SKRIPSI

PENGEMBANGAN PROTOTIPE *QUICK RESPONSE CODE* (*QR CODE*) SEBAGAI AUTENTIKASI KEAMANAN LOGIN SISTEM DENGAN MEMANFAATKAN TEKNOLOGI ANDROID

DISUSUN OLEH :

FAUSTO ERNESTO KARUNA
NIM : 04109023

PROGRAM STUDI SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS NAROTAMA
SURABAYA

2016

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa dalam Tugas Akhir ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat Karya/Pendapat yang pernah ditulis oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam Daftar Acuan/Daftar Pustaka.

Apabila ditemukan suatu Jiplakan/Plagiat maka saya bersedia menerima akibat berupa sanksi Akademis dan sanksi lain yang diberikan oleh yang berwenang sesuai ketentuan peraturan dan perundang-undangan yang berlaku.

Surabaya, 9 Februari 2016

Yang membuat pernyataan

Nama : Fausto Ernesto Karuna

NIM : 04109023

HALAMAN MOTO DAN PERSEMBAHAN

MOTTO

"*Not all of us can do great things, But we can do small things with great love*"

(Mother Teresa)

PERSEMBAHAN

Penulis persembahkan secara khusus kepada pihak-pihak yang penulis hormati dan kasihi :

1. Bapak Cahyo Darujati, ST., MT. selaku Dekan Fakultas Ilmu Komputer Universitas Narotama Surabaya.
2. Bapak Slamet Winardi, ST., MT. selaku Ketua Program Studi Sistem Komputer Universitas Narotama Surabaya.
3. Bapak M. Noor Al Azam, S.Kom., M.MT. selaku dosen pembimbing yang telah membimbing saya dalam menyelesaikan tugas akhir ini.
4. Orang tua yang selalu memberikan motivasi, dukungan, dan doanya.
5. Segenap dosen dan karyawan Universitas Narotama Surabaya atas bimbingan dan kontribusinya.
6. Terima Kasih Untuk Fika, Kang Erwin, Om Bas, Pungky dan 15F Crew yang selalu siap sedia memberikan bantuannya sampai skripsi ini terselesaikan.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa, yang karena karuniaNya Laporan Skripsi dengan judul : **"Pengembangan *Prototipe Quick Response Code* (*QR CODE*) Sebagai Autentikasi Keamanan Login Sistem Dengan Memanfaatkan Teknologi Android"** dapat penulis selesaikan dengan baik.

Penulis menyadari dalam penyusunan Laporan Skripsi ini masih jauh dari kesempurnaan karena terbatasnya pengetahuan, pengalaman, dan waktu yang ada. Untuk itu segala masukan baik kritik maupun saran yang bersifat membangun sangat diharapkan demi kesempurnaan Laporan Skripsi ini, sehingga nantinya dapat bermanfaat bagi penulis dan pembaca pada umumnya.

Akhir kata, penulis sampaikan terima kasih kepada semua pihak yang berperan serta dalam penyusunan Laporan Skripsi ini dari awal sampai akhir.

Surabaya, 9 Februari 2016

Penulis,

Fausto Karuna

PENGEMBANGAN PROTOTIPE *QUICK RESPONSE CODE* (*QR CODE*) SEBAGAI AUTENTIKASI KEAMANAN LOGIN SISTEM DENGAN MEMANFAATKAN TEKNOLOGI ANDROID

Oleh : Fausto Ernesto Karuna

Pembimbing : M. Noor Al Azam, S.Kom., M.MT.

ABSTRAK

Seiring perkembangan teknologi informasi yang begitu pesat, dan pertumbuhan infrastruktur dunia maya. Jumlah situs jejaring sosial, blog, dan forum juga semakin bertambah. Semakin banyak situs yang ada, mendorong pengguna internet untuk memiliki akun di berbagai situs tersebut. Dengan banyaknya akun, maka semakin lama waktu yang dibutuhkan untuk melakukan *login*. Semakin sulit pula untuk mengingat *username* dan *password* milik *user*. Untuk itu penulis berinovasi untuk membuat sebuah aplikasi dengan memanfaatkan teknologi android yang dikombinasikan dengan teknologi *QR code* yang dinilai praktis dan dapat memberi solusi pada masalah tersebut. *User* tidak perlu menghafal *username* dan *password* serta menginputkan melalui *keyboard*. *Username* serta *password* milik *user* disimpan dalam suatu *database*, saat *user* ingin melakukan *login*, maka aplikasi akan memberikan *QR code* yang telah berisi *username* dan *password* milik user, untuk melakukan proses login pada layanan situs tersebut dengan cara melakukan scanning *QR Code* pada *webcam*. Sehingga *user* mendapatkan respon yang cepat, praktis, sederhana, dan aman pada akses *website* yang dituju.

Kata Kunci : Android, QR Code, Website, Login

PENGEMBANGAN PROTOTIPE *QUICK RESPONSE CODE* (*QR CODE*) SEBAGAI AUTENTIKASI KEAMANAN LOGIN SISTEM DENGAN MEMANFAATKAN TEKNOLOGI ANDROID

Oleh : Fausto Ernesto Karuna

Pembimbing : M. Noor Al Azam, S.Kom., M.MT.

ABSTRACT

Along with the development of information technology so rapidly, and the growth of cyberspace infrastructure. Number of social networking sites, blogs, and forums are also increasing. The more sites there, encouraging Internet users to have accounts on these sites. By many accounts, the longer the time it takes to login. More difficult it is to remember the user's username and password. To the authors innovate to create an application by using android technology combined with QR code technology is considered practical and can provide solutions to these problems. Users do not need to memorize the username and password as well as input via the keyboard. The user's username and password are stored in a database, when the user wants to log in, then the application will provide a QR code that already contain the user's username and password, to perform the login process on the services these sites by scanning a QR Code on webcam. So that users get a quick response, practical, simple, and secure the access to the targeted site.

Keywords : Android, QR Code, Website, Login.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam dunia komunikasi data global dan perkembangan teknologi informasi yang senantiasa berubah serta cepatnya perkembangan *software*, keamanan merupakan suatu hal yang sangat penting. Baik itu keamanan fisik, keamanan data maupun keamanan aplikasi. Salah satu metode pengamanan sistem informasi yang umum diketahui oleh banyak orang adalah *username* dan *password*. Tanpa disadari *username* dan *password* mempunyai peranan penting dalam mengamankan informasi-informasi yang bersifat pribadi (*confidential*). Dimulai dari hal yang terlihat sangat sederhana, tetapi pada prakteknya cukup banyak terjadi kebobolan dari sisi halaman login yang menggunakan *username* dan *password* yang diakibatkan oleh kecerobohan pengguna maupun kelemahan pada sistem.

Username dan *password* berfungsi untuk perlindungan (*protection*) dan bersifat rahasia. Dengan demikian, hanya orang yang tahu *username* dan *password* saja yang bisa membuka data ataupun mengakses layanan. *Password* telah diterapkan untuk autentikasi berbagai layanan. Dalam transaksi perbankan, kita semua mengenal adanya ATM (Anjungan Tunai Mandiri). ATM mengharuskan pemilik kartu ATM untuk menghafal dan

merahasiakan *password* yang biasanya hanya berupa nomor. *Password* tersebut sering disebut dengan *Personal Identification Number* (PIN). Di internet, banyak *website* yang mengharuskan pengguna memasukkan *username* (nama pengguna) dan *password* untuk mengakses layanan. Mulai dari *e-mail*, forum, data, *web hosting*, situs berita, dan masih banyak lagi. *Password* yang berhubungan dengan komputer, biasanya dapat menggunakan seluruh karakter standar, yaitu kombinasi huruf, angka, dan simbol. Berkaitan dengan penggunaan komputer, *password* dapat digunakan untuk memproteksi data dan sistem. Data yang diproteksi bisa berupa *file*, sedangkan sistem yang diproteksi bisa berupa program aplikasi, sistem operasi, dan *BIOS* komputer.

Meskipun terlihat sederhana dan praktis *username* dan *password* ternyata mempunyai kelemahan dan kelebihan. Kelebihannya, Pengguna hanya perlu menghafal *username* dan *password* tanpa harus menggunakan berbagai alat tambahan. Selain kelebihannya yang bersifat praktis, penggunaan *password* juga mempunyai kelemahan, yang diakibatkan oleh beberapa faktor seperti kecerobohan pengguna dalam mengatur *password*-nya. mulai dari memilih *password* yang gampang ditebak, *password* tidak dijaga dengan baik sehingga bocor, pengguna lupa *password*-nya sendiri, atau *password* digunakan secara sembarangan. Contohnya setiap kali *password* diketikkan, maka pada saat itulah bisa terjadi insiden pencurian *password*. Insiden kecil seperti gerakan tangan yang diamati orang lain saat mengetik *password*, dapat menyebabkan *password* pengguna dijebol.

Berdasarkan penjelasan diatas penulis mempunyai inovasi untuk mengembangkan sebuah sistem autentikasi yang lebih aman serta memudahkan dan memanjakan pengguna dengan memanfaatkan *QR-code* yang berorientasi pada pengguna telepon selular. Tentunya banyak kemudahan yang didapat dengan menggunakan aplikasi ini. Diantaranya pengguna tidak perlu repot untuk mengetik, menghafal *username* dan *password* milik pengguna pada halaman login website, *QR-code* yang mudah dibaca oleh pemindai *QR* sehingga pengguna mendapatkan respon yang cepat, praktis, dan aman pada akses *website* yang dituju.

1.2 Rumusan Masalah

1. Bagaimana mengembangkan sistem autentikasi login yang cepat, praktis dan aman menggunakan *QR code*?
2. Bagaimana mengembangkan *QR code* sebagai inovasi dalam melakukan login dengan memanfaatkan android sebagai perangkat tambahan?

1.3 Batasan Masalah

1. Pengembangan sistem ini hanya untuk halaman login *website*.
2. Sistem autentikasi login ini dikembangkan pada sistem operasi android versi 4.1 keatas.

1.4 Tujuan Penelitian

1. Tujuan pengembangan prototipe *QR code* ini untuk memudahkan pengguna melakukan login tanpa harus mengingat dan mengetikkan *username* dan *password*.

2. Sistem yang lebih aman karena *QR code* yang tidak mudah dihafal dan ditebak, serta *QR code* yang berubah-ubah setiap waktu.

1.5 Manfaat Penelitian

Secara umum manfaat penelitian dari sistem yang akan dikembangkan adalah :

1. Memberikan kemudahan dan memanjakan pengguna untuk dapat mengakses website dengan cepat, praktis, dan aman. Tanpa harus perlu mengetik, mengingat *username* dan *password* yang seringkali dilupakan oleh pengguna.

2. Meminimalisir tingkat pencurian *username* dan *password* milik pengguna dikarenakan *QR code* yang berubah-ubah setiap waktu.

1.6 Sistematika Penulisan

1. Bab I : Pendahuluan

   Bab ini berisi tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan dilaksanakannya penelitian, manfaat penelitian, dan sistematika penulisan laporan.

2. Bab II : Tinjauan Pustaka

   Terdiri dari landasan teori yang ada hubungannya dengan penelitian yang akan dilakukan dan uraian sistematis tentang hasil-hasil penelitian yang didapat oleh peneliti terdahulu.

3. Bab III : Metodologi Penelitian

   Bab ini berisi tentang langkah-langkah pengembangan prototipe *QR-code* sebagai autentikasi keamanan login sistem dengan memanfaatkan teknologi android. Pengembangan sistem ini terdiri dari deskripsi umum sistem, alur kerja sistem, perancangan database, dan perancangan antar muka sistem.

