah membuka halaman ketiga dan ingin kembali ke halaman pertama, maka harus terlebih dahulu melewati halaman kedua.

Daftar Istilah

1. Grafik File
2. Canvas Gambar
3. Manipulasi Grafik
4. Fungsi *imagecreate*
5. Paging Static
6. Fungsi ***imagedestroy***
7. Listing Program
8. Paging Halaman

Latihan Soal

1. Sebutkan dan jelaskan 4 dasar untuk membuat gambar menggunakan fungsi *image!*
2. Apa tujuan membentuk empat fungsi dasar tersebut pada *point* 1?
3. Sebutkan *sorcode* dari 4 dasar untuk membuat gambar menggunakan fungsi *image!*
4. *Apa kegunaan utam*a Sorcode ? jabarkan!
5. Kenapa fungsi *image* terkesan sangat penting?

BAB 10

Pemahaman Tentang Pemrograman Berorientasi Obyek (OOP) dan Obyek

Tujuan Instruksional Umum

Setelah mempelajari bab ini, Mahasiswa dipastikan dapat:

1. Menjelaskan pengertian OOP
2. Menjelaskan kelas dan objek
3. Menjelaskan konstruktor dan destruktor
4. Menjelaskan polimorfisme

BAB

10

PEMAHAMAN TENTANG OOP DAN OBYEK

10.1. Apa Itu OOP?

Pemrograman Berorientasi Objek (OOP) tidak selalu mempunyai kesepakatan konsep yang sama dari satu bahasa komputer ke bahasa komputer lainnya. Namun ada tiga hal penting yang harus dipunyai oleh sebuah bahasa komputer untuk dapat disebut mendukung OOP:

- **enkapsulasi**, yang mencegah pengaksesan langsung ke fungsi atau variabel yang ingin dilindungi;
- **inheritance**, yang memungkinkan pendefinisian kelas baru yang diturunkan dari kelas yang sudah ada;
- **polimorfisme**, yang memungkinkan pengaksesan ke suatu metode bergantung dari tipe data objek yang diakses. Proses ini disebut juga dengan dynamic binding, yaitu penentuan alamat dari metode atau variabel yang akan diakses dilakukan pada saat run time.

Dari sini jelas bahwa PHP4 belum dapat dikategorikan sebagai bahasa yang mendukung OOP. Persyaratan enkapsulasi tidak dapat sepenuhnya diimplementasikan oleh PHP4, yang pada bahasa C++ atau Java direalisasikan dengan kata kunci **private** maupun **protected**. Pada PHP setiap atribut maupun metode masih dapat diakses langsung baik dari dari dalam kelas turunan, dari dalam kelas lainnya, maupun secara global.

Walaupun PHP4 tidak mengenal kata kunci abstract dan virtual seperti pada C++ maupun interface seperti pada Java, namun dukungan polimorfisme di PHP4 sudah diimplementasikan dengan cukup baik.

Faktor kekurangan lain seperti tidak adanya dukungan overloading metode (bukan overriding), tidak mengurangi pengguna PHP untuk menggunakan OOP dalam menulis program Web mereka, karena banyak faktor keuntungan yang dapat diperoleh seperti:

- kemudahan dalam mengelola kode OOP;
- kemudahan untuk memahami alur program;
- dan tidak kalah pentingnya adalah kemudahan penggunaan kode yang sama untuk aplikasi lainnya.

10.2. Kelas dan Objek

Dalam konteks PHP, kelas dapat dibayangkan sebagai kumpulan dari variabel dan fungsi yang akan menangani variabel tersebut. Kelas juga dapat dipandang sebagai bentuk proses enkapsulasi, walau tidak sepenuhnya kriteria ini terpenuhi seperti yang telah diutarakan sebelumnya.

Dalam contoh aplikasi nanti, kita akan mengambil tema koneksi ke database MySQL dan PostgreSQL yang direalisasikan dalam bentuk OOP. Kelas MySQLDb dipakai sebagai kelas dasar yang akan digunakan oleh kelas PostgreSQL sebagai kelas turunan. Namun agar lebih mudah dimengerti, pada pembahasan awal kita akan menggunakan kelas sederhana yang tidak berhubungan dengan kedua kelas tadi. Berikut ini adalah contoh penulisan sebuah kelas A:

```
class A {
  // Pendeklarasian variabel
  var $host_1;

  // Konstruktor
  function A() {
    ...
  }

  // Definisi metode MetodeB
  function MetodeB($args="") {
     ...
  }
}
```

Menurut Wikipedia Bahasa Indonesia Pemrograman berorientasi objek *(Inggris: object-oriented programming disingkat OOP)* mendefinisikan bahwa OOP merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam para-digma ini dibungkus dalam kelas-kelas atau objek-objek. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya,

10.3. Variabel Anggota

Variabel anggota (member variable) dideklarasikan dengan menggunakan kata kunci **var** diikuti dengan nama variabel. Namun penggunaan variabel anggota ini tanpa pendeklarasian sebelumnya tidak akan menyebabkan kesalahan pada PHP4. Variabel ini hanya boleh diinisialisasi dengan konstanta. Penginisialisasian dengan non-konstanta harus dilakukan di dalam konstruktor. Dalam OOP, terminologi lain untuk variabel anggota adalah properti atau atribut.

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik peranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.
(Sumber: https://id.wikipedia.org/wiki/Pemrograman_berorientasi_objek)

```
// Global variabel
$gHost = "172.20.172.10";

// Class
class A {
  // Benar, inisialisasi dengan konstanta
  var $host_1="localhost";
  // Deklarasi variabel host_2
  var $host_2;

  // Konstruktor
  function A() {
    global $gHost;

    // Benar, inisialisasi dengan non-konstanta
    $this->host_2=$gHost;

    // Benar, definisi variabel baru dari class
    $this->host_3="192.168.0.0";
  }
}
```

Tentang Metode

Cara mendefinisikan sebuah metode sama seperti pada fungsi. Hanya saja, proses ini harus dilakukan dalam lingkup kelas yang bersangkutan. Metode MetodeB() adalah salah satu metode yang didefinisikan pada kelas A.

Pada PHP4 tidak dikenal istilah overloading metode, yaitu pendefinisian beberapa metode berbeda yang mempunyai nama sama namun memiliki daftar argumen yang berbeda. Dalam PHP4 setiap metode harus memiliki nama berbeda. Walau begitu, PHP4 mengenal terminologi argumen default untuk fungsi atau metode. Di sini jumlah dan tipe data dari argumen yang digunakan pada pemanggilan metode dapat berbeda.

```
...
class A {
   ...
  // Metode B
  function MetodeB($args="") {
     ...
  }
}
...
```

Metode MetodeB() mempunyai argumen default string kosong jika dipanggil tanpa argumen. Dua kemungkinan untuk memanggil metode ini dengan benar adalah sebagai berikut:

```
ObjekA = new A;
ObjekA->MetodeB();
```

atau

```
ObjekA->MetodeB("localhost");
```

Pada penulisan program memakai gaya OOP, sangat disarankan untuk tidak mengakses atribut secara langsung, melainkan melalui metode. Di sini umumnya dipakai awalan **Set** untuk memberikan nilai yang baru sebuah variabel. Sedang untuk mengakses nilai atribut digunakan awalan **Get.**

```
function SetHost($host) {
  $this->host_1 = $host;
}

function GetHost() {
  return $this->host_1;
}
```

Cara ini berhubungan erat dengan prinsip enkapsulasi, yang pada PHP4 belum didukung sepenuhnya, sehingga secara tidak langsung menghilangkan keuntungan cara penulisan metode seperti di atas, karena tidak ada cara untuk melindungi atribut dari pengaksesan secara langsung.

10.4. Konstruktor dan Destruktor

PHP4 juga mengenal konstruktor yang dideklarasikan dengan memakai metode yang memiliki nama sama dengan nama kelas yang mendefinisikannya. Konstruktor ini akan dipanggil pada pembuatan sebuah objek dari kelas yang bersangkutan. Ada sedikit perbedaan antara PHP3 dan PHP4 dalam proses pemanggilan konstruktor—terutama pada kelas turunan—yang akan diterangkan di bagian akhir artikel.

Dalam contoh sebelumnya, metode A() adalah konstruktor dari kelas A. Di sini biasanya dilakukan inisialisasi terhadap variabel dengan nilai non-konstanta.

PHP tidak mengenal destruktor. Jika memang diinginkan, dapat digunakan fungsi PHP **register_shutdown_function()** yang akan dipanggil jika skrip PHP selesai dikerjakan.

10.5. Pembuatan Objek

Untuk dapat menggunakan kelas yang telah didefinisikan, kita perlu membuat instan dari kelas tersebut. Proses ini dapat dilakukan dengan bantuan operator new yang akan menghasilkan sebuah objek dari kelas bersangkutan. Dengan operator ini kita dapat membuat beberapa objek dari kelas yang sama. Karena objek adalah individu kelas, maka objek ini tidak mempunyai ketergantungan terhadap objek yang lain.

Setelah kita mempunyai sebuah objek, maka kita dapat mengakses variabel anggota maupun metode yang dimiliki kelas tersebut dengan bantuan operator ->.

```
// Pembuatan objek baru
ObjekA = new A;
```

KONSEP DASAR PHP

CLASS - kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh *'class of dog'* adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object.

Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

OBJEK - membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer; objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.

ABSTRAKSI - Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

```
// Pemanggilan metode MetodeB
ObjekA->MetodeB();
```

Cara Pengaksesan

Pengaksesan ke metode atau ke variabel di dalam lingkup kelas itu sendiri dilakukan dengan bantuan operator **$this->.** Sebagai contoh, untuk mengubah variabel $host_1 dalam metode **MetodeB(),** dapat dilakukan dengan **$this->host_1 = "localhost".**

10.6. Inheritance dengan Extends

Seperti telah disinggung sebelumnya, inheritance adalah proses pembuatan sebuah kelas yang diturunkan dari kelas lain yang sudah didefinisikan sebelumnya. Salah satu faktor keuntungannya adalah, kita hanya tinggal menambahkan atribut atau metode yang dibutuhkan dalam kelas baru yang akan kita buat. Terminologi kelas yang dijadikan dasar pembuatan kelas baru ini adalah kelas dasar (base class) atau kelas induk (parent class). Sedang kelas yang baru disebut juga kelas turunan (extended class) atau kelas anak (child class).

Pembuatan kelas turunan di PHP4 dapat dilakukan dengan operator **extends**. Namun harus diingat bahwa dalam PHP4 hanya diperbolehkan menggunakan satu kelas dasar. Penurunan dari beberapa kelas (multiple inheritance) tidak diperbolehkan.

Hal yang juga penting di sini adalah pemanggilan konstruktor kelas turunan. Pada PHP4, jika konstruktor kelas turunan tidak didefinisikan, maka otomatis konstruktor kelas dasar yang akan dipanggil, jika ada. Sedangkan pada PHP3, jika konstruktor kelas turunan tidak ada, konstruktor kelas dasar tidak akan dipanggil.

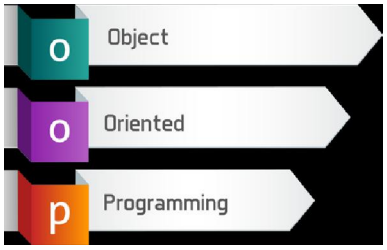
Berikut ini ilustrasi cara pembuatan kelas turunan dan pemanggilan konstruktor:

```
// Kelas dasar
class A {
  function A() {
    echo "Konstruktor class A.<br>\n";
  }
}

// Kelas turunan
class B extends A {
}
$ObjekB = new B;
```

Kelas B yang tidak mempunyai konstruktor diturunkan dari kelas A yang mempunyai konstruktor. Seperti diharapkan, pada pembuatan objek dari kelas B, maka konstruktor kelas A akan dipanggil. Program di atas akan menampilkan keluaran:

```
Konstruktor kelas A.
```

KONSEP DASAR PHP (lanjutan)



ENKAPSULASI - Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

POLIMORFISME melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari.

Dalam manual PHP4 dikatakan, jika kelas turunan tidak mempunyai konstruktor, maka konstruktor dari kelas dasar, yaitu metode yang mempunyai nama sama dengan nama kelas dasar, akan dipanggil. Namun pada PHP versi 4.0.4 maupun 4.0.5 hal ini belum dilakukan. Kita akan melihat kasus berikut.

Sekarang jika kita menambahkan sebuah metode di kelas A dengan nama **B()** (sengaja namanya sama dengan nama kelas B), dan kita jalankan lagi program di atas, maka yang keluar tidak seperti yang kita harapkan.

```
class A {
  ...
  // Metode tambahan
  function B() {
    echo "Metode normal B.<br>\n";
  }
}
```

Tampilan yang dikeluarkan adalah seperti berikut:

```
Metode normal B.
```

Di sini tiba-tiba metode **B()** di kelas A menjadi konstruktor dari kelas B yang akan dipanggil secara otomatis pada setiap pembuatan objek baru dari kelas A. Hal ini seharusnya hanya ditemui pada PHP3, dan tidak pada PHP4 seperti informasi di manual. Kemungkinan besar versi yang sekarang ada masih mengandung kesalahan.

Kelas turunan dapat melakukan overriding metode dari kelas dasar. Jika metode di kelas dasar yang sudah dioverride ini hendak dipanggil dari kelas turunan, maka dapat digunakan operator ::. Operator ini juga dapat dipakai dari lingkup global untuk memanggil metode dari suatu kelas tanpa membuat objek terlebih dahulu.

```
class A {
  function hallo() {
    echo "Metode hallo dari class A.<br>\n";
  }
}

class B extends A {
  // Override metode hallo dari class A
  function hallo() {
    echo "Metode hallo dari class B.<br>\n";
    // Memanggil metode hallo di class A
    A::hallo();  // Atau dapat juga
parent::hallo()
  }
}

// Memanggil metode tanpa objek
A::hallo();
$ObjekB = new B;
$ObjekB->hallo();
```

KONSEP DASAR PHP (lanjutan)

Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tungal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.

Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bag administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bag administrasi untuk mengambilnya.

Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas adminiistrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri.