4. Bab IV : Hasil dan Pembahasan

   Bab ini berisi tentang hasil dan pembahasan sistem yang didapat pada tahap implementasi serta pengujian sistem.

5. Bab V : Penutup

   Berisi kesimpulan yang diambil berkaitan dengan sistem yang dibuat dan saran untuk pengembangan sistem lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Sebagai referensi atau bahan pertimbangan dalam penelitian ini, maka penulis akan mencantumkan hasil penelitian terdahulu yang pernah ditulis oleh peneliti sebelumnya. Sebagai berikut :

Penelitian yang dilakukan oleh Masdito Bachtiar dan Ary Mazharuddin pada tahun 2012, dengan judul "Smart Login Pada Situs Web Menggunakan *QR Code*". Penelitian tersebut menjelaskan tentang autentikasi login pada website menggunakan *QR Code*. Terlihat dari judul penelitian sebelumnya yang hampir sama. Namun pada kenyataannya berbeda. Pada penelitian terdahulu, peneliti sebelumnya menggunakan media kartu sebagai hasil generate *QR code* yang merupakan username dan password dari pengguna. Dari segi keamanan, kartu yang digunakan untuk login nantinya masih dapat dicuri atau difoto oleh orang lain. Dan *QR code* yang digunakan tidak mempunyai batas waktu, apakah itu berlaku untuk satu kali login saja atau dapat digunakan untuk login selanjutnya.

Untuk itu penulis mengkaji ulang penelitian sebelumnya, dengan melakukan penelitian menggunakan *QR code* sebagai sarana autentikasi keamanan login pada website dengan memanfaatkan teknologi android. Dimana hasil generate *QR code* yang berupa password dan username tidak dicetak pada kartu namun terdapat langsung pada aplikasi perangkat seluler android milik

pengguna, dan *QR code* yang berubah-ubah setiap waktu, berlaku hanya untuk satu kali login.

2.2 Android

Android adalah sistem operasi perangkat mobile berbasis linux yang mencakup sistem operasi dan aplikasi. Seiring perkembangannya android berubah menjadi platform yang begitu cepat dalam melakukan inovasi. Hal ini tidak lepas dari pengembang utama dibelakangnya yaitu google. Sebagai generasi baru *platform mobile*, android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi dan memberikan kesempatan kepada pengembang secara leluasa untuk menciptakan aplikasi sesuai dengan yang diharapkan. Sederhananya android terdiri dari tiga kombinasi komponen yaitu :

a. Sistem operasi yang bersifat *open source* untuk perangkat mobile.

b. Sebuah *open source platform* untuk menciptakan aplikasi *mobile*.

c. Perangkat serta telepon seluler khususnya yang menjalankan sistem operasi android.

Android terdiri dari bagian-bagian yang saling membutuhkan dan memiliki ketergantungan antar bagian seperti (Meier, 2010, p4) :

1. Sebuah desain referensi hardware yang menggambarkan kemampuan yang dibutuhkan untuk sebuah perangkat mobile untuk mendukung software stack.

2. Sebuah kernel sistem operasi linux yang menyediakan antarmuka tingkat rendah dengan hardware manajemen memori, dan kontrol proses, semua dioptimalkan perangkat mobile.

3. Open source libraries untuk pengembangan aplikasi termasuk SQLite, WebKit, OpenGL, dan media manager.

4. Sebuah run time yang digunakan untuk mengeksekusi dan host aplikasi android, termasuk dalvik virtual machine dan core libraries yang menyediakan fungsi spesifik dari android. Run time ini dirancang kecil dan efisien untuk digunakan perangkat mobile.

5. sebuah framework aplikasi yang menghadapkan layanan sistem ke lapisan aplikasi, termasuk window manager, dan location manager, penyedia konten, telephony, dan sensor.

6. sebuah framework antarmuka yang digunakan sebagai host dan menjalankan aplikasi.

7. aplikasi yang sudah terpasang sebagai bagian dari stack.

8. sebuah software development kit yang digunakan untuk membuat aplikasi, termasuk tools, plugins, dan dokumentasi.

Sistem operasi android mirip sebuah kue yang terdiri dari berbagai lapisan. Setiap lapisan memiliki karakteristik dan fungsinya masing-masing. Setiap layer tidak seutuhnya terpisahkan tetapi saling bergantungan. (Gargenta, 2011, p7).

Android software stack terdiri dari unsur-unsur yang dimana sebuah kernel linux dan koleksi libraries C/C++ mengarah pada framework aplikasi yang menyediakan layanan, pengelolaan, run time, dan aplikasi (Meier, 2010, p13).

Kernel linux menjadi layanan inti termasuk penyedia hardware drivers, manajemen proses dan memori, keamanan jaringan, dan manajemen tenaga. Semuanya itu ditangani oleh kernel linux. Kernel tersebut juga menyediakan sebuah layer abstraksi di antara hardware dan sisa dari stack (Meier, 2010, p13).

Libraries berjalan di kernel teratas terdiri dari berbagai macam C/C++ coe libraries seperti libc dan SSL, serta (Gargenta, 2011, p3) :

a. Sebuah media library untuk pemutaran media suara dan video.

b. sebuah surface manager yang menyediakan manajemen tampilan.

c. Graphic libraries yang mencakup SGL dan OpenGL untuk grafis 2D dan 3D.

d. SQLite untuk dukungan database native.

e. SSL dan Webkit untuk integrasi web browser dan keamanan internet.

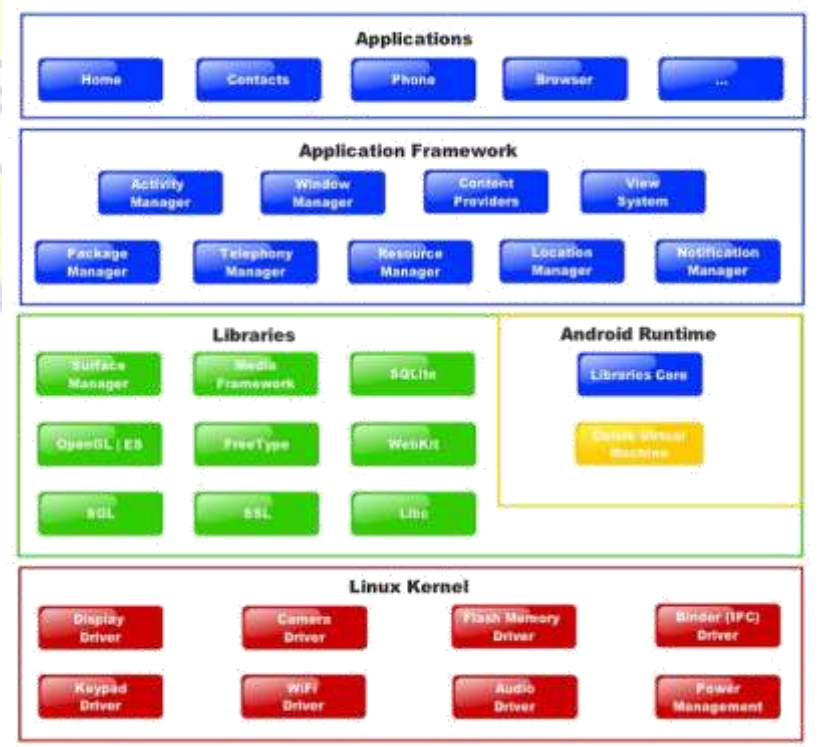
Android run time membuat perangkat android lebih dari sebuah perangkat mobile dengan implementasi linux. Terdapat core libraries dan Dalvik virtual machine di dalamnya. Run time pada android adalah mesin yang memberikan kekuatan pada aplikasi, bersama dengan libraries, membentuk dasar framework aplikasi. Core libraries menyediakan sebagian besar fungsi yang tersedia di dalamnya. Sedangkan dalvik virtual machine adalah virtual machine yang telah

dioptimalkan untuk memastikan bahwa sebuah perangkat dapat menjalankan beberapa hal secara efisien (Meier, 2010, p13).

Application framework menyediakan kelas-kelas yang digunakan untuk membuat aplikasi android. Selain itu juga menyediakan abstraksi umum untuk akses hardware dan mengatur user interface dan application resources (Meier, 2010, p14).

Application layer berjalan dalam run time android, menggunakan kelas-kelas dan layanan yang disediakan dari application framework (Meier, 2010, p14).

Arsitektur pada android mendorong konsep penggunaan kembali komponen, memungkinkan untuk mempublikasikan dan berbagi activities, layanan, dan data dengan aplikasi lainnya.



Gambar 2.1 *Stack Pada Android*

Berikut layanan-layanan aplikasi yang menjadi pilar arsitektur dari semua aplikasi android (Meier, 2010, p15) :

a. Activity Manager, digunakan untuk mengontrol daur hidup dari aktivitas, termasuk manajemen aktivitas stack.

b. Views, digunakan untuk membangun user interfaces untuk aktivitas.

c. Notification Manager, menyediakan mekanisme yang konsisten dan tidak menggangu untuk memberitahu user.

d. Content Providers, membiarkan aplikasi berbagi data.

e. Resource Manager mendukung non-code resources seperti strings dan grafis.

Android mendukung aplikasi dan layanan yang didesain untuk berjalan secara tidak terlihat di background. Ponsel modern saat ini merupakan perangkat perangkat yang multifungsi. Namun, dengan ukuran layar yang terbatas berarti hanya ada satu aplikasi interaktif yang dapat terlihat dalam satu waktu. Platform yang tidak mendukung layanan background membatasi kelangsungan hidup suatu aplikasi. Layanan background memungkinkan untuk membuat komponen aplikasi yang tidak terlihat yang secara otomatis menjalankan proses tanpa tindakan langsung dari pengguna. Seperti halnya software lain, android berkembang dari waktu ke waktu, yang mana terlihat dari versi yang dikeluarkan android. Berikut daftar versi android.

Android Version	API Level	Nickname
Android 1.0	1	-
Android 1.1	2	-
Android 1.5	3	Cupcake
Android 1.6	4	Donut
Android 2.0	5	Éclair
Android 2.0.1	6	Éclair
Android 2.1	7	Éclair
Android 2.2-2.2.3	8	Froyo
Android 2.3-2.3.2	9	Gingerbread
Android 2.3.3-2.3.7	10	Gingerbread
Android 3.0	11	Honeycomb
Android 3.1	12	Honeycomb
Android 3.2	13	Honeycomb
Android 4.0-4.0.2	14	Ice cream Sandwich
Android 4.0.3–4.0.4	15	Ice cream Sandwich
Android 4.1	16	Jelly Bean
Android 4.2	17	Jelly Bean
Android 4.3	18	Jelly Bean
Android 4.4	19	KitKat
Android 5.0	21	Lollipop

Tabel 2.1 *Tabel Versi Android*

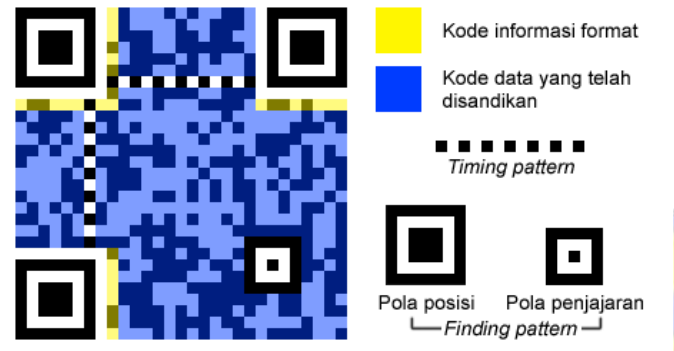
2.3 QR Code

QR code merupakan wujud barcode dua dimensi yang memiliki kemampuan menyimpan informasi berupa teks atau string. Penggunaan QR-code untuk menyimpan informasi-informasi penting belakangan ini semakin marak dan awam. QR code adalah jenis kode matriks dua dimensi yang dikembangkan oleh Denso Wave, sebuah perusahaan di Jepang, yang dipublikasikan pada tahun 1994. QR merupakan singkatan dari quick response (respon / tanggapan cepat), sehingga fungsi atau tujuan utama dari teknologi ini adalah penyampaian informasi dengan cepat dan mendapat tanggapan atau respons yang cepat pula. Oleh karena itu QR code dapat dengan mudah dibaca oleh pemindai. QR code mampu menyimpan informasi secara horisontal dan vertikal. QR code juga mampu menyimpan teks alfanumerik, kanji, kana, hiragana, simbol, biner, dan control code. Awalnya QR code digunakan untuk pelacakan bagian kendaraan pada proses manufaktur, namun kini QR code digunakan dalam konteks yang lebih luas, termasuk aplikasi komersial dan kemudahan pelacakan aplikasi yang ditujukan pada pengguna telepon seluler. Pada dasarnya bahwa QR Code dikembangkan sebagai suatu kode yang memungkinkan isinya untuk dapat diterjemahkan dengan kecepatan tinggi (Rouillard , 2008). QR Code terdiri dari sebuah untaian kotak persegi yang disusun dalam suatu pola persegi yang lebih besar.

Gambar 2.2 *QR Code*

2.3.1 Struktur QR Code

QR Code memiliki bagian-bagian struktur yang akan dijelaskan pada gambar 2.3.



Gambar 2.3 *Struktur QR Code*

Berikut merupakan penjelasan dari gambar struktur QR code :

a. Finding Pattern digunakan untuk mendeteksi posisi dari QR code.

b. Timing Pattern digunakan untuk identifikasi koordinat pusat pada QR code.

c. Daerah biru pada gambar merupakan daerah tempat data disimpan yang telah disandikan, atau data dikodekan.

d. Daerah kuning berisi informasi tentang level error correcting dan mask pattern.

2.3.2 Karakteristik QR Code

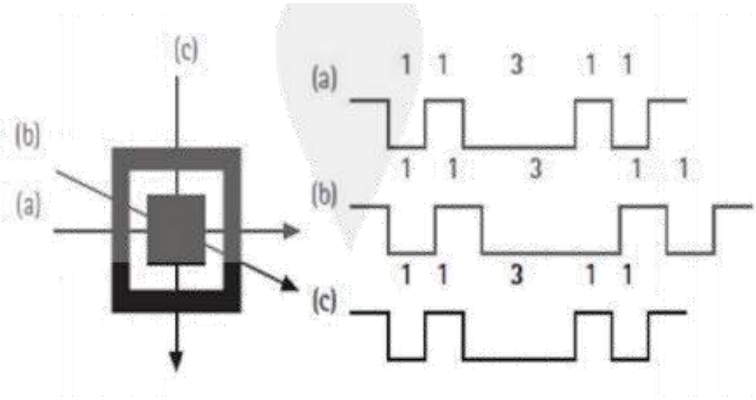
Karakteristik dari QR Code yaitu dapat menampung jumlah data yang besar. Secara teori sebanyak 7089 karakter numerik maksimum data dapat tersimpan di dalamnya, kerapatan tinggi (100 kali lebih tinggi dari kode simbol linier) dan pembacaan kode dengan cepat. QR Code juga memiliki kelebihan lain baik dalam hal unjuk kerja dan fungsi (Ariadi, 2011). Berikut ini merupakan kelebihan unjuk kerja dan fungsi yang dimiliki oleh QR Code.

1. Pembacaan data dari segala arah (360 derajat)

Pembacaan kode matriks dengan menggunakan sensor kamera CCD (ChargeCoupled Device) dimana data akan memindai baris per baris dari citra yang ditangkap dan kemudian disimpan dalam memori. Dengan menggunakan suatu perangkat lunak tertentu, detail citra akan dianalisa, finding pattern akan dikenali dan posisi simbol dideteksi. Setelah itu proses pembacaan kode akan diproses. Sedangkan pada simbol linier ataupun kode dua dimensi lain akan memakan lebih lama waktu untuk mendeteksi letak atau sudut ataupun besar dari simbol tersebut.

QR Code memiliki finding pattern yang terlihat pada gambar 2.4, untuk memberitahukan letak simbol matriks dua dimensi QR Code yang disusun pada ketiga sudutnya. Hal inilah yang membuat QR Code dapat dibaca dari segala arah atau 360 derajat. Rasio antara modul hitam dan modul putih pada finding patternnya selalu 1:1:3:1:1. Dengan rasio ini, finding pattern dapat mendeteksi keberadaan citra yang ditangkap sensor. Sebagai tambahan, dengan adanya

ketiga finding pattern maka pengkodean akan lebih cepat dua puluh kali dibandingkan kode matriks lain.



Gambar 2.4 *Finding Pattern QR Code*

2. Ketahanan Terhadap Penyimpangan Simbol

Simbol matriks 2 dimensi akan rentan terhadap penyimpangan bentuk ketika ditempatkan pada permukaan yang tidak rata (bergelombang), sehingga sensor pembaca menjadi miring karena sudut antara sensor CCD dan simbol matriks 2 dimensi ini telah berubah. Untuk memperbaiki penyimpangan ini, QR Code memiliki perata pola (Alignment pattern) yang menyusun dengan jarak yang teratur dalam satu daerah. Alignment pattern, akan memperhitungkan titik pusat dengan daerah terluar dari simbol matriks, sehingga dengan cara ini penyimpangan linier maupun non-linier masih dapat terbaca. Gambar 2.5 merupakan penyimpangan pada QR code.

Gambar 2.5 *Penyimpangan QR Code*

3. Fungsi Pemulihan Data

QR Code mempunyai empat tingkatan koreksi error (7%, 15%, 25% dan 30%) di dalam mengendalikan kerusakan yang diakibatkan kotor ataupun rusak. QR Code memanfaatkan algoritma Reed-Solomon yang tahan terhadap kerusakan tingkat tinggi. Jadi, ketika QR Code akan digunakan dalam lingkungan yang rawan kerusakan akibat dari lingkungan, disarankan untuk menggunakan koreksi error 30%. Gambar 2.6 merupakan contoh kerusakan pada QR code.

Gambar 2.6 *Kerusakan Pada QR Code*

4. Kemampuan Encode Karakter Kanji dan Kana Jepang

QR Code berkembang pesat di negara jepang. Hal ini yang menyebabkan perkembangan QR Code untuk dapat menerima input data berupa karakter yang non-alfabetis. Ketika pembuatan QR Code dengan inputan berupa huruf jepang, maka data tersebut akan diubah ke dalam bentuk biner 16 bit (2 byte) untuk karakter tunggal, sedangkan untuk gabungan karakter akan di enkode dalam biner 13 bit. Hal ini memberikan keuntungan lain dimana proses enkode huruf jepang akan meningkatkan efisien 20% lebih banyak dari simbol kode 2 dimensi lain, dimana dengan volume data yang sama akan dapat dibuat pada area percetakan yang lebih kecil. (Ariadi, 2011)

5. Fungsi *Linkin*g Pada Simbol

QR Code juga memiliki kemampuan dapat dipecah menjadi beberapa bagian dengan maksimum pembagiannya 16 bagian. Dengan fungsi linking ini, maka QR Code dicetak pada daerah yang tidak terlalu luas untuk sebuah QR Code tunggal. (Ariadi, 2011)

6. Proses *Masking*

Proses Masking pada QR Code berperan sangat penting dalam hal penyusunan modul hitam dan modul putih agar memiliki jumlah yang seimbang, untuk memungkinkan hal ini dapat digunakan pada operasi XOR yang diaplikasikan diantara area data dan daerah mask pattern. Ada sebanyak delapan

mask pattern dalam QR Code yang kesemuanya itu dalam bentuk biner tiga bit. (Ariadi, 2011)

2.3.3 Spesifikasi QR Code

QR Code memiliki kapasitas tinggi dalam hal data pengkodean, yaitu mampu menyimpan semua jenis data seperti numerik, alfanumerik, biner dan huruf kanji. Selain itu, QR Code juga memiliki empat tingkatan koreksi error yaitu 7%, 15%, 25% dan 30% di dalam mengendalikan kerusakan yang diakibatkan kotor ataupun rusak. Tabel 2.2 menjelaskan tentang spesifikasi dari QR Code.

Jenis Simbol	Minimal 21x21 Modul dan Maksimal 177x177 modul dengan peningkatan 1 versi = 4 modul	
Jenis Informasi dan Kapasitas	Numerik	Maksimum 7089 Karakter
	Alfanumerik	Maksimum 4296 Karakter
	Biner	Maksimum 2953 Karakter
	Huruf Kanji	Maksimum 1817 Karakter
Koreksi *Error*	Level L	Dapat mengembalikan data yang mengala mi kerusakan 7%
	Level M	Dapat mengembalikan data yang mengalami kerusakan 15%
	Level Q	Dapat mengembalikan data yang mengalami kerusakan 25%
	Level H	Dapat mengembalikan data yang mengalami kerusakan 30%

Tabel 2.2 *Spesifikasi QR Code*

2.4 PHP

Menurut Agus Saputra (2011, p.1) PHP atau yang memiliki kepanjangan PHP Hypertext Preprocessor merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu website dinamis. PHP menyatu dengan kode HTML, maksudnya adalah beda kondisi. HTML digunakan sebagai pembangun atau pondasi dari kerangka layout web, sedangkan PHP difungsikan sebagai prosesnya sehingga dengan adanya PHP tersebut, web akan sangat mudah di-maintenance. PHP berjalan pada sisi server sehingga PHP disebut juga sebagai bahasa Server Side Scripting. Artinya bahwa dalam setiap kali menjalankan PHP, wajib adanya web server. PHP ini bersifat open source sehingga dapat dipakai secara cuma-cuma dan mampu lintas platform, yaitu dapat berjalan pada sistem operasi Windows maupun Linux. PHP juga dibangun sebagai modul pada web server apache dan sebagai binary yang dapat berjalan sebagai CGI.

2.4.1 Kelebihan PHP

Ada beberapa alasan yang menjadi dasar pertimbangan menggunakan PHP.

1. Mudah dipelajari, alasan tersebut menjadi salah satu alasan utama untuk menggunakan PHP, Pemula pun akan mampu untuk menjadi web master PHP.
2. Mampu Lintas Platform, artinya PHP dapat / mudah diaplikasikan ke berbagai platform OS (*Operating Sytem*) dan hampir semua browser juga mendukung PHP.
3. *Free* alias gratis, bersifat *open source*.

4. PHP memiliki tingkat akses yang cepat.
5. Didukung oleh beberapa macam web server, PHP mendukung beberapa web server, seperti Apache, IIS, Lighttpd, Xitami.
6. Mendukung database, PHP mendukung beberapa database, baik yang gratismaupun yang berbayar, seperti MySQL, PostgreSQL, mSQL, Informix, SQLserver, Oracle.