(Sumber: http://newbie-21.blogspot.com/2014/08/apa-itu-oop-object-oriented-programming.html)

Program di atas akan menampilkan keluaran seperti berikut:

```
Metode hallo dari class A.
Metode hallo dari class B.
Metode hallo dari class A.
```

Untuk mengakses variabel anggota atau metode di kelas dasar tanpa harus tahu nama kelasnya, dapat digunakan kata kunci **parent**. Pada contoh di atas, jika baris **A::hallo()** diganti dengan **parent::hallo()** maka hasilnya akan identik.

10.7. Polimorfisme

Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut

Seperti dituliskan sebelumnya, polimorfisme adalah cara yang memungkinkan pengaksesan ke suatu metode bergantung dari tipe data objek yang diakses, termasuk dari metode di kelas dasar. Untuk memperjelasnya, mari kita lihat contoh berikut. Di sini kita akan membuat satu kelas dasar, **Bangun2D** dan dua kelas turunan, **Lingkaran** dan **PersegiPanjang**.

```
class Bangun2D {
  function Luas() {}
  function Keliling() {}
  function Info() {
    echo "Luas = ", $this->Luas(), ", Keliling =
", $this->Keliling();
    echo "<br>\n";
  }
}

class Lingkaran extends Bangun2D {
  var $r;
  function Lingkaran($r) { $this->r = $r; }
  function Luas() { return pi()*$this->r*$this->r;
}
  function Keliling() { return 2*pi()*$this->r; }
}

class PersegiPanjang extends Bangun2D {
  var $p;
  var $l;
  function PersegiPanjang($p, $l) { $this->p = $p;
$this->l = $l; }
  function Luas() { return $this->p*$this->l; }
  function Keliling() { return 2*($this->p+$this-
>l); }
}

$bangun[] = new Lingkaran(10);
$bangun[] = new PersegiPanjang(6, 8);

foreach ($bangun as $b) {
  $b->Info();
}
```

Program di atas akan menampilkan keluaran seperti berikut:

```
Luas = 314.15926535898, Keliling = 62.831853071796
Luas = 48, Keliling = 28
```

Dari contoh ini dapat dilihat, metode **Info**() kelas dasar **Bangun2D** tidak perlu mengetahui di kelas turunan mana ia sedang berada untuk mengetahui bagaimana cara menghitung luas dan keliling bangun. Jika metode ini dipanggil dari sebuah objek persegipanjang, maka **Luas**() dan **Keliling**() dari kelas Persegipanjang yang akan dipanggil; jika objek sebuah lingkaran, maka luas dan keliling lingkaran yang dihitung. Keuntungannya, jika suatu saat kita menambah bangun khusus lain, katakanlah **Segitiga** atau **Poligon**, maka metode **Info**() kelas **Bangun2D** tidak perlu diubah. Dengan kata lain, metode **Info**() menjadi reusable. Atau, bisa dikatakan juga bahwa OO dan polimorfisme mengizinkan kode lama (kelas **Bangun2D**) dapat memanggil kode baru (kelas **Segitiga** atau **Poligon** di masa depan).

10.8. Kelas MySQLDb dan PostgreSQLDb

Setelah kita belajar teori Pemrograman Berorientasi Objek dengan PHP, maka kita akan melihat penggunaannya dalam aplikasi yang nyata.

Kelas MySQLDb adalah kelas dasar yang mempunyai metode untuk melakukan operasi ke server database MySQL dengan instruksi sesederhana mungkin dalam pekerjaan yang kompleks. Sebuah objek dari kelas ini dapat melakukan operasi berulang kali, tanpa kehilangan informasi atau data dari operasi sebelumnya. Hal ini sangat bermanfaat terutama ketika melakukan query select.

Kelas PostgreSQLDb sebagai kelas turunan mempunyai fitur yang sama dengan kelas induknya MySQLDb.

Hubungan antara kedua kelas dalam contoh kita kali ini barangkali bukan teladan model abstraksi yang optimal. Dalam aplikasi abstraksi database yang lebih umum, biasanya dibuat satu kelas dasar abstrak yang mendefinisikan interface yang seragam, lalu dibuat kelas-kelas turunan sebagai *driver* untuk masing-masing jenis database. Jadi biasanya kelas MySQL dan PostgreSQL letaknya sederajat satu sama lain, di bawah satu kelas induk. Contohnya adalah di Perl dengan DBI sebagai kelas induk dan DBD::mysql serta DBD::Pg sebagai kelas-kelas driver. Namun demi kesederhanaan contoh, kita hanya akan membuat dua kelas seperti telah disebutkan sebelumnya.

10.9. Konstruktor MySQLDb

Kebanyakan aplikasi PHP mempunyai sebuah file konfigurasi yang berisikan banyak variabel global yang dibutuhkan oleh file lainnya. Pengguna aplikasi biasanya cukup mengedit file ini untuk menyesuaikan aplikasi di sistem yang ada.

PENGERTIAN CLASS dalam OOP/ Pemrograman Berbasis Objek

Class adalah 'cetak biru' atau *'blueprint'* dari object. Class digunakan hanya untuk membuat kerangka dasar. Yang akan kita pakai nantinya adalah hasil cetakan dari class, yakni object.

Sebagai analogi, class bisa diibaratkan dengan laptop atau notebook. Kita tahu bahwa laptop memiliki ciri-ciri seperti merk, memiliki keyboard, memiliki processor, dan beberapa ciri khas lain yang menyatakan sebuah benda tersebut adalah laptop. Selain memiliki ciri-ciri, sebuah laptop juga bisa dikenakan tindakan, seperti: menghidupkan laptop atau mematikan laptop.

Class dalam analogi ini adalah gambaran umum tentang sebuah benda. Di dalam pemrograman nantinya, contoh class seperti: koneksi_database dan profile_user.

Di dalam PHP, penulisan class diawali dengan keyword class, kemudian diikuti dengan nama dari class. Aturan penulisan nama class sama seperti aturan penulisan variabel dalam PHP, yakni diawali dengan huruf atau underscore untuk karakter pertama, kemudian boleh diikuti dengan huruf, underscore atau angka untuk karakter kedua dan selanjutnya. Isi dari class berada dalam tanda kurung kurawal.

(Sumber: http://www.duniailkom.com/tutorial-belajar-oop-php-pengertian-class-object-property-dan-method/)

Pengertian *Constructor* dalam OOP

Constructor (bahasa Indonesia: konstruktor) adalah metode khusus yang akan dijalankan secara otomatis pada saat sebuah objek dibuat (instansiasi), yakni ketika perintah *"new"* dijalankan.

Constructor biasa digunakan untuk membuat proses awal dalam mempersiapkan objek, seperti memberi nilai awal kepada property, memanggil method internal dan beberapa proses lain yang digunakan untuk 'mempersiapkan' objek.

Dalam PHP, constructor dibuat menggunakan method : __construct().

Demikian juga kelas MySQLDb dapat dikonfigurasi baik lewat daftar argumen maupun lewat variabel global di mana daftar argumen mempunyai prioritas lebih tinggi. Proses konfigurasi ini dilakukan oleh konstruktor.

```
class MySQLDb {
...
  function MySQLDb($args="") {
    global $cfgDBHost, $cfgDBPort, $cfgDBUser,
$cfgDBPasswd;

    $this->dbh    = false;

    if (isset($args["host"]))
      $this->host=$args["host"];
    else
      $this->host=$cfgDBHost;

    if (isset($args["port"]))
      $this->port=$args["port"];
    else
      $this->port=$cfgDBPort;

    if (isset($args["user"]))
      $this->user=$args["user"];
    else
      $this->user=$cfgDBUser;

    if (isset($args["passwd"]))
      $this->passwd=$args["passwd"];
    else
      $this->passwd=$cfgDBPasswd;
    $this->rownr=0;
  }
```

10.10. Metode Select()

Sebagian besar operasi ke database adalah query select. Metode Select() dalam kelas MySQLDb yang menangani operasi ini diimplementasikan sedemikian rupa, sehingga hasilnya disimpan dalam sebuah array yang sekaligus digunakan sebagai nilai kembali oleh metode ini. Dengan demikian data hasil query dapat diakses langsung tanpa harus melewati objek lagi.

```
  function Select($string) {
    $this->rowcount=0;
    $result = mysql_query($string, $this->dbh);
    $this->error=mysql_error();
    if (empty($this->error)){
      $entry = array();
      $this->rowcount=mysql_num_rows($result);
      $i=0;
```

```
      while($row=mysql_fetch_array($result)){
        $entry[$i] = $row;
        $i++;
      }
      mysql_free_result($result);
    }
    return $entry;
  }
```

10.11. Kelas PostgreSQLDb

Kelas PostgreSQLDb diturunkan dari kelas MySQLDb. Karena proses pengkonfigurasian sama seperti pada MySQL, maka konstruktor kelas ini hanya memanggil konstruktor dari base kelasnya.

```
class PostgreSQLDb extends MySQLDb {

  function PstgreSQLDb($args="") {
     parent::MySQLDb($args);
  }
...
}
```

Jika kelas PostgreSQLDb ini tidak diturunkan dari kelas MySQLDb maka kita harus menulis lagi konstruktor yang sama seperti pada konstruktor MySQLDb.

Untuk mengetahui metode lainnya, pembaca dapat melihat langsung source code dari kedua kelas tersebut.

Contoh Aplikasi

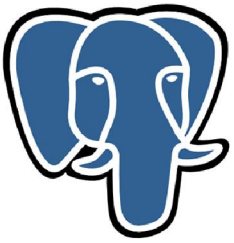
Di sini akan diberikan satu aplikasi sederhana yang melakukan query ke database MySQL atau PostgreSQL, dan menampilkan hasilnya ke browser. Variabel yang diperlukan dalam proses koneksi ke server database disimpan dalam file konfigurasi **config.inc.php** yang akan digunakan oleh konstruktor kelas dasar MySQLDb.

Setelah mengedit variabel global yang harus disesuaikan dengan sistem yang Anda pakai, maka untuk memilih kelas mana yang akan digunakan, pengguna cukup menuliskan nama kelas yang dilewatkan sebagai parameter fungsi **page_open()** yang diimplementasi dalam file **page.inc.php**.

```
// Menggunakan kelas MySQLDb
if (!page_open(array("db"      => 1,
                     "dbkelas" => "MySQLDb",
                     "page"    => $PHP_SELF)))
        die;
```

Jika ingin menggunakan kelas PostgreSQLDb maka Anda tinggal mengganti parameter "dbkelas"=>"MySQLDb" dengan "dbkelas"=>"PostgreSQLDb".

MENGENAL LEBIH JAUH TENTANG POSTGRESQL



PostgreSQL

PostgreSQL adalah sebuah proyek open source berbasis 100% komunitas, dikelola oleh komunitas di seluruh dunia lebih dari seribu kontributor. PostgreSQL menyediakan versi secara fungsional tunggal, daripada beberapa komunitas yang berbeda, komersial, dan versi perusahaan seperti yang ditawarkan MySQL. Lisensi adalah liberal BSD / MIT, yang memungkinkan organisasi untuk menggunakan, menyalin, memodifikasi, dan mendistribusikan ulang kode dengan hanya pemberitahuan hak cipta yang diperlukan.

Kehandalan adalah prioritas utama dari PostgreSQL. Dikenal sangat kokoh, dirancang dengan baik, mampu mendukung transaksi tinggi, dan aplikasi yang kritis. Dokumentasi adalah tingkat pertama, dengan manual yang komprehensif yang tersedia online secara gratis, bersama dengan arsip manual untuk versi yang lama. Dukungan komunitas sangat baik, dan dukungan komersial tersedia dari vendor yang independen.

Konsistensi dan integritas data juga prioritas tinggi. PostgreSQL sepenuhnya *ACID-compliant (Atomicity, Consistency, Isolation, Durability)*Memiliki keamanan yang kuat untuk mengontrol akses ke database, membuat baik penggunaan alat-alat keamanan perusahaan seperti Kerberos dan OpenSSL. Anda dapat menentukan cek Anda sendiri untuk memastikan kualitas data sesuai dengan aturan bisnis Anda sendiri.

Sebuah fitur favorit banyak admin adalah *point-in-time recovery* (PITR), fitur ketersediaan tinggi yang fleksibel dengan kekuatan seperti kemampuan untuk membuat server siaga untuk *failover*dengan cepat, dan membuat cadangan dan mengembalikan ke titik-titik tertentu dalam waktu. Tapi itu tidak semua, proyek juga menyediakan beberapa metode untuk mengelola PostgreSQL untuk ketersediaan tinggi, load-balancing, dan replikasi, sehingga Anda dapat menggunakan apa yang sesuai dengan kebutuhan khusus Anda.

(Sumber: http://hieppies.blogspot.com/2014/05/mana-yang-terbaik-dari-postgresql-atau.html/)

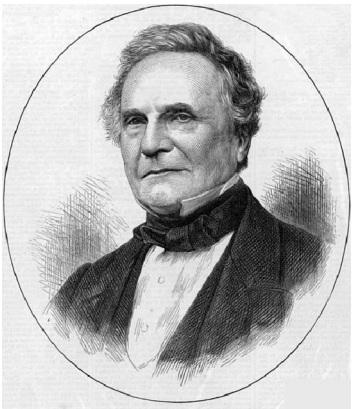
Pembuatan objek dari kelas sendiri akan ditangani oleh fungsi **page_open().** Pada PHP hal ini dimungkinkan karena kelas juga dapat disimpan dalam variabel seperti kita lihat di bawah ini:

```
function page_open($args) {
  global $cfgDatabase,$gDb;

  // Menggunakan database ?
  if (isset($args["db"])){
    // Kelas database ada di argumen?
    if (isset($args["dbclass"]))
      $gDb = new $args["dbclass"];
    // Memakai default
    else
      $gDb = new MySQLDb;

    return $gDb->Connect();
  }
  return true;
}
```

Object Oriented Programming

Variabel global $gDb adalah objek kelas yang namanya tersimpan dalam $args["dbkelas"].

CHARLES BABBAGE | Perancang mesin komputer



Charles Babbage. Potret obituari Charles Babbage (1791-1871) diterbitkan di *The Illustrated London News* pada tanggal 4 November 1871. Potret ini berasal dari foto Babbage yang diambil pada ***International Statistical Congress*** keempat yang berlangsung di London pada bulan Juli 1860.

(Sumber gambar: First published in 1871, copyright expired 70 years after publication, 1941.