2.5 MySQL

SQL merupakan kependekan Structured Query language . SQL digunakan untuk berkomunikasi dengan sebuah database. SQL adalah bahasa yang meliputi perintah-perintah untuk menyimpan, menerima, memelihara, dan mengatur aksesakses ke basis data serta digunakan untuk memanipulasi dan menampilkan data dari database.(Rosari, 2008). MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system*) atau DBMS yang *multithread, multi-user,*dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi *GNU General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

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 otomastis.

2.5.1 Kelebihan MySQL

MySQL juga memiliki beberapa kelebihan, antara lain :

1. *Portability*

   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti windows, Linux, FreeBSD, Solaris dan lain-lain.

2. *Open Source*

   MySQL didistribusikan secara *open source* (gratis), dibawah lisensi GPL sehingga dapat digunakan cuma-Cuma.

3. *Multi User*

   MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. *Performance Tuning*

   MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. *Coloumn Types*

   MySQL memiliki tipe kolom yang sangat kompleks, seperti *integer, double, char, text, date* dan lain-lain.

6. *Command and Function*

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *select* dan*where* dalam *query*.

7. *Security*

   MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta *password* terenkripsi.

8. *Scability and Limits*

   MySQL mampu menangani database dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. *Connectivity*

   MySQL dapat melakukan koneksi dengan *clients* menggunakan protokol TCP/IP, *Unix socket* (UNIX) atau *Named Pipes* (NT).

10. *Localisation*

    MySQL dapat mendeteksi pesan kesalahan pada *client* dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk didalamnya.

11. *Interface*

MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemograman dengan menggunakan fungsi API (*Aplication Programming Interface*).

12. *Clients and Tools*

MySQL dilengkapi dengan berbagai *tools* yang dapat digunakan untuk administrasi database dan pada setiap *tool* yang ada disertakan petunjuk *online*.

13. Struktur Tabel

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan database lainnya semacam PosgreeSQL ataupun Oracle.

Kelemahan MySQL dari dulu sampai saat ini adalah *feature-creep* artinya MySQL berusaha kompatibel dengan beberapa standar serta berusaha memenuhinya namun jika itu diungkapkan kenyataannya bahwa fitur-fitur tersebut belum lengkap dan belum berperilaku sesuai standar. Contoh fitur *SUB-SELECT* (nesting *SELECT* dalam *SELECT*) yang tidak optimal dan sering salah *parsing query SQL* dan jalan keluarnya dengan memecah menjadi beberapa *query*.

2.6 Java

Java merupakan bahasa pemrograman yang disusun oleh James Gosling yang dibantu oleh rekan-rekannya di suatu perusahaan perangkat lunak yang bernama *Sun Microsystems*, pada tahun 1991. Bahasa pemrograman ini mula-mula di inisialisasi dengan nama "*Oak*", namun pada tahun 1995 diganti namanya menjadi "Java".

Java berdiri di atas sebuah mesin penterjemah (*interpreter*) yang diberi nama *Java Virtual Machine* (JVM). JVM inilah yang akan membaca kode bit (*bytecode*) dalam file *.class* dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu bahasa Java disebut sebagai bahasa pemrograman yang *portable* karena dapat dijalankan pada berbagai sistem operasi, asalkan pada system operasi tersebut terdapat JVM. Alasan utama pembentukan bahasa Java adalah untuk membuat aplikasi-aplikasi yang dapat diletakkan di berbagai macam perangkat elektronik, sehingga Java harus bersifat tidak bergantung pada platform (*platform independent*). Itulah yang menyebabkan dalam dunia pemrograman Java dikenal adanya istilah „*write once, run everywhere*", yang berarti kode program hanya ditulis sekali, namun dapat dijalankan di bawah kumpulan pustaka (*platform*) manapun, tanpa harus melakukan perubahan kode program.

2.6.1 Arsitektur Java

Secara arsitektur, Java tidak berubah sedikitpun sejak awal mula bahasa tersebut dirilis. *Compiler* Java (yang disebut dengan javac atau *Java Compiler*) akan mentransformasikan kode-kode dalam bahasa Java ke dalam suatu kode bit.

Dimana *bytecode* adalah sekumpulan perintah hasil kompilasi yang kemudian dapat dieksekusi melalui sebuah mesin komputer abstrak, yang disebut dengan JVM (*Java Virtual Machine*). JVM juga sering dinamakan sebagai *interpreter*, karena sifatnya yang selalu menerjemahkan kode-kode yang tersimpan dalam kode bit dengan cara baris demi baris. Untuk menjalankan program Java, maka file dengan ekstensi *.java* harus dikompilasi menjadi file kode bit. Dimana untuk menjalankan kode bit tersebut dibutuhkan JRE (*Java Runtime Environment*) yang memungkinkan pemakai untuk menjalankan program Java, hanya menjalankan, tidak untuk membuat kode baru lagi. JRE berisi JVM dan pustaka Java yang digunakan.

2.6.2 Java 2

Sun Microsystems telah mendefinisikan tiga buah edisi dari Java 2, yaitu sebagai berikut :

1. ***Java 2 Standard Edition*** **(J2SE)**, adalah inti dari bahasa pemrograman Java. JDK merupakan salah satu perangkat (*tool*) dari J2SE untuk mengkompilasi dan menjalankan program Java. Di dalamnya terdapat 10 perangkat untuk mengkompilasi program Java dan JRE. J2SE ini digunakan pada perangkat keras seperti layar komputer (*desktop*).

2. ***Java 2 Enterprise Edition*** **(J2EE)**, merupakan kumpulan tertinggi (*superset*) dari J2SE yang memperbolehkan kita untuk mengembangkan aplikasi-aplikasi berskala besar (*enterprise*) karena dijalankan pada jaringan komputer.

3. ***Java 2 Micro Edition* (J2ME)**, merupakan kumpulan bagian (*subset*) dari J2SE yang digunakan untuk menangani pemrograman di dalam perangkat perangkat kecil, yang tidak memungkinkan untuk mendukung implementasi J2SE secara penuh. Paket J2ME digunakan pada perangkat yang memiliki kapasitas memori kecil seperti telepon selular, pager atau PDA.

2.6.3 Java 2 Micro Edition (J2ME)

Java2 *Micro Edition* atau yang biasa disebut J2ME adalah lingkungan pengembangan yang didesain untuk meletakan perangkat lunak Java pada barang elektronik beserta perangkat pendukungnya. Pada J2ME, jika perangkat lunak berfungsi baik pada sebuah perangkat maka belum tentu juga berfungsi baik pada perangkat lainnya. J2ME membawa Java ke dunia informasi, komunikasi, dan perangkat komputasi selain perangkat komputer *dekstop* yang biasanya lebih kecil dibandingkan perangkat komputer *dekstop*. J2ME biasa digunakan pada telepon seluler, *pager*, *personal digital assistants* (PDA‟s) dan sejenisnya. 11 J2ME adalah bagian dari J2SE, karena itu tidak semua *library* yang ada pada J2SE dapat digunakan pada J2ME. Tetapi J2ME mempunyai beberapa *library* khusus yang tidak dimiliki J2SE. Arsitektur J2ME dapat dilihat pada gambar 2.7.

Gambar 2.7 *Arsitektur J2ME*

2.6.4 Kelebihan Java

1. Sederhana dan Ampuh

Java dirancang untuk mudah dipelajari, terutama bagi *programmer* yang telah mengenal C/C++ akan mudah sekali untuk berpindah ke *Java*. Pemakai dapat belajar membuat program dengan *Java* secara cepat jika telah memahami konsep dasar pemrograman berorientasi objek. *Java* tidak memiliki hal-hal yang mengejutkan dan aneh. *Java* memberi anda kemampuan untuk menuangkan semua ide, karena bahasa pemrograman 5 ini bukan merupakan *scripting language* (bahasa naskah) yang menghilangkan kemampuan kita untuk berinovasi, tetapi dengan cara berorientasi objek yang mudah dan jelas.

2. Aman

Java dirancang sebagai bahasa pemrograman yang handal dan aman. Aplikasi-aplikasi yang dibangun dengan bahasa *Java* sangat handal dengan manajemen memori yang bagus. Aplikasi *Java* juga dikenal sangat *secure,*yaitu kasus-kasus seperti *buffer everflow* yang umumnya menjadi lubang keamanan aplikasi-aplikasi berbasis C/C++ tidak terjadi di *Java*, karena pengaturan *security* yang bagus.

3. Berorientasi Objek

Paradigma pemrograman berorientasi objek merupakan paradigma pemrograman masa depan. *Java* merupakan bahasa pemrograman berorientasi objek. *Java* bukan turunan langsung dari bahasa pemrograman manapun, juga sama sekali tidak kompetibel dengan semuanya. *Java* memiliki keseimbangan, menyediakan

mekanisme peng-*class*-an sederhana, dengan model antar muka dinamik yang intuitif hanya jika diperlukan.

4. Kokoh

Java membatasi anda dari beberapa hal kunci supaya anda dapat menemukan kesalahan lebih cepat saat mengembangkan program. *Java* langsung memeriksa program saat anda menuliskannya, dan sekali lagi ketika program di jalankan. Karena *Java* adalah bahasa yang sangat ketat dalam hal tipe data dan deklarasi, banyak kesalahan umum terjadi saat kompilasi. Hal ini akan lebih menghemat waktu jika dibandingkan dengan keharusan menjalankan program terlebih dahulu dan memeriksa semua bagian program untuk melihat ketidakcocokan dinamis selama program berjalan. Ini adalah contoh di mana *Java* lebih luwes dan kokoh dari beberapa bahasa lain, tetapi dengan imbalan yang layak untuk kelebihan itu.

5. Interaktif

Java memiliki beberapa kemampuan yang memungkinkan program melakukan beberapa hal pada saat bersamaan, tanpa harus kesulitan menangani proses yang akan terjadi selanjutnya. Jalinan program-program *Java* yang mudah digunakan memungkinkan kita untuk memikirkan pembuatan perilaku khusus, tanpa harus mengintegrasikan perilaku tersebut dengan model pemrograman global yang mengatur perulangan kejadian.

6. Netral Terhadap Berbagai Arsitektur

Java telah mengambil beberapa keputusan yang sulit dalam pembuatan bahasa *Java* dan bagaimana program 6 dijalankan, jadi anda dapat sepenuhnya percaya "tulis sekali, jalan di mana saja, kapan saja, selamanya".

7. Terinterpretasi dan Berkinerja Tinggi

Java dilengkapi keajaiban lintas platform yang luar biasa dengan kompilasi ke dalam representasi langsung yang disebut kode-byte *Java* (*Java* byte-code), yang dapat diterjemahkan oleh sistem manapun yang memilki program *Java* didalamnya. *Java*, bagaimanapun dirancang untuk tetap berkinerja baik pada CPU yang tidak terlalu kuat. Walaupun *Java* merupakan bahasa terinterpretasi, kodekode *Java* telah dirancang dengan hatihati sehingga mudah diterjemahkan ke dalam bahasa asli suatu mesin untuk menghasilkan kinerja yang tinggi. Sistem program *Java* yang melakukan optimasi tepat waktu tersebut tidak kehilangan keuntungan dari program yang netral terhadap platform. "lintas platform berkinerja tinggi" bukan sekedar omong-kosong. Dalam aplikasi *Java* (*.class) merupakan *Java bytecode* yang berjalan di atas jvm (*Java Virtual Machine*), yang kemudian jvm-lah yang akan menginterpresentasikan kode-kode tersebut ke kode native atau kode mesin dari arsitektur yang bersangkutan. Hal sangat menarik karena urusan arsitektur mesin bukan jadi masalah bagi programmer tapi menjadi urusan kompiler pada bahasa pemrograman *Java*.

2.7 Konsep Kriptografi

Kriptografi secara umum adalah ilmu dan seni untuk menjaga kerahasiaan informasi. Selain pengertian tersebut terdapat pula pengertian kriptografi sebagai ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek kemanan informasi seperti kerahasiaan, keutuhan data, integritas data serta autentikasi.

Konsep kriptografi sendiri telah lama digunakan oleh manusia misalnya pada peradaban Mesir dan Romawi walau masih sangat sederhana. Prinsip-prinsip yang mendasari kriptografi yakni (Nababan, 2011) :

1. *Confidentality*

Confidentiality (kerahasiaan) yaitu layanan yang ditujukan untuk menjaga agar isi pesan yang di kirimkan tidak dapat dibaca oleh pihak lain (kecuali pihak pengirim, pihak penerima atau pihak-pihak yang memiliki ijin). Umumnya hal ini dilakukan dengan cara menyandikan pesan menjadi ciphertext sehingga sulit dibaca dan dipahami.

2. *Data Integrity*

Data integrity (keutuhan data) yaitu layanan yang mampu menjamin pesan masih asli atau utuh atau belum pernah dimanipulasi selama masa waktu 23 pengiriman. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi adanya manipulasi pesan tersebut oleh pihak-pihak yang tidak berhak antara lain penghapusan, pengubahan atau penambahan data yang tidak sah oleh pihak lain.

3. *Authentication*

Authentication (otentikasi) yaitu layanan yang berhubungan dengan identifikasi. Baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi maupun mengidentifikasi kebenaran sumber pesan. Dua pihak yang saling berkomunikasi harus dapat mengotentikasi satu sama lain sehingga ia dapat memastikan sumber pesan. Pesan yang di kirim melalui saluran komunikasi juga harus di otentikasi asalnya. Dengan kata lain, aspek keamanan ini dapat di ungkapkan sebagai pertanyaan : apakah pesan yang diterima benar-benar berasal dari pengirim yang benar.

4. *Non-Repudiation*

Non-repudiation (anti penyangkalan) yaitu layanan yang dapat mencegah suatu pihak untuk menyangkal aksi yang dilakukan sebelumnya, misalnya pengirim pesan menyangkal melakukan pengiriman atau penerima pesan menyangkal telah menerima pesan. Sebagai contoh, misalnya pengiriman pesan memberi otoritas kepada penerima pesan untuk melakukan pembelian, namun kemudian ia menyangkal telah memberikan otoritas tersebut.

Dalam bidang kriptografi terdapat istilah-istilah yang sering digunakan, diantaranya adalah sebagai berikut :

1. *Message Plaintext* dan *Chiphertext*

Message (pesan) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah plaintext (cleartext). Pesan dapat berupa data atau informasi yang di kirim melalui kurir, saluran

telekomunikasi maupun saluran lain. Pesan yang tersimpan tidak hanya berupa text, tetapi juga dapat berbentuk citra (image), suara atau bunyi (audio) dan video atau berkas biner lainnya. Agar pesan tidak dapat dimengerti maknanya oleh pihak lain, maka pesan perlu disandikan kebentuk lain yang tidak dapat dipahami. Bentuk pesan yang tersandi disebut chiphertext atau sering juga disebut kriptogram. Chiphertext harus dapat ditransformasikan kembali menjadi plaintext semula agar pesan yang diterima dapat dibaca.

2. Pengirim dan Penerima

Komunikasi data melibatkan pertukaran pesan antara dua entitas. Pengirim (sender) adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima (receiver) adalah entitas yang menerima pesan. Entitas yang dimaksud dapat berupa orang, mesin (komputer, kartu kredit) dan sebagainya. Jadi orang dapat bertukar pesan dengan orang lainnya (Kevin berkomunikasi dengan John) sementara didalam jaringan komputer mesin (komputer) berkomunikasi dengan mesin. Contoh : mesin ATM dengan komputer server di bank.

3. Enkripsi dan Deskripsi

Proses menyandikan plaintext menjadi chiphertext disebut enkripsi, sedangkan proses mengembalikan chiphertext menjadi plaintext kebentuk teks semula (pesan asli) disebut deskripsi. Enkripsi dan deskripsi dapat diterapkan baik pada pesan yang dikirim maupun pada pesan tersimpan.

4. *Chiper* dan Kunci

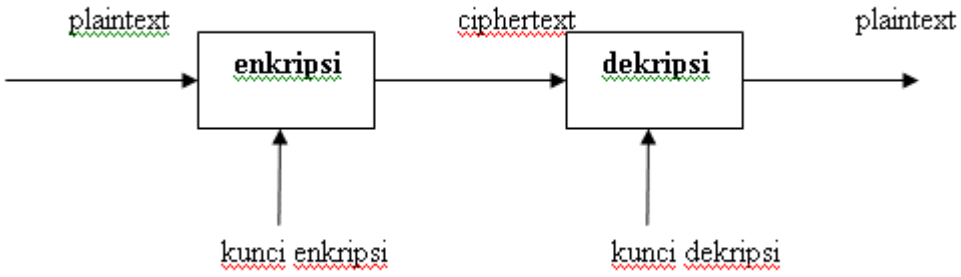
Algoritma kriptografi disebut juga chipher yaitu aturan untuk enchiphering dan dechiphering, atau fungsi matematik yang digunakan untuk enkripsi dan deskripsi. Konsep matematis yang mendasari algoritma kriptografi adalah relasi antara dua buah himpunan yaitu himpunan yang berisi elemen-elemen plaintext dan himpunan yang berisi chiphertext. Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemen antara kedua himpunan tersebut. Misalkan P menyatakan plaintext dan C menyatakan chiphertext, maka fungsi enkripsi E memetakan P ke C E(P) = C Dan fungsi dekripsi D memetakkan C ke P D(C) = P Karena proses enkripsi kemudian deskripsi mengembalikan pesan ke pesan asal, maka kesamaan berikut harus benar : D (E(P) = P

Kriptografi modern juga telah banyak mengatasi masalah dengan penggunaan kunci, yang dalam hal ini algoritma tidak lagi dirahasiakan, tetapi kunci harus dijaga kerahasiaannya. Kunci (key) adalah parameter yang digunakan untuk transformasi enciphering dan deciphering. Kunci biasanya berupa string atau deretan bilangan. Dengan menggunakan kunci K, maka fungsi enkripsi dan dekripsi dapat ditulis sebagai berikut :

$E_k(P)=C \text{dan} D_k(C)=P$

Dan kedua fungsi ini memenuhi: $D_k(E_k(P))= P$

Gambar 2.7 berikut ini merupakan sebuah ilustrasi dari skema enkripsi dan deskripsi dengan menggunakan kunci terhadap sebuah pesan.



Gambar 2.8 *Proses Enkripsi Dan Deskripsi Sederhana*

sedangkan untuk besar data yang akan diolah dalam satu kali proses, maka algoritma kriptografi dapat dibedakan menjadi dua jenis yaitu :

a. Algoritma *Block Chiper*

Informasi atau data yang hendak dikirim dalam bentuk blok-blok besar (misalnya 64 bit) dimana blok-blok ini dioperasikan dengan fungsi enkripsi yang sama dan akan menghasilkan informasi rahasia dalam blok-blok yang berukuran sama.

b. Algortima *Stream Chiper*

Informasi atau data yang hendak dikirim dioperasikan dalam bentuk blok-blok yang lebih kecil (byte atau bit), biasanya satu karakter persatuan waktu proses, menggunakan tranformasi enkripsi yang berubah setiap waktu.

5. Penyadap

Penyadap (eavesdropper) adalah orang yang mencoba menangkap pesan selama ditransmisikan. Tujuan penyadap adalah untuk mendapatkan informasi

sebanyak banyaknya mengenai sistem kriptografi yang digunakan untuk berkomunikasi dengan maksud memecahkan chiphertext.

6. Kriptanalisis dan Kriptologi

Kriptanalisis adalah ilmu dan seni untuk memecahkan ciphertext menjadi plainteks tanpa mengetahui kunci yang digunakan dan pelakunya disebut Kriptanalis. Jika seorang kriptografer mentransformasikan plainteks menjadi ciphertext dengan suatu algoritma dan kunci maka sebaliknya seorang kriptanalis berusaha untuk memecahkan chiphertext tersebut untuk menemukan plainteks atau kunci.

2.7.1 *Authentication* (Autentikasi)

Autentikasi, berhubungan dengan identifikasi atau pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri, untuk validasi user pada saat memasuki sistem. Dalam hal ini autentikasi merupakan sebuah proses identifikasi yang dilakukan oleh pihak yang satu terhadap pihak yang lain ataupun sebaliknya dengan melakukan berbagai proses identifikasi untuk memastikan keaslian dari informasi yang diterima.

Identifikasi terhadap suatu informasi dapat berupa waktu pembuatan informasi, waktu pengiriman informasi, isi informasi, kepastian pengirim maupun penerima data. Pada umumnya hal yang paling mendasar dalam penggunaan metode autentikasi adalah berhubungan dengan metode untuk memastikan dan menyatakan bahwa data informasi benar-benar asli dan orang yang mengirim maupun menerima data adalah benar-benar orang yang asli. Autorisasi ini di set

up oleh administrator, webmaster atau pemilik situs sebagai pemegang hak tertinggi atau mereka yang ditunjuk pada sistem tersebut.

Untuk proses ini masing-masing user akan di cek dari data yang diberikannya seperti nama, password serta hal-hal lainnya. Dalam aplikasi web dibutuhkan mekanisme yang dapat melindungi data dari pengguna yang tidak berhak mengaksesnya. misalnya sebuah situs web yang memiliki informasi yang bersifat rahasia dan hanya dapat diakses oleh pengguna yang bersangkutan saja. Berikut merupakan beberapa contoh metode autentikasi yang digunakan :

a. Sesuatu yang diketahui oleh pengguna, seperti: password, passphrase, dan PIN (Personal Identification Number).

b. Sesuatu yang dimiliki oleh pengguna, seperti: ID card, kartu kredit, telepon seluler, dan perangkat token.

c. Sesuatu yang ada pada pengguna, seperti: sidik jari, DNA, suara, pola retina, atau aspek biometrik lain.

d. Berbasis pengenalan (recognition) atau autentikasi cognometric, yaitu sesuatu yang dikenal oleh pengguna seperti: Pengguna harus mengenali dari beberapa wajah yang dirahasiakan.

e. Berbasis cybermetric, yaitu sesuai yang ada pada komputer seperti: Membatasi akses hanya dari komputer yang memiliki kombinasi unik hardware dan software tertentu.

f. Berbasis lokasi, seperti: Membatasi penggunaan ATM atau kartu kredit hanya pada cabang tertentu, membatasi login root hanya dari terminal tertentu.

g. Berbasis waktu, seperti: Membatasi penggunaan sebuah akun hanya pada waktu tertentu, misalnya jam kerja.

h. Berbasis ukuran, seperti: Membatasi terjadinya transaksi pada jumlah tertentu saja.

Untuk itu pada penelitian ini, penulis melakukan inovasi dengan mengembangkan prototipe *QR code* yang akan digunakan sebagai autentikasi keamanan login pada website dengan memanfaatkan teknologi android.