Komputer telah menjadi bagian yang sangat penting dalam kehidupan masyarakat modern sekarang. Komputer modern dimungkinkan berkat perkembangan alat elektronik selama Perang Dunia II. Tapi, gagasan di balik komputer modern sebenarnya telah dipikirkan lebih dari seratus tahun sebelumnya oleh Charles Babbage. Sayang, teknologi pada zaman itu belum cukup maju sehingga dia tidak bisa menyaksikan penerapan gagasannya.

Charles Babbage (lahir 26 Desember 1791 – meninggal 18 Oktober 1871 pada umur 79 tahun) adalah seorang matematikawan dari Inggris yang pertama kali mengemukakan gagasan tentang komputer yang dapat diprogram. Sebagian dari mesin yang dikembangkannya, namun tidak selesai. Sekarang dapat dilihat di Museum Sains London. Pada tahun 1991, dengan menggunakan rencana asli dari Babbage, sebuah mesin diferensial dikembangkan dan mesin ini dapat berfungsi secara sempurna, yang membuktikan bahwa gagasan Babbage tentang mesin ini memang dapat diimplementasikan.

Charles Babbage lahir di Inggris, di jalan Crosby Row no 44, Walworth Road, London. Ada beberapa pendapat tentang tanggal kelahiran Babbage. Obituari yang dimuat dalam harian The Times menyebutkan kelahirannya pada tanggal 26 Desember 1792. Namun beberapa hari kemudian seorang keponakan Babbage menulis bahwa Babbage sebenarnya dilahirkan setahun sebelumnya, pada 1791. Telah terdaftar di Paroki St. Mary Newington, London, menunjukkan bahwa Babbage dibaptis pada tanggal 6 Januari 1792, itulah yang mendukung tahun kelahiran 1791.

Ayahnya, Benjamin Babbage, saudagar dan bankir kaya. Keluarga Babbage tinggal di Walworth, Surrey, di pinggiran kota London. Charles adalah putra pertama dari empat bersaudara, namun kedua saudara laki-lakinya meningggal pada masa kanak-kanak. Ketika Charles sakit parah tahun 1799, kedua orang tuanya takut kehilangan dia juga. Karena itu, dia dibawa ke Devon untuk mendapatkan udara pedesaan yang lebih sehat.

Charles memulai sekolahnya di Devon. Ia mempelajari antara lain matematika untuk navigasi sederhana dan akuntansi. Inilah awal dari minat yang akan membentuk kariernya. Kesalahan perhitungan dalam navigasi sering kali mengakibatkan kecelakaan kapal. Charles telah mengabdikan sebagian besar hidupnya untuk mengembangkan beberapa jenis mesin yang dapat menghitung dan mencetak tabel matematik dan astronomik secara cermat sehingga kesalahan dapat ditiadakan.

Charles Babbage's Difference Engine. Bagian dari mesin ciptaan Babage dirakit setelah kematiannya oleh anak Babbage, menggunakan bagian-bagian yang ditemukan di laboratoriumnya. Pada foto tersebut, bagian sisi kiri dikaburkan oleh pantulan dari kotak kaca. *(Sumber: Photograph © Andrew Dunn, 5 November 2004. Website: http://www.andrewdunnphoto.com)*

Nama Lengkap:
Charles Babbage

Lahir:
25 Desember 1792
London, Inggris

Meninggal:
18 Oktober 1871 (umur 79)
Marylebone, London, Inggris

Kebangsaan:
Britania Raya

Bidang:
Matematika
Ilmu komputer

Institusi:
Trinity College, Cambridge

Alma mater:
Peterhouse, Cambridge

Dikenal karena:
Matematika
Komputer

(Sumber: https://id.wikipedia.org/wiki/Charles_Babbage)

Ketika Charles sembuh dari sakitnya, ia kembali ke London. Dia bersekolah di Enfield, dan guru-gurunya segera melihat kemampuannya dalam bidang matematika. Tahun 1803, keluarganya pindah dan menetap di Devon. Charles bersekolah di Totnes Grammar School sampai tahun 1810, dan kemudian masuk Trinity College di Universitas Cambridge.

Pada masa itu, perhitungan dengan menggunakan tabel matematika sering mengalami kesalahan. Babbage ingin mengembangkan cara melakukan perhitungan secara mekanik, sehingga dapat mengurangi kesalahan perhitungan yang sering dilakukan oleh manusia. Saat itu, Babbage mendapat inspirasi dari perkembangan mesin hitung yang dikerjakan oleh Wilhelm Schickard, Blaise Pascal, dan Gottfried Leibniz. Gagasan awal tentang mesin Babbage ditulis dalam bentuk surat yang ditulisnya kepada Masyarakat Astronomi Kerajaan berjudul "*Note on the application of machinery to the computation of astronomical and mathematical tables*" ("catatan mengenai penerapan mesin bagi penghitungan tabel astronomis dan matematis") tertanggal 14 Juni 1822.

Charles Babbage mendapat gelar dalam bidang matematika tahun 1814. Tahun itu juga, dia menikah dengan Georgina Whitmore. Mereka memiliki delapan anak, tapi hanya lima yang hidup melewati usia kanak-kanak. Georgina meninggal tahun 1827. Tidak lama setelah menikah, Babbage memutuskan untuk menjadi pendeta. Dia melamar ke beberapa gereja. Sayang, para pemimpin gereja terlalu memercayai tuduhan bahwa Babbage adalah seorang liberal yang tidak patriotik, sehingga lamarannya ditolak.

Kerugian gereja menjadi keuntungan bagi matematika. Charles dan istrinya pindah ke London tahun 1815. Di sini, dia menunjukkan kemampuan praktisnya dalam matematika dan memberikan serangkaian ceramah mengenai manfaat eksperimen, di samping teori matematika. Berkat kegiatan ini, tahun 1816, dia terpilih sebagai anggota Royal Society -- perkumpulan paling bergengsi untuk para ilmuwan Inggris. Babbage memperoleh gelar master tahun 1817.

Selama beberapa tahun berikutnya, Babbage memberikan sumbangan penting dalam bidang matematika murni, seperti aljabar dan teori fungsi. Tapi keinginannya yang utama adalah mempraktikkan matematika. Dengan dukungan para ahli matematika, navigator, dan ilmuwan, dia mulai mengerjakan mesin analitis.

Mesin Analitik Babbage

Babbage terus meningkatkan kemampuan mesin perbedaannya hingga tahun 1830-an. Kemudian dia mendapat gagasan untuk menciptakan "mesin analitis". Mesin ini terdiri dari empat bagian gudang yang menjadi memori, pabrik tempat melakukan perhitungan matematika, suatu sistem roda gigi dan pengumpil untuk pemindahan data antara pabrik dan gudang, serta satu unit masukan/keluaran (susunan ini sesuai dengan susunan komputer modern, meskipun komponennya berbeda).

Gudang mesin analitis memakai roda dengan sepuluh posisi yang berbeda untuk menyimpan angka, sebagaimana dilakukan mesin perbedaan. Gudang itu bisa menyimpan sampai 1.000 angka dengan 50 digit setiap angka.

Ide tentang mekanisme masukan ini diperoleh Babbage dari sumber yang tidak biasa, yakni industri penenunan sutra Perancis. Tahun 1801, Joseph Marie Jacquard menciptakan mesin tenun yang memakai kartu berlubang-lubang untuk "memprogram" pola yang diinginkan ke dalam mesin tenun. Dengan demikian, pola yang sama bisa dicetak dalam jumlah banyak. Babbage menyadari bahwa sistem ini dapat dipakai untuk memasukkan data dan menyimpan instruksi ke dalam mesin.

Sayangnya, Babbage tidak berhasil membentuk model kerja untuk mesin analitisnya. Dia terus-menerus menghadapi kesulitan keuangan karena besarnya biaya untuk merancang dan membuat mesin baru. Tapi masalah terbesar adalah ketidakmampuan teknik rekayasa pada masa itu untuk menghasilkan komponen-komponen yang cukup akurat dan fleksibel. Kegagalan teknologi ini membuat Babbage sangat kecewa.

"Babbage mengupayakan sesuatu yang mustahil dengan sarana yang dia miliki. Namun, konsep dan asas di balik mesin analitis memang mutlak benar." Hal ini terungkap ketika buku catatan Babbage ditemukan tahun 1937 dan rancangannya dipelajari kembali. Dengan teknologi tahun 1940-an, komputer modern menjadi kenyataan.

Babbage tidak hanya merancang cikal bakal peranti keras komputer (mesinnya) masa kini, tapi juga telah mengonsepsikan unsur-unsur penting dari peranti lunak (program) komputer yang kita kenal sekarang. Konsepsi Babbage mengenai cara menyusun program mesin analitis sangat mirip dengan teknik yang dipakai untuk memprogram komputer modern.

(Sumber: https://id.wikipedia.org/wiki/Charles_Babbag; http://biokristi.sabda.org/charles_babbage_1791_1871*)*

Rangkuman

1. Pada PHP4 tidak dikenal istilah overloading metode, yaitu pendefinisian beberapa metode berbeda yang mempunyai nama sama namun memiliki daftar argumen yang berbeda
2. Kelas PostgreSQLDb diturunkan dari kelas MySQLDb. Karena proses pengkonfigurasian sama seperti pada MySQL, maka konstruktor kelas ini hanya memanggil konstruktor dari base kelasnya
3. Kebanyakan aplikasi PHP mempunyai sebuah file konfigurasi yang berisikan banyak variabel global yang dibutuhkan oleh file lainnya. Pengguna aplikasi biasanya cukup mengedit file ini untuk menyesuaikan aplikasi di sistem yang ada

Daftar Istilah

1. Kelas
2. Variabel Anggota
3. Inheritance
4. Polimorfisme
5. Kelas MySQLDb
6. Kelas Postgre SQLDb

Latihan Soal

1. Sebutkan dan jelaskan makna yang dipunyai oleh sebuah bahasa komputer untuk dapat disebut mendukung OOP!
2. Jelaskan Penertiaan dibawah ini?
   a. Kelas
   b. Variabel Anggota
   c. Inheritance
   d. Polimorfisme
   e. Kelas MySQLDb
   f. Kelas Postgre SQLDb
4. Jelaskan hubungan dari Kelas MySQLDb dan Kelas PostgreSQLDb
5. Mengapa Kelas MySQLDb dan Kelas PostgreSQLDb harus saling berhubungan satu sama lain ?

UNIVERSITAS NAROTAMA

BAB 11

Pemahaman Tentang Pembuatan Aplikasi E-Commerce

Tujuan Instruksional Umum

Setelah mempelajari bab ini, Mahasiswa dipastikan dapat Setelah mempelajari bab ini, Mahasiswa dipastikan dapat membuat aplikasi E-Commerce sederhana bagi UKM

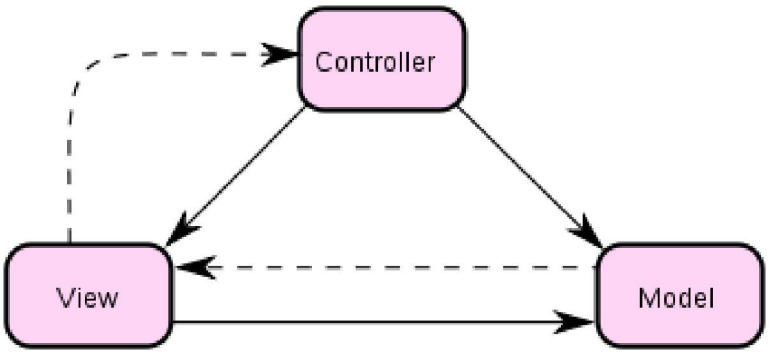
BAB

11

Pemahaman Tentang Pembuatan Aplikasi E-Commerce

11.1. Konsep MVC

Model-View-Controller atau **MVC** adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data *(Model)* dari tampilan (View) dan cara bagaimana memprosesnya *(Controller)*. Dalam implementasinya kebanyakan *framework* dalam aplikasi *website* adalah berbasis arsitektur MVC. MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna, dan bagian yang menjadi kontrol dalam sebuah aplikasi web.



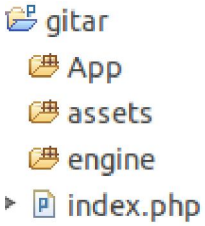
Gambar 11.1 Konsep Model-View-Controller (*sumber: id.wikipedia.org*)

1. **Model**, Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.

2. **View**, View adalah bagian yang mengatur tampilan ke pengguna. Bisa di katakan berupa halaman web.
3. **Controller**, Controller merupakan bagian yang menjembatani model dan view. Controller berisi perintah-perintah yang berfungsi untuk memproses suatu data dan mengirimkannya ke halaman web.

10.2. Struktur Direktori Aplikasi

Struktur folder dari aplikasi yang akan kita buat tampak pada gambar di bawah ini.



Gambar 11.2 Tampilan Struktur direktori aplikasi

Keterangan folder:

1. Engine.

Folder ini berisi *core (inti)* dari aplikasi mvc yang akan kita buat. Karena termasuk di dalam folder *system,* maka kita tidak akan terlalu banyak berurusan dengan isi dari folder ini.

2. Assets

Di dalam folder ini kita akan menyimpan file-file yg dapat di akses langsung oleh publik, seperti file css, javascript, dan gambar.

3. App

Di folder inilah nanti kita akan membuat aplikasi e-commerce kita. Menyimpan file *konfigurasi*, *controller*, *model* dan *template* dari aplikasi kita. Struktur folder di dalam folder ini seperti berikut.

```
1. App/
2.          Controller/
3.              IndexController.php
4.              FooController.php
5.          Model/
6.          View/
```

11.3. Membuat *Layout*

Kita mulai pembuatan aplikasi dengan membuat layout dari aplikasi yang akan kita buat. Layout inilah yang akan di kirim ke client melalui browser. Layout kita bagi menjadi beberapa bagian untuk mempermudah pengelolaan konten, yaitu header, sidebar, footer dan konten utama.

Gambar 11.3 Tampilan *layout*

Model View Controller (MVC)

Model View Controller merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman *Small Talk* (Wikipedia), MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user interface,* dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu MVC *pattern* dalam suatu aplikasi yaitu:

View, merupakan bagian yang menangani presentation logic. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh controller. View berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.

Model, biasanya berhubungan langsung dengan database untuk memanipulasi data *(insert, update, delete, search),* menangani validasi dari bagian *controller,* namun tidak dapat berhubungan langsung dengan bagian *view.*

Controller, merupakan bagian yang mengatur hubungan antara bagian model dan bagian *view, controller* berfungsi untuk menerima *request* dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.

Sumber :dari Wikipedia bahasa Indonesia ensiklopedia bebas (Dialihkan dari E-commerce)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Tran-
sitional//EN" "http://www.w3.org/TR/xhtml1/DTD/
xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/
html; charset=utf-8" />
<title>Untitled Document</title>

</head>

<body>

<div class="container">
  <div class="header"><a href="#"><img src=""
alt="Insert Logo Here" name="Insert_logo"
width="180" height="90" id="Insert_logo"
style="background: #C6D580; display:block;" /></a>
    <!-- end .header --></div>
  <div class="sidebar1">
    <ul class="nav">
      <li><a href="#">Link one</a></li>
      <li><a href="#">Link two</a></li>
      <li><a href="#">Link three</a></li>
      <li><a href="#">Link four</a></li>
    </ul>
    <p> </p>
    <!-- end .sidebar1 --></div>
  <div class="content">
    <h1>Content</h1>
    <p> </p>
<h4> </h4>
    <!-- end .content --></div>
  <div class="footer">
    <p>footer</p>
```

```
    <!– end .footer –></div>
  <!– end .container –></div>
</body>
</html>
```

```
  <div class="content">
    <h1>Content</h1>
    <p> </p>
<h4> </h4>
    <!– end .content –></div>
  <div class="footer">
    <p>footer</p>
    <!– end .footer –></div>
  <!– end .container –></div>
</body>
</html>
```

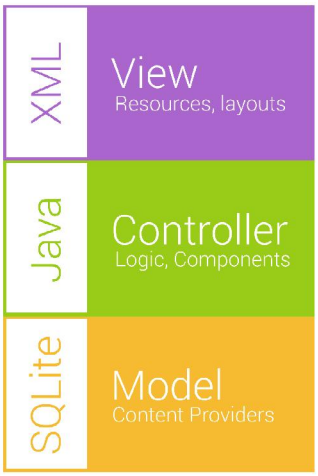
Simpan layout di atas dengan nama default.phtml ke dalam folder View/Layouts.

```
Controller/
Model/
View/
  Layouts/
        default.phtml
```

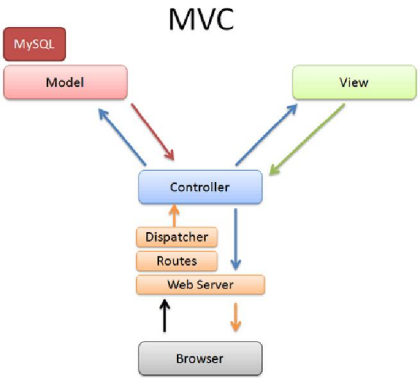
Dan file css berikut dengan nama layout.css ke dalam folder assets/css.

```
body {
      font: 100%/1.4 Verdana, Arial, Helvetica, sans-serif;
      background: #42413C;
      margin: 0;
      padding: 0;
      color: #000;
}

ul, ol, dl {
      padding: 0;
      margin: 0;
}
h1, h2, h3, h4, h5, h6, p {
      margin-top: 0;
      padding-right: 15px;
      padding-left: 15px;
}
a img {
      border: none;
}
```

Model View Controller (MVC)

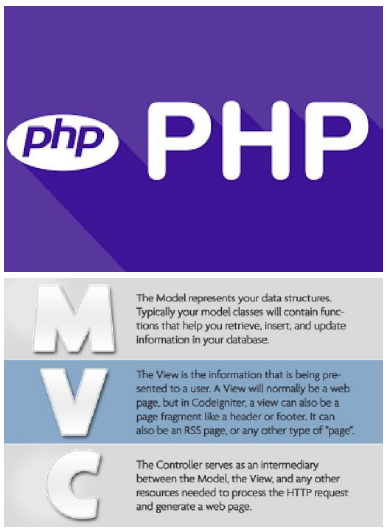


Singkat kata Model untuk mengatur alur *data base, View* untuk menampilkan *web,* sedangkan *Controller* untuk mengatur alur kerja antara *Model* dan *View.* Jadi misalnya Anda ingin membuat akun *facebook* atau *e-mail.* Pertama Anda akan melihat tampilan *sign-up/register* itulah *View,* kemudian Anda menuliskan *form* pengisian seperti *username, password,* dan lain-lain dan Anda *klik sign-up/register* maka di sini *View* memanggil *Controller* dan *Controller* memanggil *Model,* sedangkan *Model* mengecek apakah Anda sudah sesuai dengan kriteria pendaftaran seperti sudah mengisi *username, password* dan lain sebagainya. Kemudian Model menggembalikan *(callback)* ke *Controller* dan *Controller* mengembalikan ke View, dan kita akan melihat berhasil atau tidak kita *sign-up/register.*

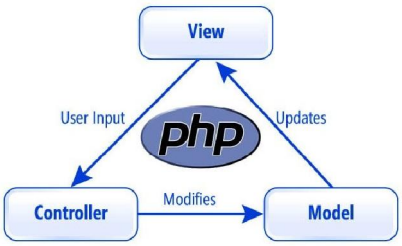


Sumber :dari Wikipedia bahasa Indonesia ensiklopedia bebas (Dialihkan dari E-commerce)

```
a:link {
    color: #42413C;
    text-decoration: underline;
}
a:visited {
    color: #6E6C64;
    text-decoration: underline;
}
a:hover, a:active, a:focus {
    text-decoration: none;
}

.container {
    width: 960px;
    background: #FFF;
    margin: 0 auto;
}


.header {
    background: #ADB96E;
}

.sidebar1 {
    float: left;
    width: 180px;
    background: #EADCAE;
    padding-bottom: 10px;
}
.content {

    padding: 10px 0;
    width: 780px;
    float: left;
}

.content ul, .content ol {
    padding: 0 15px 15px 40px;
}

ul.nav {
    list-style: none;
    border-top: 1px solid #666;
    margin-bottom: 15px;
}
ul.nav li {
    border-bottom: 1px solid #666;
}
ul.nav a, ul.nav a:visited {
    padding: 5px 5px 5px 15px;
```

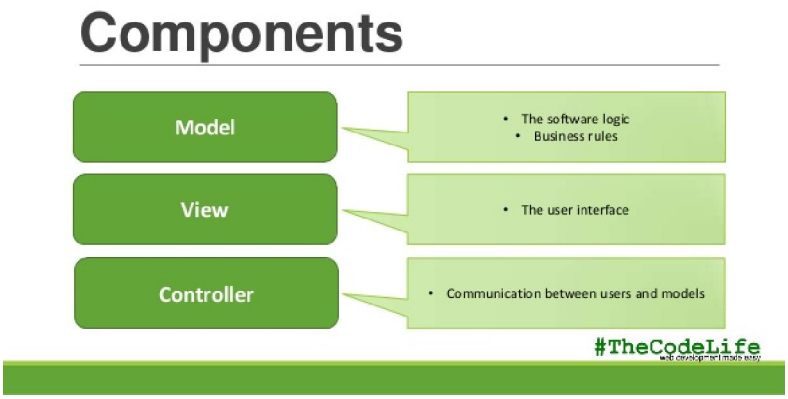


```
      display: block;
      width: 160px;
      text-decoration: none;
      background: #C6D580;
}
ul.nav a:hover, ul.nav a:active, ul.nav a:focus {
      background: #ADB96E;
      color: #FFF;
}

.footer {
      padding: 10px 0;
      background: #CCC49F;
      position: relative;
      clear: both;
}

/* ~~ miscellaneous float/clear classes ~~ */
.fltrt {  /* this class can be used to float an element right in your
page. The floated element must precede the element it should be next to
on the page. */
      float: right;
      margin-left: 8px;
}
.fltlft { /* this class can be used to float an element left in your
page. The floated element must precede the element it should be next to
on the page. */
      float: left;
      margin-right: 8px;
}
.clearfloat { /* this class can be placed on a <br /> or empty div as the
final element following the last floated div (within the #container) if
the #footer is removed or taken out of the #container */
      clear:both;
      height:0;
      font-size: 1px;
      line-height: 0px;
}


.container {
      width: 960px;
      background: #FFF;
      margin: 0 auto;
}


.header {
      background: #ADB96E;
}
```




```
.sidebar1 {
    float: left;
    width: 180px;
    background: #EADCAE;
    padding-bottom: 10px;
}
.content {

    padding: 10px 0;
    width: 780px;
    float: left;
}

.content ul, .content ol {
    padding: 0 15px 15px 40px;
}

ul.nav {
    list-style: none;
    border-top: 1px solid #666;
    margin-bottom: 15px;
}
ul.nav li {
    border-bottom: 1px solid #666;
}
ul.nav a, ul.nav a:visited {
    padding: 5px 5px 5px 15px;
    display: block;
    width: 160px;
    text-decoration: none;
    background: #C6D580;
}
ul.nav a:hover, ul.nav a:active, ul.nav a:focus {
    background: #ADB96E;
    color: #FFF;
}

.footer {
    padding: 10px 0;
    background: #CCC49F;
    position: relative;
    clear: both;
}

/* ~~ miscellaneous float/clear classes ~~ */
.fltrt {  /* this class can be used to float an
element right in your page. The floated element
must precede the element it should be next to on
the page. */
    float: right;
    margin-left: 8px;
}
```

```
.fltlft { /* this class can be used to float an element left in your
page. The floated element must precede the element it should be next to
on the page. */
      float: left;
      margin-right: 8px;
}
.clearfloat { /* this class can be placed on a <br /> or empty div as the
final element following the last floated div (within the #container) if
the #footer is removed or taken out of the #container */
      clear:both;
      height:0;
      font-size: 1px;
      line-height: 0px;
}
```




```
.fltrt {   float: right;
      margin-left: 8px;
}
.fltlft {  float: left;
      margin-right: 8px;
}
.clearfloat     clear:both;
      height:0;
      font-size: 1px;
      line-height: 0px;
}
```

Sisipkan file css diatas ke dalam tag <head> di layout tadi dengan kode berikut..

```
$this->Html()->css('layout')
```

Sehingga menjadi.

```
…….
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
<?php echo $this->Html()->css('layout') ?>
</head>
<body>
….
```

Layout sudah siap, sekarang coba kita akses halaman pertama kita. Ketik http://localhost/gitar di browser.

Gambar 11.4 Tampilan *Upload File*

Akan muncul exception dengan pesan "Halaman "index/index" tidak di temukan". Tentu saja, karena kita belum membuat controller untuk mengontrol halaman pertama yang kita request, maka aplikasi tidak dapat (baca: bingung) menampilkan apa yang akan di kirim ke browser.

11.4. Membuat Controller

Membuat *Controller*

Framework adalah kumpulan intruksi-intruksi yang di kumpulkan dalam *class* dan *function-function* dengan fungsi masing-masing untuk memudahkan *developer* dalam pemanggilannya tanpa harus menuliskan *syntax program* yang sama berulang-ulang. hal ini memiliki kegunaan untuk menghemat waktu dan mencegah penulisan *syntax* secara berulang-ulang agar terciptanya *source code* yang bersih dan terstruktur.

Codeigniter adalah sebuah framework php yang bersifat open source dan menggunakan metode MVC *(Model, View, Controller). codeigniter* bersifat free alias tidak berbayar jika anda menggunakannya. *framework codeigniter* dibuat dengan tujuan sama seperti *framework* lainnya yaitu untuk memudahkan *developer* atau *programmer* dalam membangun sebuah aplikasi berbasis *web* tanpa harus membuatnya dari awal.

Controller merupakan kumpulan intruksi aksi yang menghubungkan *model* dan *view*, jadi *user* tidak akan berhubungan dengan *model* secara langsung, intinya dari *view* kemudian *controller* yang mengolah intruksi.

Sumber :dari Wikipedia bahasa Indonesia ensiklopedia bebas (Dialihkan dari E-commerce)

Seperti di jelaskan pada bagian pertama dari halaman ini, **controller** merupakan bagian yang berfungsi untuk menjembatani antara data yang akan di proses dengan apa yang akan di tampilkan ke browser.

Untuk itu kita akan membuat sebuah objek di folder Controller. Controller di sini berupa objek untuk menghindari penggunaan global variable.

```
class IndexController extends
ZendX_Controller_Action
{

}
```

Controller diatas belum selesai karena kita belum membuat *method* untuk memproses halaman yang akan di request. Pada akhir bagian pembuatan layout kita mendapat exception "Halaman "index/index" ...", *index* pertama di sini merujuk ke nama controller, dan *index* kedua merujuk ke method dari controller. Untuk itu kita buat method index pada controller di atas, dengan namespace Action untuk menghindari collision dengan method selain request.

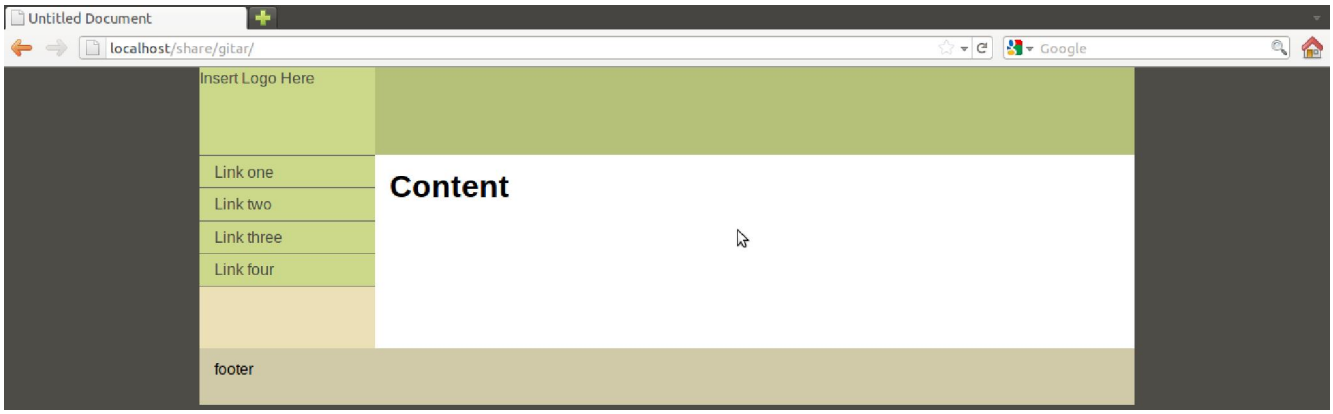
```
class IndexController extends
ZendX_Controller_Action
{

    public function indexAction()
    {

    }
}
```

Dengan memanggil method diatas, Controller secara otomatis akan mencari *view* dari method yang di panggil. Jika view yang di cari tidak ditemukan, controller akan mengirim exception. Untuk itu kita buat terlebih dahulu view yang di butuhkan oleh controller untuk mengirim data ke browser.

Buat file kosong dengan nama index.phtml dan simpan di folder View/Index. Sehingga tampilan direktori **View** kita menjadi seperti berikut.