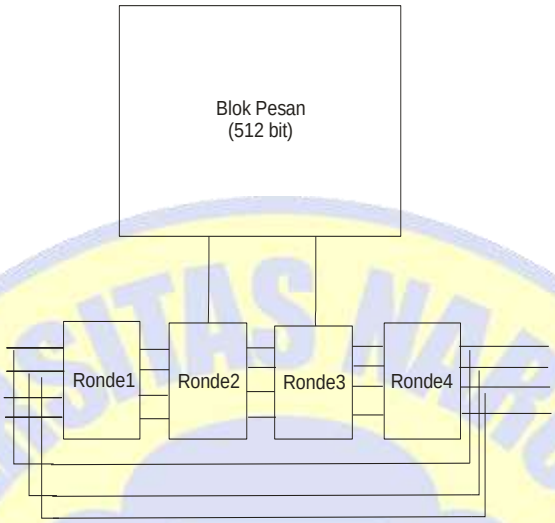
2.7.2 MD5 (*Message-Digest Algorithm 5*)

MD5 ialah fungsi hash kriptografik yang digunakan secara luas dengan *hash value* 128-bit. Pada bagian ini dijelaskan mengenai sistem kriptografi MD5 secara spesifik, yaitu sistem kriptografi algoritma MD5 yang menjelaskan dari awal masukan hingga keluarannya.

A. Prinsip dasar MD5

Message Digest 5 (MD-5) adalah salah satu penggunaan fungsi hash satu arah yang paling banyak digunakan. MD-5 merupakan fungsi hash kelima yang dirancang oleh Ron Rivest dan didefinisikan pada RFC 1321[10]. MD-5 merupakan pengembangan dari MD-4 dimana terjadi penambahan satu ronde. MD-5 memproses teks masukan ke dalam blok-blok bit sebanyak 512 bit, kemudian dibagi ke dalam 32 bit sub blok sebanyak 16 buah. Keluaran dari MD-5

berupa 4 buah blok yang masing-masing 32 bit yang mana akan menjadi 128 bit yang biasa disebut nilai hash. Pada Gambar 2.9 terlihat simpul utama dari MD- 5.



Gambar 2.9 *Simpul Utama MD5*

Simpul utama MD5 mempunyai blok pesan dengan panjang 512 bit yang masuk ke dalam 4 buah ronde. Hasil keluaran dari MD-5 adalah berupa 128 bit dari byte terendah A dan tertinggi byte D. Terdapat 5 langkah yang dibutuhkan untuk menghitung intisari pesan. Adapun langkah–langkah tersebut dijelaskan sebagai berikut:

1. Menambahkan bit

Pesan akan ditambahkan bit-bit tambahan sehingga panjang bit akan kongruen dengan 448, mod 512. Hal ini berarti pesan akan mempunyai panjang yang hanya kurang 64 bit dari kelipatan 512 bit. Penambahan bit selalu dilakukan walaupun panjang dari pesan sudah kongruen dengan 448, mod 512 bit. Penambahan bit dilakukan dengan menambahkan “1” di awal dan diikuti “0”

sebanyak yang diperlukan sehingga panjang pesan akan kongruen dengan 448, mod 512.

2. Penambahan panjang pesan

Setelah penambahan bit, pesan masih membutuhkan 64 bit agar kongruen dengan kelipatan 512 bit. 64 bit tersebut merupakan perwakilan dari b (panjang pesan sebelum penambahan bit dilakukan). Bit-bit ini ditambahkan ke dalam dua word (32 bit) dan ditambahkan dengan low-order terlebih dahulu. Penambahan pesan ini biasa disebut juga MD Strengthening atau Penguatan MD.

3. Inisialisasi MD5

Pada MD-5 terdapat empat buah word 32 bit register yang berguna untuk menginisialisasi message digest pertama kali. Register-register ini di inisialisasikan dengan bilangan hexadesimal.

word A: 01 23 45 67

word B: 89 AB CD EF

word C: FE DC BA 98

word D: 76 54 32 10

Register-register ini biasa disebut dengan nama Chain variabel atau variabel rantai.

4. Proses pesan didalam blok 16 word

Pada MD-5 juga terdapat 4 (empat) buah fungsi nonlinear yang masing-masing digunakan pada tiap operasinya (satu fungsi untuk satu blok),

yaitu:

$F(X,Y,Z) = (X \wedge Y) \vee((\neg X) \wedge Z)$

$G(X,Y,Z) = (X \wedge Z) \vee(Y \wedge(\neg Z))$

$H(X,Y,Z) = X \oplus Y \oplus Z$

$I(X,Y,Z) = Y \oplus(X \vee(\neg Z))$

($\oplus$untuk XOR, $\wedge$untuk AND, $\vee$untuk OR dan $\neg$untuk NOT).

Pada Gambar 3.2 dapat dilihat satu buah operasi dari MD-5 dengan operasi yang dipakai sebagai contoh adalah FF(a,b,c,d,Mj,s,ti) menunjukan a = b + ((a + F(b,c,d) + Mj + ti) <<< s)

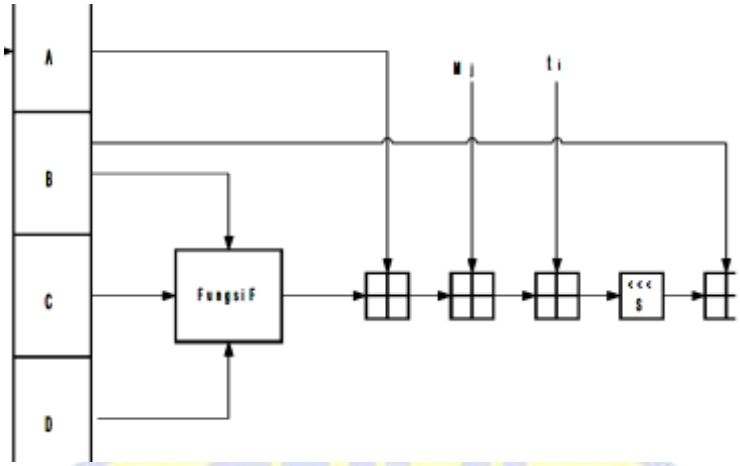
Bila Mj menggambarkan pesan ke-j dari sub blok (dari 0 sampai 15) dan <<<s menggambarkan bit akan digeser ke kiri sebanyak s bit, maka keempat operasi dari masing-masing ronde adalah:

FF(a,b,c,d,Mj,s,ti) menunjukan a = b + ((a +F(b,c,d) + Mj + ti) <<< s)

GG(a,b,c,d,Mj,s,ti) menunjukan a = b + ((a +G(b,c,d) + Mj + ti) <<< s)

HH(a,b,c,d,Mj,s,ti) menunjukan a = b + ((a +H(b,c,d) + Mj + ti) <<< s)

II(a,b,c,d,Mj,s,ti) menunjukan a = b + ((a +I(b,c,d) + Mj + ti) <<< s)



Gambar 2.10 *Operasi MD5*

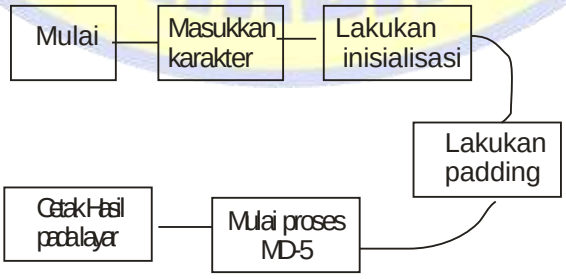
Konstanta ti didapat dari integer 232. abs(sin(i)), dimana i dalam radian.

5. Keluaran MD5

Keluaran dari MD-5 adalah 128 bit dari word terendah A dan tertinggi word D masing-masing 32 bit.

B. Proses MD5 dengan masukan berupa string

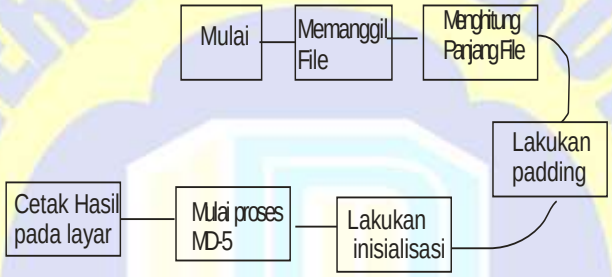
Proses MD5 dengan masukan berupa string adalah proses yang masukan nya berupa karakter-karakter yang dimasukan melalui keyboard. Hal ini dapat dilihat pada gambar 2.11.



Gambar 2.11 *Proses MD5 Berupa String*

C. Proses MD5 dengan masukkan berupa file

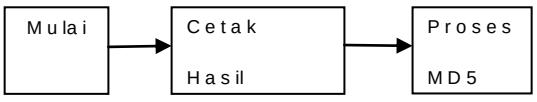
Proses MD5 dengan masukan berupa file adalah proses MD5 yang masukannya memanggil file yang kemudian dihitung berapa panjang bitnya, dalam keadaan ini file diperlakukan sebagai bit memori sehingga masukannya tidak terpengaruh pada ekstensinya. Kemudian dilakukan proses MD5. Hal ini dapat dilihat pada Gambar 2.12



Gambar 2.12 *Proses MD5 Berupa File*

D. Proses MD5 sebagai test suite

Test suite dilakukan untuk mengetahui apakah program yang dibuat ini sudah benar atau tidak. Sebagai perbandingannya digunakan hasil yang sudah dibuat oleh Ron Rivest yang sudah didefinisikan pada RFC 1321. Pada Gambar 2.13 dapat dilihat bahwa masukan dari MD5 sudah ditentukan sehingga hanya membandingkan hasil pada layar dengan yang tercantum pada RFC 1321.



Gambar 2.13 *Proses MD5 Sebagai Test Suite*

2.8 XAMPP

XAMPP adalah software open source multiplatform yang fungsinya sebagai web server. yang dapat dijalankan pada sistem operasi (Windows, Linux, MacOS, dan Solaris), Apache, MySQL, PHP dan Perl. Artinya XAMPP merupakan server Apache dan MySQL yang ditulis dalam bahasa PHP dan Perl. XAMPP merupakan perangkat lunak yang cukup populer digunakan untuk mengembangkan web karena sangat mudah untuk di install dan digunakan. Pada pengembangan prototipe aplikasi ini, XAMPP digunakan sebagai web server untuk menjalankan aplikasi php yang berfungsi sebagai halaman untuk autentikasi login menggunakan QR code. Halaman php ini disimpan dalam folder "htdocs" yang merupakan folder penyimpanan untuk halaman web yang dihosting di XAMPP.

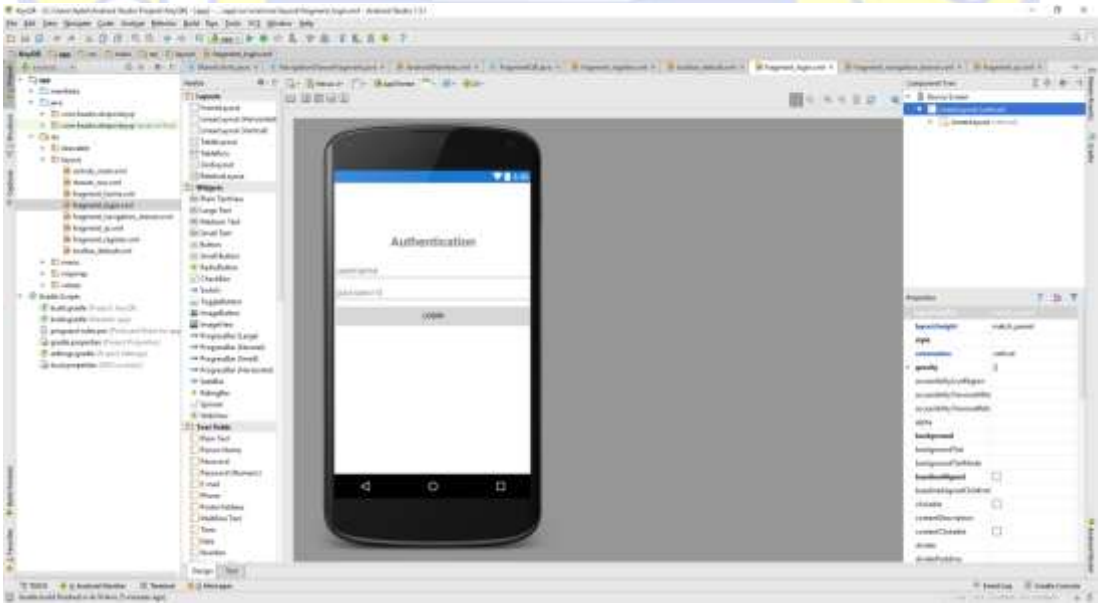
2.9 Android Studio

Android Studio ini adalah lingkungan pengembangan baru dan terintegrasi penuh, yang baru saja dirilis oleh Google untuk sistem operasi Android. Android Studio dirancang untuk menjadi peralatan baru dalam pengembangan aplikasi dan juga memberi alternatif lain selain Eclipse yang saat ini menjadi IDE yang paling banyak dipakai. Saat memulai proyek baru dengan Android Studio, struktur proyek akan muncul bersama dengan hampir semua berkas yang ada di dalam direktori SDK, peralihan ke sistem manajemen berbasis Gradle ini memberikan fleksibilitas yang lebih besar pada proses pembangunanya.

Android Studio mengijinkan untuk melihat perubahan visual apapun yang dilakukan pada aplikasi secara langsung. dan juga bisa melihat perbedaannya jika

dipasang pada beberapa perangkat Android berbeda, termasuk konfigurasi dan resolusinya secara bersamaan. Fitur lain di Android Studio adalah alat-alat baru untuk mengepak dan memberi label kode. Dengan begitu mengijinkan pengguna tetap menjadi yang teratas ketika berurusan dengan banyak kode.

Program ini juga memakai sistem seret dan jatuhkan untuk memindahkan komponen melalui antar muka pengguna. Selain itu, lingkungan baru ini juga mendukung Google Cloud Messaging. Sebuah fitur yang mengijinkan pengguna untuk mengirim data dari server ke perangkat Android pengguna melalui cloud, cara terbaik untuk mengirim pengingat pada aplikasi pengguna. Program ini juga membantu untuk melokalisasi aplikasi, memberi gambaran visual untuk tetap memprogram sambil mengontrol alur dari aplikasi. Framework android studio dapat dilihat pada gambar 2.14.



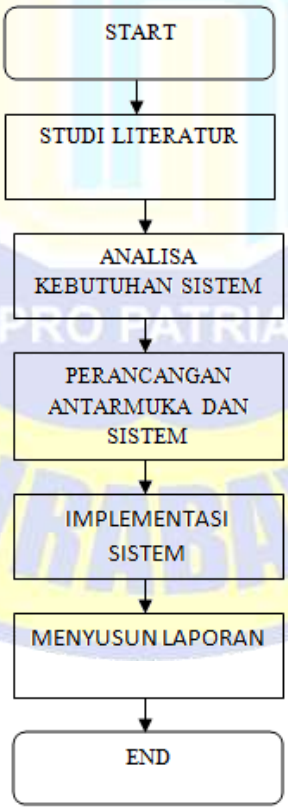
Gambar 2.14 *Framework Android Studio*

BAB III

METODOLOGI PENELITIAN

3.1 Diagram Metodologi Penelitian

Metode yang digunakan dalam kegiatan penelitian ini meliputi: Studi literatur, analisa kebutuhan sistem, perancangan antarmuka dan sistem, implementasi sistem, dan menyusun laporan. Tahapan yang dilakukan adalah seperti pada gambar 3.1.



Gambar 3.1 *Diagram Metodologi Penelitian*

3.2 Studi Literatur

Studi literatur, penulis gunakan untuk mengumpulkan referensi-referensi yang berkaitan dengan penelitian yang sedang penulis lakukan. Studi literatur dilakukan dengan membaca langsung dari media buku, beberapa jurnal penelitian terdahulu dan internet. Setelah mendapat referensi yang relevan maka penulis melakukan pencarian informasi-informasi yang dibutuhkan dalam penelitian. Penulis menggunakan informasi yang didapatkan sebagai acuan untuk menyusun landasan teori, metodologi penelitian, dan tentunya untuk pengembangan aplikasi secara langsung. Berbagai macam referensi yang didapatkan dapat dilihat langsung pada daftar pustaka.

3.3 Analisa Kebutuhan Sistem

Pada tahap ini dilakukan analisa kebutuhan dari sistem, meliputi spesifikasi software dan hardware yang diperlukan untuk pembuatan aplikasi. Kebutuhan yang diperlukan untuk membuat aplikasi ini adalah sebagai berikut :

1. Spesifikasi Software

- Windows 7 Ultimate 64-bit sebagai sistem operasi
- Android Studio sebagai IDE platform pada android
- XAMPP sebagai web server
- MySQL sebagai database

2. Spesifikasi Hardware

- Samsung galaxy core 2 SM-G355H. Android version kitkat 4.4.2

3.4 Perancangan Antarmuka Dan Sistem

Perancangan alur kerja sistem memvisualisasi ide dari aplikasi yang akan dibangun. Sehingga dapat memberikan gambaran dari aplikasi yang akan dihasilkan. Dan perancangan antarmuka bertujuan untuk memudahkan dalam pembuatan aplikasi, serta merancang agar aplikasi yang ditampilkan menjadi menarik dan mudah untuk digunakan pengguna.

3.5 Implementasi Sistem

Implementasi sistem merupakan tahapan dimana rancangan sistem dituliskan dalam baris kode dimulai dengan pembuatan kode php yang akan ditempatkan pada web server. dan aplikasi mobile yang akan dijalankan dari perangkat android milik pengguna.

3.6 Menyusun Laporan

Tahapan terakhir dari penelitian ini adalah menuliskan hasil dari penelitian yang telah dilakukan ke dalam bentuk laporan tugas akhir. Laporan berisi latar belakang yang melandasi pembuatan sistem, perumusan masalah beserta batasannya, tujuan dan manfaat, landasan teori yang mendukung, metodologi penelitian, hasil dan pembahasan, dan yang terakhir adalah kesimpulan dan saran.

BAB IV

HASIL DAN PEMBAHASAN

Setelah metodologi penelitian disusun secara sistematis mengenai metode-metode yang digunakan dalam perancangan aplikasi ini, maka tahap hasil dan pembahasan ini merupakan hasil dokumentasi dari pengaplikasian metode yang digunakan pada bab sebelumnya (metodologi penelitian). Berikut ini adalah item-item yang dibahas dalam hasil dokumentasi dari penelitian "**Pengembangan Prototipe *QR Code* sebagai autentikasi keamanan login sistem dengan memanfaatkan teknologi android**" :

1. Deskripsi Umum Sistem
2. Perancangan Desain Sistem
3. Perancangan Antarmuka
4. Pengujian Sistem

Masing-masing item akan dibahas dan dijelaskan secara lebih terperinci sebagai berikut :

4.1 Deskripsi Umum Sistem

Pada bagian ini digambarkan proses keseluruhan dari sistem atau merupakan gambaran umum dari sistem yang akan dibuat.

Pengembangan aplikasi ini dirancang untuk menyimpan data *username* dan *password* yang didaftarkan oleh *user* ke dalam database, *username* dan

password dari database tersebut kemudian digenerate kedalam *QR Code*. Setelah proses generate *QR code* selesai, *user* dapat melakukan *request QR code* yang nantinya akan digunakan pada saat login di website yang telah terdapat data *user* yaitu *username* dan *password* pada database. Dengan cara pengguna harus memindai *QR Code* yang telah digenerate oleh aplikasi tersebut melalui kamera *webcam*. Sistem kemudian melakukan pencocokan *QR Code* pada database.

Apabila *QR Code* tersebut cocok, maka pengguna dapat langsung masuk kedalam sistem dan secara otomatis *QR Code* tersebut hangus sehingga jika *user* ingin melakukan login lagi maka sistem secara otomatis melakukan *regenerate QR Code* yang baru. Apabila *QR Code* tersebut tidak cocok maka pengguna diharuskan untuk melakukan *request* ulang *QR Code*. Dan tentunya *QR Code* yang didapat nantinya berbeda dengan *QR Code* sebelumnya, dikarenakan sistem hanya memperbolehkan satu kali pemakaian *QR Code* saja untuk login. Aplikasi ini membutuhkan XAMPP sebagai web server dan MySQL sebagai database untuk menjalankan aplikasi ini.

4.2 Perancangan Desain Sistem

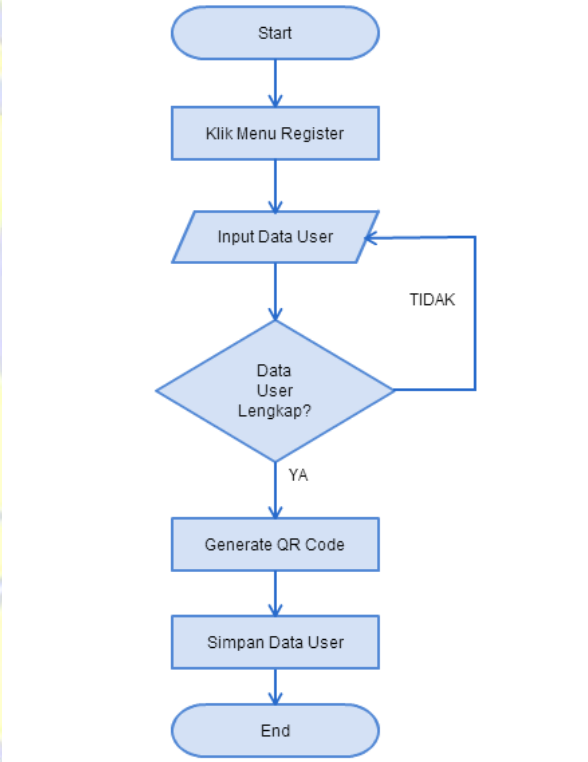
Perancangan aplikasi ini dibagi dalam dua tahap proses. Yaitu tahap proses registrasi, dan tahap proses autentikasi. Berikut adalah proses kerja pada masing – masing tahapan proses.

1. Tahap Registrasi

Pada tahap registrasi, *user* diminta untuk memberikan input *username* dan *password* yang hendak didaftarkan. Terdapat prosedur pemeriksaan pada tahap registrasi ini. Apabila *user* tidak memberi input pada *textbox* (textbox dibiarkan

kosong) atau menggunakan username yang sudah ada, kemudian menekan tombol daftar.

Sistem akan memberikan peringatan kepada *user* untuk melengkapi pendaftaran atau mengganti *username* yang belum ada pada database. Setelah penginputan data selesai sistem akan melakukan *generate QR Code* secara otomatis dan menyimpan hasil *generate QR code* tersebut ke dalam database. Alur kerja tahap registrasi dapat dilihat pada gambar 4.1.