```
View/
      Index/
            Index.phtml
Layouts/
            Default.phtml
```

Folder *Index* merujuk ke nama controller, dan file *index.phtml* merujuk ke method yang di panggil. Sehingga jika ingin membuat view dengan action index/home maka kita akan membuat file *home.phtml* di dalam folder *Index,* begitu hingga seterusnya. Sekarang panggil kembali aplikasi kita di browser.



Gambar 11.5 Tampilan Halaman Awal

Halaman awal aplikasi sudah muncul di browser, dengan layout yang telah kita buat tadi. Tetapi … bagaimana dengan view index.phtml tadi?

11.5. Membuat View

Pada aplikasi e-commerce yang akan kita buat, pada halaman awal kita akan menampilkan produk – produk terbaru yang kita masukkan ke database. Edit view file index.phtml tadi dengan kode berikut:

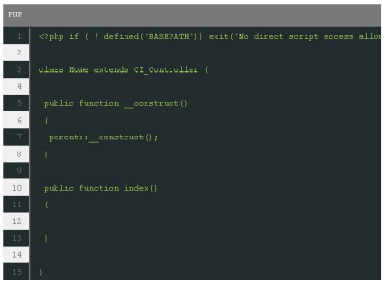
```
<div class="content">
  <h1>Welcome</h1>
    <table width="100%" border="0" cellpadding="0"
cellspacing="0" class="produk_list">
      <tr>
        <td><div class="title"><a href="tes">Nama
Produk</a></div>
          <table width="100%" border="0"
```

```
cellspacing="0" cellpadding="0">
          <tr>
            <td width="14%" rowspan="3"><img
src="assets/images/sampel.jpg" width="150"
height="125" /></td>
            <td width="86%">Harga: Rp.
<strong>250.000</strong>,-</td>
          </tr>
          <tr>
            <td>Yamaha L Series LL16 Dreadnought
Acoustic Guitar with Case Features:
              <ul>
                <li>Top: Solid Engelmann Spruce</
li>
                  <li>Scale Length: 650mm (25-9/
16&amp;quot;)</li>
            </ul></td>
          </tr>
          <tr>
            <td valign="top">&amp;nbsp;</td>
          </tr>
        </table>
        </td>
      </tr>

      <tr>
        <td><div class="title"><a href="tes">Nama
Produk</a></div>
          <table width="100%" border="0"
cellspacing="0" cellpadding="0">
          <tr>
            <td width="14%" rowspan="3"><img
src="assets/images/sampel.jpg" width="150"
height="125" /></td>
            <td width="86%">Harga: Rp.
<strong>250.000</strong>,-</td>
          </tr>
          <tr>
            <td>Yamaha L Series LL16 Dreadnought
Acoustic Guitar with Case Features:
      <ul>
                  <li>Body Wood: Solid Rosewood</
li>
                  <li>Top: Solid Engelmann
Spruce</li>
                  <li>Scale Length: 650mm (25-9/
16&amp;quot;)</li>
            </ul></td>
          </tr>
          <tr>
```

Membuat *Controller* (lanjutan)

Lagkah Pertama
Anda harus memiliki *framework codeingiter* .

Langkah Kedua
Run di folder *root server local* , kemudian eksekusi akan tampil seprti gambar di bawah ini, yang artinya *framework codeigniter* siap digunakan.



Langkah Ketiga
Saat membuka folder *controller* yang berada di folder *codeigniter* (nama folder saya di *local) > aplication > controller,* kemudian buat file baru dengan nama, Home.php, perlu ingat bahwa penulisan nama *controller* harus diawali dengan huruf besar, jika tulis dengan huruf kecil bisa menyebabkan *error* kalau sudah *diupload* ke *hosting* walaupun di *server local* berjalan dengan baik.

(Sumber: http://www.aguzrybudy.com/2016/08/cara-membuat-controller-di-framework.html)

```
            <td valign="top"> </td>
          </tr>
        </table>
        </td>
      </tr>
    </table>
</div>
          </tr>
        </table>
        </td>
      </tr>

      <tr>
        <td><div class="title"><a href="tes">Nama
Produk</a></div>
          <table width="100%" border="0"
cellspacing="0" cellpadding="0">
          <tr>
            <td width="14%" rowspan="3"><img
src="assets/images/sampel.jpg" width="150"
height="125" /></td>
            <td width="86%">Harga: Rp.
<strong>250.000</strong>,-</td>
          </tr>
          <tr>
            <td>Yamaha L Series LL16
Dreadnought Acoustic Guitar with Case Features:
      <ul>
                <li>Body Wood: Solid Rosewood</li>
                <li>Top: Solid Engelmann Spruce</li>
                <li>Scale Length: 650mm (25-9/
16&amp;quot;)</li>
            </ul></td>
          </tr>
          <tr>
            <td valign="top"> </td>
          </tr>
        </table>
        </td>
      </tr>
    </table>
</div>
```

Dan simpan css berikut ke dalam assets/css/table.css.

```
@charset "utf-8";
/* CSS Document */

.content .produk_list {
   margin: 10px;
   font-size: 12px;
}
```

... *Controller* (lanjutan)

Penjelasan:
Home adalah nama *class* dari *controller* yang di *extend* dari *CI_Controller,* harus diawali dengan huruf capital
function::__construct adalah tempat dimana Anda bisa me*load library, helper,* fungsi, ataupun variabel yang bisa diaksess di seluruh *conrtroller* yang masih berada di *Home controller.*
function index adalah *controller* yang pertama kali diakasess ketika kita menjalankan *Home controller.*

(Sumber: http://www.aguzrybudy.com/2016/08/cara-membuat-controller-di-framework.html)

Sejarah Dabatabase MySQL

MySQL pada awalnya diciptakan pada tahun 1979, oleh Michael "Monty" Widenius, seorang programmer komputer asal Swedia. Monty mengembangkan sebuah sistem database sederhana yang dinamakan UNIREG yang menggunakan koneksi low-level ISAM database engine dengan indexing. Pada saat itu Monty bekerja pada perusahaan bernama TcX di Swedia.

TcX pada tahun 1994 mulai mengembangkan aplikasi berbasis web, dan berencana menggunakan UNIREG sebagai sistem database. Namun sayangnya, UNIREG dianggagap tidak cocok untuk database yang dinamis seperti web.

TcX kemudian mencoba mencari alternatif sistem database lainnya, salah satunya adalah mSQL (miniSQL). Namun mSQL versi 1 ini juga memiliki kekurangan, yaitu tidak mendukung indexing, sehingga performanya tidak terlalu bagus.

Dengan tujuan memperbaiki performa mSQL, Monty mencoba menghubungi David Hughes (programmer yang mengembangkan mSQL) untuk menanyakan apakah ia tertarik mengembangkan sebuah konektor di mSQL yang dapat dihubungkan dengan UNIREG ISAM sehingga mendukung indexing. Namun saat itu Hughes menolak, dengan alasan sedang mengembangkan teknologi indexing yang independen untuk mSQL versi 2.

Dikarenakan penolakan tersebut, David Hughes, TcX (dan juga Monty) akhirnya memutuskan untuk merancang dan mengembangkan sendiri konsep sistem database baru. Sistem ini merupakan gabungan dari UNIREG dan mSQL (yang source codenya dapat bebas digunakan). Sehingga pada May 1995, sebuah RDBMS baru, yang dinamakan MySQL dirilis.

David Axmark dari Detron HB, rekanan TcX mengusulkan agar MySQL di 'jual' dengan model bisnis baru. Ia mengusulkan agar MySQL dikembangkan dan dirilis dengan gratis. Pendapatan perusahaan selanjutnya di dapat dari menjual jasa "support" untuk perusahaan yang ingin mengimplementasikan MySQL. Konsep bisnis ini sekarang dikenal dengan istilah Open Source.

Pada tahun 1995 itu juga, TcX berubah nama menjadi MySQL AB, dengan Michael Widenius, David Axmark dan Allan Larsson sebagai pendirinya. Titel "AB" dibelakang MySQL, adalah singkatan dari "Aktiebolag", istilah PT (Perseroan Terbatas) bagi perusahaan Swedia.

(Sumber: http://www.aguzrybudy.com/2016/10/sejarah-dabatabase-mysql.html)

```
.content table tr td .title {
    font-family: "Times New Roman", Times, serif;
    font-weight: bold;
    margin-bottom: 5px;
    font-size: 16px;
    border-top-width: 1px;
    border-top-style: solid;
    border-top-color: #333;
    margin-top: 10px;
}
```

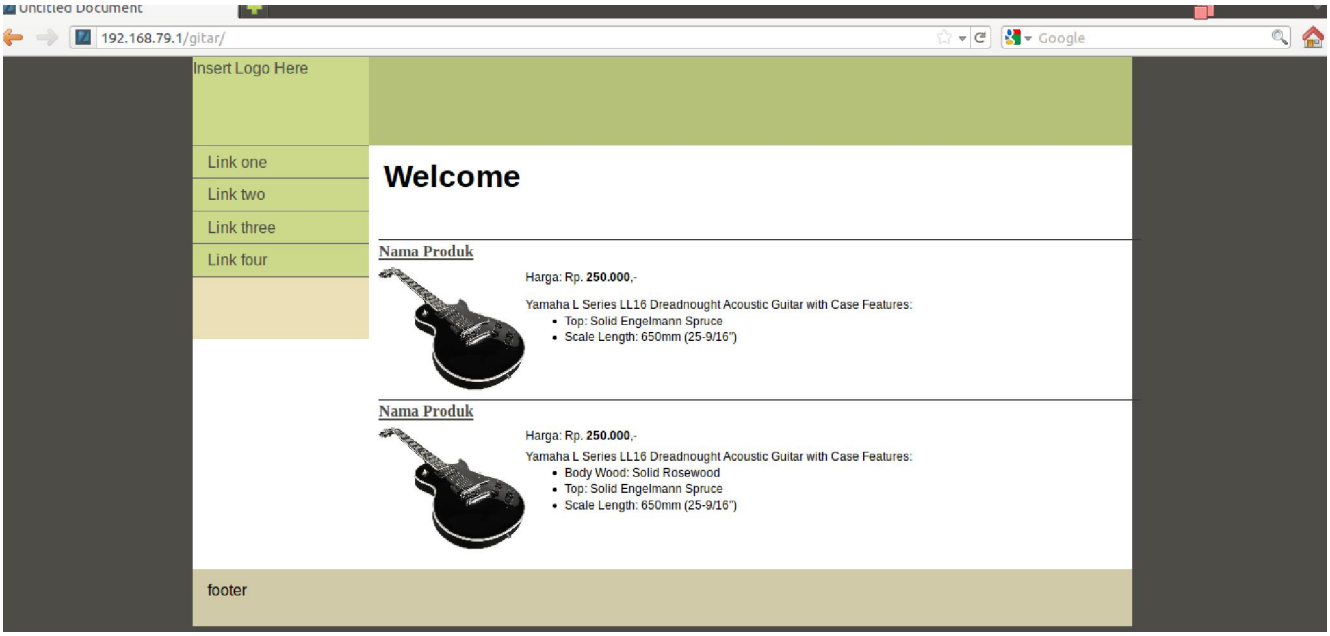
Kemudian edit layout view tadi dengan kode berikut.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
<?php
    echo $this->Html()->css('layout');
    echo $this->Html()->css('tabel');
?>
</head>
<body>
<div class="container">
  <div class="header"><a href="#"><img src="" alt="Insert Logo Here" name="Insert_logo" width="180" height="90" id="Insert_logo" style="background: #C6D580; display:block;" /></a>
    <!-- end .header --></div>
  <div class="sidebar1">
    <ul class="nav">
      <li><a href="#">Link one</a></li>
      <li><a href="#">Link two</a></li>
      <li><a href="#">Link three</a></li>
      <li><a href="#">Link four</a></li>
    </ul>
    <p> </p>
    <!-- end .sidebar1 --></div>

  <?php echo $this->layout()->content ?>

  <div class="footer">
    <p>footer</p>
    <!-- end .footer --></div>
  <!-- end .container --></div>
</body>
</html>
```

Dengan menambahkan code:

```
<?php echo $this->layout()->content ?>
```

Sekarang kita bisa menampilkan view yang diminta oleh controller. Sehingga tampilan halaman awal aplikasi menjadi seperti gambar di bawah ini.



Gambar 11.6 Tampilan Tampilan Halaman Awal

GRACE MURRAY HOPPER | Penyusun Konsep COBOL

Grace Murray Hopper salah satu wanita pelopor komputer elektronik.

Grace Murray Hopper (lahir 9 Desember 1906 – meninggal 1 Januari 1992 pada umur 85 tahun) adalah seorang ilmuwan komputer dan Laksamana Angkatan Laut Amerika Serikat. Menjadi pelopor di bidangnya, ia adalah salah satu pemrogram komputer Harvard Mark I, dan mengembangkan kompilator pertama untuk bahasa pemrograman komputer. Ia menyusun konsep bahasa pemrograman independen dari mesin yang berujung pada pengembangan COBOL, salah satu bahasa pemrograman modern pertama. Hopper juga dianggap sebagai tokoh yang memopulerkan istilah "debugging" untuk memperbaiki gangguan pada komputer (terinspirasi dari ngengat yang ditemuinya pada komputernya). Karena banyaknya prestasi dan pangkatnya di angkatan laut, ia dijuluki dengan "Amazing Grace". Kapal penghancur Angkatan Laut Amerika Serikat, USS Hopper (DDG-70), dinamai berdasarkan namanya, begitu juga dengan superkomputer "Hopper" Cray XE6 di NERSC dan ruang belajar Universitas Teknologi Chalmers.

Kehidupan dan karier

Hopper terlahir dengan nama Grace Brewster Murray di New York City. Ia adalah anak tertua dari tiga bersaudara. Hopper kecil adalah seorang anak yang selalu ingin tahu; saat berusia tujuh tahun, ia memutuskan untuk mengetahui bagaimana sebuah jam alarm bekerja, dan membongkar tujuh buah jam alarm sebelum ibunya memarahinya. Pada usia 16 tahun, ia ditolak masuk Vassar College karena nilai pelajaran bahasa Latinnya terlalu rendah. Ia akhirnya lulus dari Vassar pada tahun 1928 dengan gelar sarjana dalam bidang matematika dan fisika, dan meraih gelar master dari Universitas Yale pada tahun 1930. Pada tahun 1934, ia meraih gelar Ph.D. dalam bidang matematika dari Yale di bawah bimbingan Øystein Ore. Disertasinya yang berjudul "Jenis Baru Kriteria Iredusibilitas" diterbitkan pada tahun yang sama. Hopper mulai mengajar matematika di Vassar pada 1931, dan diangkat sebagai profesor pada 1941.

Pada tahun 1943, saat Perang Dunia II meletus, Hopper memperoleh cuti dari Vassar dan bergabung dengan Reservasi Angkatan Laut Amerika Serikat, menjadi satu dari banyak perempuan yang menjadi sukarelawan di WAVES. Pada tahun 1949, Hopper

Grace Hopper, pada usia 76 tahun, dipromosikan menjadi Komodor di Gedung Putih pada tahun 1983. (nampak Presiden Ronald Reagan dibelakangnya) Pangkatnya ditugaskan kembali pada tahun 1985 kepada Laksamana Muda.

Nama Lengkap:
Grace Murray Hopper

Kebangsaan:
Amerika Serikat

Tempat dan Tanggal Lahir:
New York City, New York, Amerika Serikat pada 9 Desember 1906

Meninggal:

1 Januari 1992 (umur 85) di Arlington, Virginia, Amerika Serikat

Dimakamkan di:

Pemakaman Nasional Arlington

Almamater:
Horace Greeley High School di Chappaqua, New York (1977-1981). Utica College, Utica, New York, *Bachelor of Science* dalam ilmu komputer (1981-1986)

Julukan:

"*Amazing Grace*"

Pengabdian

Dinas Angkatan Laut AS

Lama dinas 1943-1966, 1967-971, 1972-1986

Pangkat: Laksamana

Penghargaan yang diperoleh:

Defense Distinguished Service Medal
Legion of Merit
Meritorious Service Medal
American Campaign Medal
World War II Victory Medal
National Defense Service Medal
Naval Reserve Medal

(Sumber: https://id.wikipedia.org/wiki/Grace_Hopper)

bekerja sebagai karyawan di *Eckert-Mauchly Computer Corporation,* menjabat sebagai ahli matematika senior dan bergabung dengan tim pengembang UNIVAC I. Pada tahun 1952, ia menciptakan sebuah kompilator operasional, dan pada 1954 Hopper ditunjuk sebagai direktur pemrograman otomatis perusahaan. Hopper dan departemennya merilis beberapa bahasa pemrograman berbasis kompilator pertama, termasuk MATH-MATIC dan FLOW-MATIC.

Grace Murray Hopper dengan papan ketik UNIVAC, 1960

(Sumber: http://flickr.com/photo/8212496@N06/493885707 using Flickr upload bot)

Pada musim semi 1959, konferensi dua hari yang dikenal dengan Konferensi Bahasa Sistem Data CODASYL digelar oleh para ahli komputer, bersama dengan kalangan industri dan pemerintah. Dalam konferensi itu, Hopper menjabat sebagai konsultan teknis, dan ia bersama mantan karyawannya mengembangkan bahasa pemrograman baru yang dikenal dengan COBOL.

Pensiun dan kematian

Hopper pensiun dari Angkatan Laut pada usia 60 tahun, sesuai dengan peraturan yang berlaku, dengan pangkat komandan pada tahun 1966.[15] Ia kemudian dipanggil kembali untuk bertugas pada Agustus 1967 selama enam bulan, dan pensiun kembali pada tahun 1971. Hopper dipromosikan sebagai kapten pada tahun 1973 oleh Laksamana Elmo R. Zumwalt, Jr. Setelah pensiun, Hopper bekerja sebagai konsultan senior di Digital Equipment Corporation, posisi yang dijabatnya hingga kematiannya pada tahun 1992 di usia 85 tahun. Ia dimakamkan secara militer di Arlington National Cemetery.

Kehidupan Pribadi

Hopper menikah dengan seorang profesor Universitas New York bernama Vincent Foster Hopper (1906–1976[17]) dari tahun 1930 hingga perceraian mereka pada 1945. Setelah itu ia tak pernah menikah lagi dan tetap menggunakan nama belakang Hopper.

Penghargaan dan Pengakuan

1969 : Hopper dianugerahi penghargaan "computer sciences man of the year" oleh Data Processing Management Association.[18]

1971 : Penghargaan tahunan Grace Murray Hopper Award for Outstanding Young Computer Professionals didirikan pada 1971 oleh Association for Computing Machinery.

1973 : Hopper menjadi orang Amerika Serikat pertama dan wanita pertama di dunia yang dianugerahi Distinguished Fellow of the British Computer Society.

1982 : Ia menerima American Association of University Women Achievement Award.

1986 : Menerima Defense Distinguished Service Medal.

1987 : Menjadi penerima Computer History Museum Fellow Award.[19]

1988 : Menerima Golden Gavel Award di konvensi Toastmasters International di Washington, DC.

1991 : Menerima National Medal of Technology.

1996 : USS Hopper (DDG-70) diluncurkan. Dijuluki dengan Amazing Grace, ini adalah satu-satunya kapal militer AS yang dinamai berdasarkan nama wanita.

2001 : Eavan Boland menulis sebuah puisi yang didedikasikan untuk Grace Hopper berjudul "*Code*".

2009 : *National Energy Research Scientific Computing Center* menamai kapalnya "Hopper".

2013 : Tampil di fitur *Google Doodle* pada halaman utama mesin pencari Google.

(Sumber: https://id.wikipedia.org/wiki/Berkas:Commodore_Grace_M._Hopper,)

Rangkuman

1. **Model-View-Controller** atau **MVC** adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (Model) dari tampilan (View) dan cara bagaimana memprosesnya (Controller)
2. Pembuatan aplikasi dengan membuat layout dari aplikasi yang akan kita buat. Layout inilah yang akan di kirim ke client melalui browser
3. **Controller** merupakan bagian yang berfungsi untuk menjembatani antara data yang akan di proses dengan apa yang akan di tampilkan ke browser.
4. MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna, dan bagian yang menjadi kontrol dalam sebuah aplikasi web

Daftar Istilah

1. Controller
2. Folder Indeks
3. Global variabel
4. File Home
5. Folder View
6. Engine
7. Exception
8. Halaman Indeks
9. Edit View File

Latihan Soal

1. Bagaimana cara memproses Model dan *View* dalam sebuah proses *controller?* uraikan jawaban anda dan boleh sertakan contoh
2. Mengapa *Client* perlu dan harus ditransfer ke dalam *browser* melalui sebuah *layout?* apakah dapat dilakukan dengan cara lain ?
3. *Controller* merupakan bagian yang berfungsi untuk menjembatani antara data yang akan di proses dengan apa yang akan di tampilkan ke *browser.* Mengapa ?
4. Apa yang dimaksud istilah *"Exception"*? apa fungsi utamanya?
5. Dalam proses pengelolaan *layout* mengapa *header, sidebar, footer* dan konten utama harus disajikan dalam konten yang terpisah kedalam beberapa bagian?
6. Apa yang dimaksud Struktur, Direktori dan Aplikasi? jelaskan masing-masing dengan singkat.

KASUS : JARINGAN INTERNET MEMBANTU HIDUPNYA UKM

Saqina Butik, salah satu toko perlengkapan muslim terlengkap yang sudah buka di lima kota di Indonesia, yaitu Jakarta, Karawang, Purwakarta, Jombang dan Mojokerto. **Saqina Butik** yang diluncurkan sejak 2004 ini awalnya hanya dipasarkan melalui toko *riil* yang ada di kawasan Duren Tiga, Jakarta Selatan. Namun karena pesatnya perkembangan teknologi, pada 2008 Rosihan mulai mengembangkan Saqina via layanan internet, dengan cara membuat toko online. Bicara mengenai produk, sebagai *one-stop shopping*, Saqina menyediakan berbagai macam kebutuhan fashion keluarga muslim.

Mulai baju muslim, kaus, peci, kerudung, mukena, sajadah, hingga aksesori seperti bros dan tasbih. Saqina juga menyediakan produk lain mulai buku agama, CD lagu religi, hingga lukisan kaligrafi. Rosihan menyatakan, sejak dibuka toko online, sebagian besar omzet yang didapatkan saat ini berasal dari penjualan via internet tersebut. Persentase pembelian online bisa mencapai angka 70%.

Hingga saat ini, sebagian besar omzet yang didapat Saqina berasal dari pembelian konsumen via layanan internet (online). Dari total omzet yang didapat per bulan, 70% disumbang penjualan via website, sedangkan 30% berasal dari konsumen yang datang langsung ke toko kami, ujarnya kepada Seputar Indonesia. Perkembangan usaha online Saqina dibuktikan dengan banyaknya pengunjung via dunia maya itu. Setiap hari sedikitnya ada 3.000 pengunjung website.

Dari jumlah itu, 15% pengunjung pasti melakukan transaksi pembelian. Menurut Rosihan, menggunakan layanan internet untuk menjalankan usaha merupakan salah satu cara tepat dalam era perkembangan teknologi seperti sekarang ini. Selain tidak membutuhkan modal yang besar untuk menyewa tempat, menjual lewat internet juga mudah dilakukan semua kalangan. Kendati begitu, Rosihan mengingatkan pelaku bisnis online. Bagi mereka yang tidak memiliki toko *riil*, para pelaku bisnis online tetap harus memperhatikan tampilan dan kelengkapan data dari toko online yang dimiliki.

Baik dengan cara menampilkan foto produk yang akan dipasarkan maupun menuliskan tentang data diri pemilik, alamat, dan kontak telepon yang dapat dihubungi. *Dengan hal tersebut, para pelanggan biasanya yakin dan berkeinginan membeli barang di toko online yang kita dirikan*, ungkapnya. Karena itu, Rosihan menyarankan agar para pebisnis online tidak hanya sekadar menampilkan testimoni pada website, tetapi juga harus menuliskan informasi jelas mengenai toko online yang dijalankan, seperti menampilkan profil perusahaan (*company profile*).

Di Indonesia, penggunaan *e-commerce* memang masih dalam tahap awal. Meningkatnya kehadiran UKM secara online cenderung lebih banyak melalui situs media atau situs web statis yang tidak memiliki fasilitas transaksi online. Namun, penggunaan *forum online* untuk mempertemukan pembeli dan penjual sudah lumrah. Mengakses produk dan jasa baru via internet dan mencari harga yang lebih murah, seperti melalui akses ke layanan keuangan yang lebih luas, sangat berpotensi meningkatkan kemakmuran.

(Sumber : http://seputar-indonesia.com)

UNIVERSITAS NAROTAMA

DAFTAR PUSTAKA

Anonim. 2006. *Menguasai Pemrograman Web dengan PHP 5*. Yogyakarta :Wahana Komputer dan Andi Offset.

Arbie. *Manajemen Database dengan MySQL*,Andi Offset.Yogyakarta 2004

Tutang, *Belajar Cepat HTML*, Medikom Pustaka Mandiri, 2001

Bunafit Nugroho. (2008), *Aplikasi Pemrograman Web Dinamis Dengan PHP dan MySQL*, Gava Media, Yogyakarta.

Chaudhury, Abijit & Jean-Pierre Kuilboer (2002), *e-Business and e-Commerce Infrastructure*, McGraw-Hill

Dadan Sutisna, (2007), *7 Langkah Mudah Menjadi Webmaster, Mediakita*, Jakarta Selatan.

Hakim, L., Musalini, U. 2004. *150 Rahasia dan Trik Menguasai PHP*. Jakarta : Komputindo.

Hermanwan, E. (2008).*Definisi & Konsep web 2.0* (Januari 7, 2005).

Jonathan Sarwono dan Tutty Martadireja, (2008), *Teori E-Commerce Kunci Sukses Perdagangan Di Internet*, Gava Media, Yogyakarta.

Kadir, A. 2002. *Pemrograman Web mencakup : HTML, CSS, Java Script & PHP*. Yogyakarta : Andi Offset.

Kessler, M. (2003). *More shoppers proceed to checkout online.*

Kurniawan, R. 2008. *Membangun Situs dengan PHP untuk Orang Awam*. Palembang.

Madcoms, (2007). *Seri panduan : Macromedia Dreamweaver 8*. Jakarta: penerbit Andi

Membuat Efek Spesial dengan Dreamweaver 4, Gregorius Agung, Elex Media Komputindo, 2002

Mutmainah, Siti. 2005. *Proyek Desain Web Berbasis Grafis dengan Dreamweaver dan Firefox.* Yogyakarta: Andi.

Paul Timmers, (2000), *Electronic Commerce - strategies & models for business-to-business trading*, pp.31, John Wiley & Sons, Ltd

Pramono, Andi, Syafii, M. 2005. *Kolaborasi Flash, Dreamweaver, dan PHP untuk Aplikasi Web Site*.Yogyakarta: Andi.

Prihatna, Henky, *Kiat Praktis Menjadi Webmaster Profesional,* PT Elex Media Komputindo, 2005, Jakarta.

Rainer, jr., R kelly. *Introduction to Information Systems: Supporting and Transforming Business.* Hoboken, New Jersey: John Willey & Sons,2007

Seybold, Pat (2001), *Customers.com*, Crown Business Books (Random House)

Sugiri, S. H. 2008. Pengelolaan Database MySQL dengan PHPMyAdmin. Yogyakarta: Graha Ilmu

Sulhan. M. 2007. *Pengembangan Aplikasi Berbasis Web dengan PHP*. Yogyakarta

Syafii, M. Panduan Membuat Aplikasi Database Dengan PHP MySQL. AndiOffset. Yogyakarta.2005

Tunggala, W. 2003. *Tips dan Trik Membuat Website Dengan PHP*. www.ilmukomputer.com., Diakses tanggal 25 Januari 2006.

Wahyono, Teguh. 2005. *PHP Triad Fundamental [Memahami Pemrograman Web dengan PHP dan MySQL dalam 24 jam]*. Yogyakarta : Gava Media.

Wahyono, Teguh. 2005. *PHP Triad Fundamental [Memahami Pemrograman Web dengan PHP dan MySQL dalam 24 jam]*. Yogyakarta: Gava Media.

Wiwit Siswoutomo, (2005), *PHP Enterprise Kiat Jitu Membangun Web Skala Besar*, PT. Elex Media Komputindo, Jakarta.

SITUS WEB:

http://www.cybervillages.blogspot.com/2008/01/sekarang-eranya-web-20.html.

http://tiangazerockz.blogspot.com/2013/01/ukm-dan-masalah-pemasaran.html

https://id.wikipedia.org/wiki/Ray_Tomlinson

https://id.wikipedia.org/AdaLovelace/bahasa_Indonesia

http://iputan6.com /aribicara.com

https://id.wikipedia.org/wiki/kata_sandi

https://id.wikipedia.org/wiki/Tim_Berners-Lee

http://www.ibiblio.org/pioneers/lee.html)

http://nella13.blogspot.com/2012/09/kasus-keamanan-komputer-3-kasus.html

https://id.wikipedia.org/wiki/Rasmus_Lerdorf

http://suarapengusaha.com/2013/10/02/pengusaha-ukm-perlu-bangun-jaringan-di-malaysia

http://www.ib-article.com/2012/09/dr-fujio-masuoka-biography-inventors.html

https://www.penemu.co

http://www.biografiku.com/; https://www.wikidata.org/wiki/Q1297016;

http://nationalgeographic.co.id/berita/2014/10/penemu-hard-disk-dulunya-seorang-guru

http://microentrepreneurshipcommunity.blogspot.com /2012 /04/ membangun jaringan-usaha-bagi-usaha.html

http://www.biografiku.com/2012/07/biografi-marissa-mayer-ceo-yahoo.html

https://id.wikipedia.org/wiki/Ray_Tomlinson

http://www.ib-article.com/2012/08/biography-of-andy-rubin-inventor-of.html

https://id.wikipedia.org/wiki/Dennis_Ritchie

http://blog.hostdime.com.br/wp-content/uploads/2011/10/dennis_ritchie-hostdime.png

http://www.andrewdunnphoto.com

https://id.wikipedia.org/wiki/Charles_Babbage

https://id.wikipedia.org/wiki/Charles_Babbag; http://biokristi.sabda.org/charles_babbage_1791_1871

http://seputar-indonesia.com

https://id.wikipedia.org/wiki/Grace_Hopper

https://id.wikipedia.org/wiki/Berkas:Commodore_Grace_M._Hopper

TENTANG PENULIS

Made Kamisutara, ST., M.Kom, lahir di Singaraja, 06 Februari 1975 penulis yang telah malang melintang di dunia pendidikan dan penelitian ini menyelesaikan studi S1 Teknik Informatika pada tahun 1999 di ITATS Surabaya, dan Alumnus S2 Manajemen ITS Surabaya pada tahun 2007, selain mengampu mata kuliah E-Commerce, Konsep Sistem Informasi, dan Kewirausahaan di Univesitas Narotama Surabaya penulis juga menjabat sebagai Kaprodi Teknik Informatika beliau juga aktif dalam berbagai bidang penelitian dan pengabdian pada masyarakat utamanya berfokus pada UMKM.

Di sela-sela kesibukannya, penulis juga menuangkan gagasannya dalam berbagai karya ilmiah dan jurnal, diantaranya dalam jurnal NEUTRON dan SINATEK juga menerbitkan kepiawaiannya dalam buku *Penerapan Program Web Untuk Memasarkan Produk UKM* (2013) dan *Micro Controller: Wireless Smart Switching* (2017) yang diterbitkan oleh Narotama Press Surabaya.

Gatut Purwantoro, S.Kom., M.M, lahir di Surabaya, 01 Desember 1972. Saat ini tercatat sebagai staff desain grafis di Narotama University Press, penulis mengawali karirnya pada suatu perusahaan penerbitan di Surabaya, berkutat dalam kesibukan dan berbaur dengan banyak penulis membuatnya terdorong untuk menuangkan ide dan kreatifitasnya dalam karya tulis. Penulis yang sempat mengenyam kuliah di Arsitektur ITATS tahun 1991 namun tidak diselesaikannya, Alumnus Teknik Informatika dan Komputer UNITOMO tahun 2003, serta menyelesaikan studi S2 Magister Manajemen di Universitas Narotama Surabaya.

Mengawali kiprah menulisnya dalam penulisan buku-buku umum diantaranya Tema Pokok Sumber Manajemen Sumber Daya Manusia (2001) Insan Cendekia; Tumbuhan Berkhasiat Nusantara (2009) Pustaka Tanah Air; RPUL Rangkuman Pengetahuan Umum Lengkap GLOBAL Seri: Nasional dan Internasional (2010) Widyatamma, Jakarta; Pedoman Umum Ejaan Bahasa Indonesia Yang Disempurnakan dan Pedoman Umum Pembentukan Istilah (2012) Bintang Surabaya; Daftar Logaritma Dalam Empat Desimal (2012) Bintang Surabaya.

GLOSSARIUM

***Ad Units*,** Yang dimaksud dengan Ad Units adalah iklan *AdSense* itu sendiri. *Ad Units* terdiri dari beberapa jenis dan ukuran. Yang paling umum adalah jenis iklan teks. Pada saat pengunjung mengklik unit iklan ini, maka (jika sah) pemasang iklan akan mendapatkan pemasukan sesuai dengan nilai CPC-nya.

AdSense for Content, adalah iklan *AdSense* yang dipasang di dalam suatu halaman. Iklan-iklan yang muncul adalah iklan-iklan yang berhubungan dengan isi halaman tersebut. Atau istilahnya menggunakan konsep kontekstual. *Ad Units* dan *Link Units* adalah yang termasuk dalam *AdSense for Content* ini.

AJAX *[Asynchronous JavaScript dan XML]* - Menggunakan perangkat tambahan dalam JavaScript yang memungkinkan halaman Web menjadi lebih interaktif dan berperilaku seperti aplikasi lokal, yang juga dikenal sebagai aplikasi "klien kaya".AJAX memungkinkan halaman Web untuk mengambil sejumlah kecil data dari server tanpa reload seluruh halaman.

Alamat IP alamat IP adalah alamat komputer dan alamat interface router dalam lingkungan jaringan berprotokol TCP/IP. Alamat ini harus *unique.* Alamat IP berfungsi sebagai ID suatu komputer/ *router* dalam sebuah alamat *network.* Alamat *network* adalah alamat dimana sejumlah alamat IP berada pada satu segmen jaringan. (menunjukan deretan angka biner antar 32-bit sampai 128-bit yang dipakai sebagai alamat identifikasi untuk tiap komputer *host* dalam jaringan Internet. Panjang dari angka ini adalah 32-bit (untuk IPv4 atau IP versi 4), dan 128-bit (untuk IPv6 atau IP versi 6))

***Alternate Ads*,** Pada *AdSense for Content,* iklan tidak selalu muncul. Sebabnya antara lain bisa karena memang stok iklan yang berhubungan dengan isi situs sudah habis atau Google tidak dapat memperkirakan apa isi situs itu sebenarnya. Jika ini terjadi, secara default, yang ditampilkan adalah iklan layanan masyarakat atau sering dikenal dengan istilah PSA (*Public Service Ads*). Karena bertipe donasi, maka jika diklik, iklan ini

tidak menghasilkan apa-apa bagi publisher. Untuk mengatasinya, Google memperbolehkan kita untuk memasang Alternate Ads atau iklan alternatif. Jika Ad Units yang dibuat telah diatur dengan menggunakan *Alternate Ads,* maka apabila *Ad Units* tersebut tidak dapat tampil, yang muncul adalah iklan alternatif yang telah diatur sebelumnya.

Apache, *Server HTTP Apache* atau Server *Web/WWW Apache* adalah server web yang dapat dijalankan di banyak sistem operasi (Unix, BSD, Linux, Microsoft Windows dan Novell Netware serta *platform* lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini mengunakan HTTP. Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigur, autentikasi berbasis basis dataantarmuka pengguna berbasis grafik (GUI) yang memungkinkan penanganan server menjadi mudah. dan lain-lain. Apache juga didukung oleh sejumlah. Apache merupakan perangkat lunak sumber terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan Apache Software Foundation.

ASP [Active Server Pages] - Microsoft teknologi serupa dengan CGI yang digunakan untuk membuat konten dinamis untuk halaman web. Halaman menggunakan ASP diciptakan dengan skrip pemrograman (misalnya, JavaScript) dan terintegrasi dengan HTML halaman. Ini adalah bahasa scripting server-side dan banyak digunakan pada platform Windows.

ASP.NET Juga dikenal sebagai ASP +, adalah versi dari ASP untuk platform. NET. Mendukung program dieksekusi dikompilasi dari C #, C + + dan bahasa lainnya, yang tidak kompatibel dengan kode ASP biasa. Halaman ASP.NET selalu disusun bukan diartikan sebagai adalah halaman ASP.

Bandwidth Bandwidth adalah jumlah data yang dapat ditransfer melalui jaringan dalam jumlah waktu yang tetap. Di Internet, biasanya dinyatakan dalam bit per detik (bps). Sebuah server hosting akan mengalokasikan situs Anda jumlah yang tetap penggunaan bandwidth dalam jangka waktu yang teratur.

Channels, adalah semacam label yang dapat diberikan pada Ad Units, Link Units, AdSense for Search, dan Referrals. Satu unit iklan dapat memiliki lebih dari satu label, dan sebaliknya, satu label dapat digunakan untuk lebih dari satu unit iklan. Di halaman laporan Google AdSense, hasil laporan akan dikelompokkan berdasarkan Channels, sehingga

penggunaan Channels akan sangat memudahkan publisher untuk menganalisa performa AdSense mereka. Umumnya, publisher akan memberikan nama Channels yang sama pada unit-unit iklan yang ada di satu situs. Jika ingin lebih detail, sah-sah saja untuk memberikan nama *Channels* yang berbeda pada setiap unit iklan di masing-masing situs. Yang perlu diingat, maksimal jumlah *Channels* yang diperbolehkan saat ini adalah 200 kanal.

Chipertext, merupakan bentuk terenkripsi dari data

Clicks, adalah jumlah klik pada Ad Units milik publisher. Dalam halaman laporan *AdSense, publisher* dapat melihat total klik yang ia dapatkan, maupun berdasarkan Ad Units atau Channelnya.

CPC (Cost Per Click), adalah jumlah uang yang akan didapatkan oleh publisher apabila Ad Units tertentu diklik. Nilai CPC masing-masing Ad Units berbeda dan ditentukan oleh banyak faktor, termasuk performa dan kualitas situs milik publisher. Namun secara umum, nilai maksimal yang mungkin adalah 20% dari nilai tawaran dinamis yang ditawarkan oleh pemasang iklan.

CTR *(Clickthrough Rate),* adalah perbandingan dalam persen antara jumlah klik yang diterima suatu *Ad Units* dengan jumlah tampilan *Ad Units tersebut.* Misalnya, satu Ad Units yang ditampilkan 40 kali dan diklik 10 kali memiliki nilai CTR 25% (10:40).

DHTML *[Dynamic HTML]* - Kombinasi HTML tambahan, bahasa *scripting* dan antarmuka yang digunakan untuk memberikan animasi, interaksi dan dinamis memperbarui pada halaman Web. Dua unsur utama adalah bahasa *ECMAScript* dan model objek DOM. *ECMAScript* adalah turunan dari JavaScript, dan DOM adalah sebuah antarmuka yang menyajikan dokumen HTML untuk programmer sebagai model obyek untuk memudahkan dalam *update.*

Digital money adalah terminologi global untuk berbagai *e-cash* dan mekanisme pembayaran elektronik di Internet.

Disintermediation adalah proses untuk memotong jalur perantara. Kira-kira pada saat perusahaan yang berbasiskan web membypass kanal retail tradisional dan menjual secara langsung ke pelanggan/pembeli, maka perantara tradisional – seperti toko dan jasa mail order – akan kehilangan pekerjaan.

DNS *[Domain Name System (Layanan)]* - Sistem Internet/layanan yang menerjemahkan nama domain menjadi alamat IP. Nama domain adalah abjad sehingga mereka lebih mudah untuk

diingat. Internet bagaimanapun, adalah benar-benar berdasarkan alamat IP. Setiap kali Anda menggunakan nama domain, karena itu, layanan DNS harus menerjemahkan nama ke alamat IP yang sesuai.

eCPM (Effective CPM), adalah hasil pembagian antara jumlah pendapatan publisher dengan jumlah impresi halaman (per 1.000) yang ia dapatkan dari iklan-iklannya. Sebagai contoh, *publisher* yang menghasilkan USD 200 dari 50.000 impressi akan memiliki nilai CPM sebesar USD 4 (USD 200 dibagi 50).

Electronic Cash juga dikenal sebagai *e-cash,* istilah ini ditujukan untuk beberapa pola/metoda yang memungkinkan seseorang untuk membeli barang atau jasa dengan cara mengirimkan nomor dari satu komputer ke komputer yang lain. Nomor tersebut, seperti yang terdapat di mata uang, diisukan oleh sebuah bank dan merepresentasikan sejumlah uang betulan. Salah satu kelebihan yang dibawa oleh digital cash adalah sifatnya yang anonymous dan dapat di pakai ulang, seperti uang cash biasa. Hal ini merupakan perbedaan utama antara *e-cash* dengan transaksi kartu kredit melalui Internet.

Electronic Checks pada saat ini sedang di ujicoba oleh *CyberCash* http://www.cybercash.com/, sistem *check* elektronik seperti *PayNow* akan mengambil uang dari account *check* di bank pelanggan untuk membayar PAM atau telepon.

Electronic Wallet Pola pembayaran – seperti *CyberCash Internet Wallet* http://www.cybercash.com/, akan menyimpan nomor kartu kredit anda di harddisk anda dalam bentuk terenkripsi yang aman. Anda akan dapat melakukan pembelian-pembelian pada situs Web yang mendukung electronic wallet tersebut. Jika anda ingin membeli sesuatu pada toko yang mendukung *electronic wallet,* maka pada saat menekan tombol *Pay* maka proses pembayaran melalui kartu kredit akan dilakukan transaksinya secara aman oleh server perusahaan *electronic wallet.* Vendor browser pada saat ini telah berusaha untuk melakukan negosiasi untuk memasukan teknologi *e-wallet* tadi ke produk mereka.

Extranet adalah sebuah kelanjutan dari intranet perusahaan yang mengkaitkan jaringan internal satu perusahaan dengan jaringan internal supplier mereka maupun pelanggan mereka. Dengan cara itu sangat mungkin untuk mengembangkan aplikasi *e-commerce* yang memungkinkan menyambungkan semua aspek bisnis, dari proses pemesanan hingga pembayaran.

Frames Sebuah teknik HTML untuk menggabungkan dua atau lebih dokumen HTML yang terpisah dalam satu layar web browser tunggal. Sebuah situs web menggunakan *frame* sering menyebabkan masalah besar bagi mesin pencari, dan tidak akan di jelajahi dan diindeks dengan benar.

FTP *[File Transfer Protocol]* - Salah satu metode yang umum untuk mentransfer file melalui Internet. Sebuah metode khas yang digunakan untuk meng-*upload file* (halaman) ke *server hosting* untuk dilihat di Internet.

Hosting Biasanya mengacu pada sebuah komputer (atau jaringan server) yang menyimpan file-file dari sebuah situs web yang memiliki perangkat lunak web server yang berjalan diatasnya, terhubung ke Internet. Situs Anda kemudian dikatakan *Hosted.*

HTML *[HyperText Markup Language]* - HTML adalah bahasa *markup* dasar berasal dari *markup* umum *Standar Language* (SGML), menyediakan sarana untuk membuat dokumen *hypertext* sederhana, dimaksudkan untuk penerbitan di *World Wide Web.*

HTTP, adalah sebuah protokol meminta/menjawab antara *client* dan *server.* Sebuh *client* HTTP seperti web browser, biasanya memulai permintaan dengan membuat hubungan TCP/IP ke port tertentu di tuan rumah yang jauh (biasanya *port* 80). Sebuah server HTTP yang mendengarkan di *port* tersebut menunggu *client* mengirim kode permintaan *(request),* seperti *"GET / HTTP/1.13* (yang akan meminta halaman yang sudah ditentukan), diikuti dengan pesan MIME yang memiliki beberapa informasi kode kepala yang menjelaskan aspek dari permintaan tersebut, diikut dengan badan dari data tertentu. Beberapa kepala *(header)* juga bebas ditulis atau tidak, sementara lainnya (seperti tuan rumah) diperlukan oleh protokol HTTP/1.1. Begitu menerima kode permintaan (dan pesan, bila ada), server mengirim kembali kode jawaban, seperti "200 OK", dan sebuah pesan yang diminta, atau sebuah pesan error atau pesan lainnya.

Internet Sebuah jaringan global yang menghubungkan jutaan komputer. Setiap komputer internet, yang disebut *host*, bersifat independen. Internet tidak sama dengan *World Wide Web.* Internet dan Web adalah dua hal yang terkait tetapi terpisah.

IP *[Internet Protocol]* - Metode atau protokol dengan data yang dikirim dari satu komputer ke komputer lain di Internet. Setiap komputer (dikenal sebagai tuan rumah) di Internet memiliki

setidaknya satu alamat IP yang unik mengidentifikasi dari semua komputer lain di Internet.

JavaScript JavaScript adalah berbasis obyek, client-side scripting bahasa yang dikembangkan oleh Netscape. Melekat pada bagian kepala dokumen web, ini dapat menghasilkan interaktivitas ke halaman web dinamis.

Kriptoanalis, yaitu orang yang mempelajari enkripsi dan pesan terenkripsi dengan tujuan menemukan makna yang tersembunyi

Kriptografer, memiliki tujuan sama seperti kriptoanalis, bedanya adalah kriptografer bekerja untuk tujuan legal sedang kriptoanalis cenderung secara illegal.

Kriptografi, yaitu cara penggunaan enkripsi untuk menyembunyikan teks.

Link Units, Link Units hampir sama dengan Ad Units, hanya saja formatnya mirip dengan format menu yang biasa kita temui di situs-situs web. Yang membedakan Link Units dengan Ad Units adalah pada saat pengunjung meng-klik iklan ini, maka ia akan diarahkan pada halaman hasil pencarian di search engine Google. Publisher baru akan mendapatkan pemasukan apabila pengunjung mengklik salah satu Ad Unit yang ada di halaman tersebut. Pada prakteknya, Link Units terbukti menghasilkan pemasukan lebih banyak dibandingkan Ad Units biasa.

Meta Tag Sebuah tag HTML khusus yang menyediakan informasi tentang halaman Web. Tidak seperti tag HTML biasa, tag meta tidak mempengaruhi bagaimana halaman ditampilkan.

Micropaymet transaksi dalam jumlah kecil antara beberapa ratus rupiah hingga puluhan ribu rupiah, misalnya untuk mengambil / mengakses grafik, game maupun informasi. Pay-as-you-go micropayment seharusnya akan membuat revolusi di dunia e-commerce.

Contohnya ESPN SportsZone http://espn.sportszone.com/ menggunakan CyberCoin untuk membayar US$1 untuk mengaskses situs mereka selama satu hari – tanpa perlu membayar penuh langganan bulanan. Kenyataan di lapangan sebagian besar pelanggan yang potensial tidak terlalu bersedia untuk bermain-main dengan micropayment.

MySQL, adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat

closed source atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai database server, MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase

Nama *Domain* Sebuah nama unik yang mengidentifikasi satu atau lebih alamat IP. Nama Domain digunakan di URL untuk mengidentifikasi situs Web tertentu.Setiap situs web terletak berdasarkan alamat IP unik.

Page Impressions, adalah jumlah yang menunjukkan berapa kali halaman yang mengandung Ad Units dibuka oleh pengunjung. Nilainya tidak terpengaruh oleh kuantitas Ad Units yang ada di dalam halaman yang bersangkutan.

PHP *[PHP Hypertext Preprocessor]* - Sebuah server-side, HTML embedded scripting bahasa yang digunakan untuk membuat halaman Web dinamis.Dirancang untuk platform Windows dan jenis Unix.

PHP, adalah kependekan dari PHP: *Hypertext Preprocessor,* bahasa interpreter yang mempunyai kemiripan dengan C dan Perl.PHP dapat digunakan bersama dengan HTML sehingga memudahkan dalam membuat aplikasi web dengan cepat. Dapat digunakan untuk menciptakan dynamic website baik itu yang memerlukan penggunaan database ataupun tidak.

Plaintext, merupakan bentuk awal dari pesan

Publisher, adalah orang atau pemilik situs yang sudah bergabung dan memasang iklan AdSense di situs mereka.

RSS *[Really Simple Syndication]* - RSS Feed adalah sebuah format yang cukup baru dari XML yang dimaksudkan untuk berbagi informasi dalam format kental (seperti judul, deskripsi dan link ke artikel berita). Mereka menyediakan pengguna dengan cara mudah

untuk menerima *up-to-date* berita dan pengumuman tanpa harus memeriksa kembali ke situs Anda setiap terlalu sering.

Script *script* adalah suatu daftar eksekusi dari perintah yang dibuat oleh bahasa *scripting*. *Script* yang dijalankan di web server (misalnya; Perl, PHP) dikatakan sebagai *server-side script*. *Script* yang dijalankan pada PC rumah Anda sendiri (misalnya; JavaScript) yang dikatakan *script* sisi klien. *Script* dapat menanamkan dalam HTML untuk menghasilkan halaman web dengan tindakan dinamis.

Scripting **Bahasa** Sebuah bahasa *scripting* adalah bahasa pemrograman sederhana yang digunakan untuk menulis daftar eksekusi dari perintah, yang disebut naskah. JavaScript, Perl, VBScript yang bahasa *scripting,* bukan untuk keperluan umum bahasa pemrograman.

Search Engine Sebuah server (komputer) atau yang biasa kumpulan server yang didedikasikan untuk mengindeks halaman web internet, menyimpan hasilnya dalam database raksasa dan kembali daftar halaman yang cocok dengan *query* dicari tertentu dari dalam *database*-nya. Indeks-indeks biasanya dan secara otomatis dihasilkan dengan menggunakan laba-laba.

Server Sebuah komputer, program atau proses yang merespon permintaan informasi dari seorang pengguna. Di internet, semua halaman web berada pada *server* (komputer).

Spider Sebuah robot perangkat lunak otomatis yang terus merangkak *hyperlink* dan halaman di Internet dan mengumpulkan data yang dikembalikan ke *database* untuk mengindeks. Ini adalah bagaimana *Search Engine* berfungsi.Proses merangkak web, menyimpan URL ‘dan kata kunci pengindeksan, *link* dan teks, adalah tindakan *spidering*.

SSI ***[Server-Side Includes]*** - Memberitahukan server untuk memasukkan informasi (sumber dari file terpisah) dalam dokumen sebelum mengirimnya ke browser. Sebuah metode yang sangat efektif menghasilkan informasi yang sama lebih banyak halaman sebagai satu file bisa diubah untuk menghasilkan perubahan selama banyak halaman yang mencakup file SSI.

Sub-Domain (Nama) Sebuah sub-domain adalah domain yang merupakan bagian dari nama domain yang lebih besar. DNS hirarki terdiri dari domain tingkat akar di bagian atas, di bawah yang merupakan domain tingkat atas, diikuti oleh kedua-level domain dan akhirnya sub-domain.

Tag Sebuah tag HTML adalah perintah format ditulis ke dalam dokumen yang menentukan bagaimana harus diformat. Sebuah web browser menafsirkan tag ini dan output perintah yang diinginkan (tindakan).

Tautan *[Hyperlink]* Sebuah elemen dalam suatu dokumen elektronik yang menghubungkan ke tempat lain dalam dokumen yang sama atau ke dokumen sama sekali berbeda. Biasanya, Anda klik pada *hyperlink* untuk mengikuti link.*Hyperlinks* adalah bahan yang paling penting dari semua sistem hypertext, termasuk *World Wide Web.*

Template *template* HTML adalah halaman HTML kerangka dengan konten utama ditinggalkan. *Template* memberikan solusi efektif dalam menciptakan banyak halaman dengan tampilan yang sama atau struktur navigasi tapi isi yang berbeda.

URI *[Uniform Resource Identifier]* - Istilah generik untuk semua jenis nama dan alamat yang mengacu pada objek pada *World Wide Web.* URL adalah salah satu jenis URI.

URL *[Uniform Resource Locator]* - Setiap halaman yang terpisah dapat diakses di Web memiliki alamat unik yang dapat diidentifikasi oleh oleh URL itu. Bagian pertama dari alamat (misalnya; http atau ftp dan lain-lain) menunjukkan protokol yang digunakan, dan bagian kedua menentukan alamat IP atau nama domain dimana sumber daya tersebut berada.

URL, *Uniform Resource Locator* (URL) adalah rangkaian karakter menurut suatu format standar tertentu, yang digunakan untuk menunjukkan alamat suatu sumber – seperti dokumen dan gambar – di Internet.

Mengacu pada tingkat atau derajat keramahan suatu halaman beroperasi bagi pengguna.

Validasi Validasi adalah sebuah cara untuk memastikan bahwa (HTML) kode Anda kompatibel dengan spesifikasi HTML saat ini.

W3C *[World Wide Web Consortium]* - Didirikan pada bulan Oktober 1994 untuk memimpin *World Wide Web* secara maksimal dengan mengembangkan protokol umum yang mempromosikan evolusi dan memastikan interoperabilitas.

Web Server, adalah *software* yang menjadi tulang belakang dari world wide web (www). *Web server* menunggu permintaan HTTP atau HTTPS dari *client* yang menggunakan browser seperti Netscape Navigator, Internet Explorer, Modzilla, dan program browser lainnya. Jika ada permintaan dari browser, maka web server akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke browser. Data ini mempunyai format yang standar, disebut dengan format SGML (standar general markup language). Data yang berupa format ini kemudian akan ditampilkan oleh browser sesuai dengan kemampuan browser tersebut.

XHTML *[Extensible HyperText Markup Language]* - Sebuah bahasa *markup* untuk halaman Web dari W3C. XHTML menggabungkan HTML dan XML menjadi format tunggal (HTML 4.0 dan XML 1.0). Seperti XML, XHTML dapat diperpanjang dengan tag eksklusif. Juga seperti XML, XHTML harus dikodekan lebih ketat daripada HTML. Selama bertahun-tahun, *coders* HTML telah menjadi ceroboh, karena perangkat lunak browser Web pada awalnya ditulis untuk mentolerir banyak variasi dalam HTML *coding.* Dengan XHTML, *coders* harus sesuai dengan aturan.

XML *[Extensible Markup Language]* - Standar terbuka untuk mendeskripsikan data dari W3C. Hal ini digunakan untuk mendefinisikan elemen data pada halaman web dan bisnis-ke-dokumen bisnis. XML menggunakan struktur tag yang sama sebagai HTML, namun, sedangkan HTML mendefinisikan bagaimana elemen ditampilkan, XML mendefinisikan apa elemen-elemen mengandung. Sedangkan HTML menggunakan tag yang telah ditetapkan, XML memungkinkan tag untuk didefinisikan oleh pengembang halaman. Dengan demikian, hampir item data apapun, seperti "produk", "perwakilan penjualan" dan "jumlah yang jatuh tempo," dapat diidentifikasi, yang memungkinkan halaman Web untuk fungsi seperti catatan *database.* Dengan menyediakan metode umum untuk mengidentifikasi data, XML mendukung bisnis-ke-bisnis transaksi dan telah menjadi "format" untuk pertukaran data elektronik dan layanan Web.