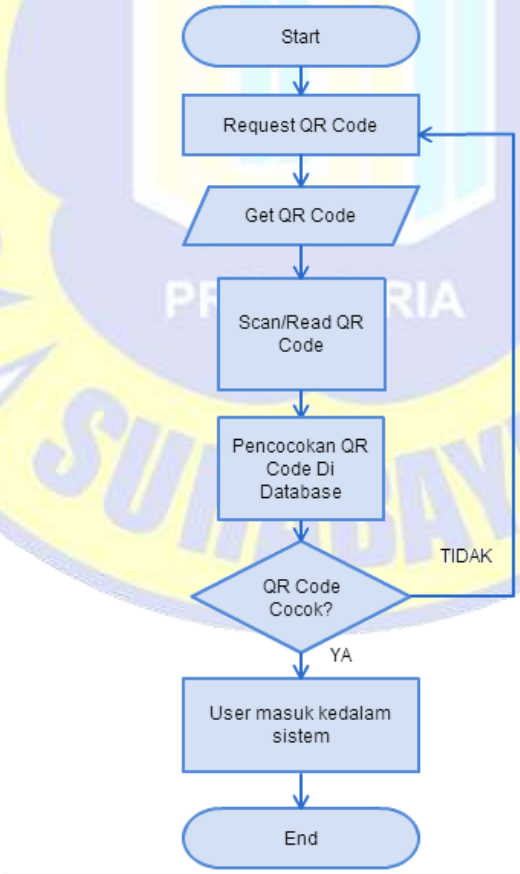
Gambar 4.1 *Tahap Registrasi*

2. Tahap Autentikasi

Yang kedua adalah tahap autentikasi, Setelah proses registrasi selesai *user* dapat melakukan *request QR code* yang nantinya akan digunakan pada saat login

di *website* yang telah terdapat data *user*, yaitu *username* dan *password* pada database. Dengan cara *user* harus memindai *QR code* yang telah *digenerate* tersebut melalui kamera *webcam*. Sistem kemudian melakukan pencocokan *QR code* pada database.

Apabila *QR code* tersebut cocok, maka pengguna dapat langsung masuk kedalam sistem. Apabila *QR code* tersebut tidak cocok maka pengguna diharuskan untuk melakukan *request* ulang *QR code*. Dan tentunya *QR code* yang didapat nantinya berbeda dengan *QR code* sebelumnya, dikarenakan sistem hanya memperbolehkan satu kali pemakaian *QR code* saja untuk login. Alur kerja proses autentikasi dapat dilihat pada gambar 4.2.



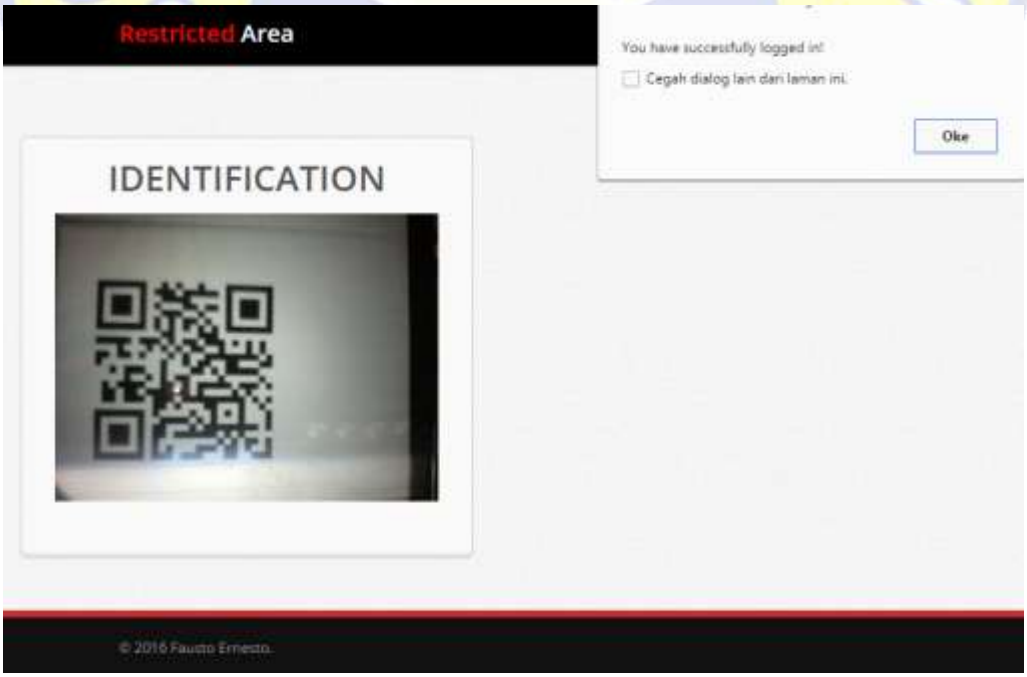
Gambar 4.2 *Tahap Autentikasi*

4.3 Perancangan Antarmuka

Implementasi tampilan/interface untuk aplikasi ini bertujuan untuk mempermudah user dalam berinteraksi dengan aplikasi yang akan dibuat. Pada pembuatan aplikasi ini terdiri dari beberapa bagian perancangan yaitu tampilan antarmuka website dan tampilan antarmuka pada perangkat android. Berikut merupakan rancangan desain yang akan dibuat beserta dengan penjelasan fungsinya.

1. Halaman Login Website

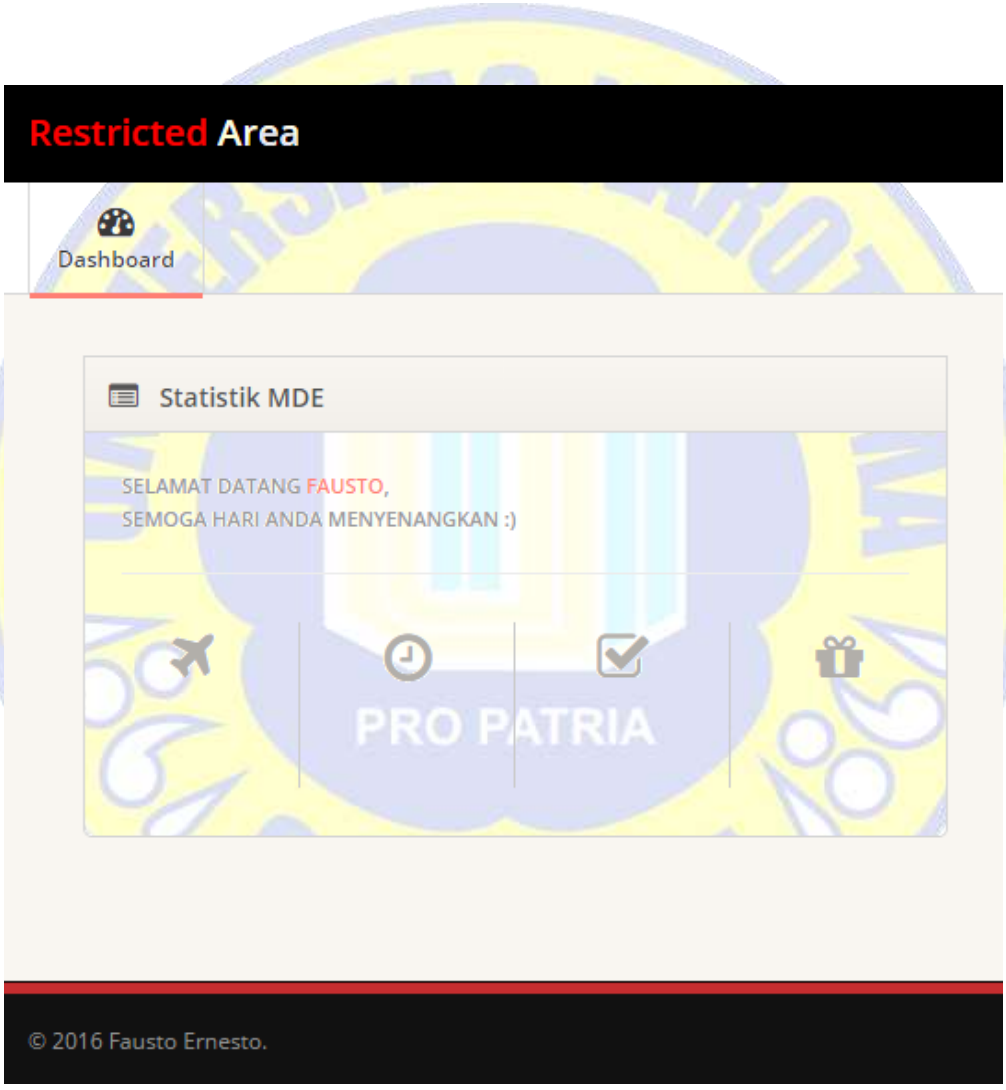
Pada tampilan halaman login terdapat kotak *identification.* Dimana pada bagian itu *QR code* yang dipindai oleh user akan diterjemahkan dan di identifikasi oleh aplikasi. Tampilan halaman login dapat dilihat pada gambar berikut.



Gambar 4.3 *Halaman Login Website*

2. *Home Dashboard Website*

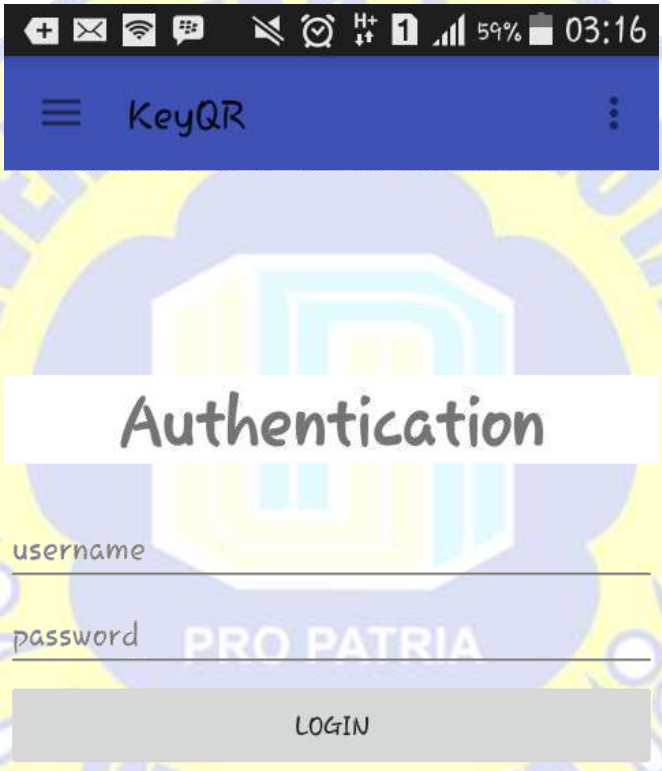
Merupakan halaman tampilan awal setelah user berhasil melakukan autentikasi pada halaman login. Home dashboard dapat dilhat pada gambar berikut.



Gambar 4.4 *Home Dashboard Website*

3. Halaman Login Android

Digunakan untuk membatasi akses terhadap aplikasi, setiap user mempunyai username dan password yang akan digunakan untuk masuk ke aplikasi dapat dilihat pada gambar berikut.



Gambar 4.5 *Halaman Login Android*

4. Form Registrasi

Digunakan untuk mendaftarkan diri menjadi bagian dari sistem untuk memperoleh hak akses terhadap aplikasi yang akan digunakan. Dengan melengkapi syarat pendaftran berupa *username*, *password*, *full name*, dan *email*. Form registrasi dapat dilihat pada gambar berikut.

Gambar 4.6 *Form Registrasi*

5. *Home Dashboard Android*

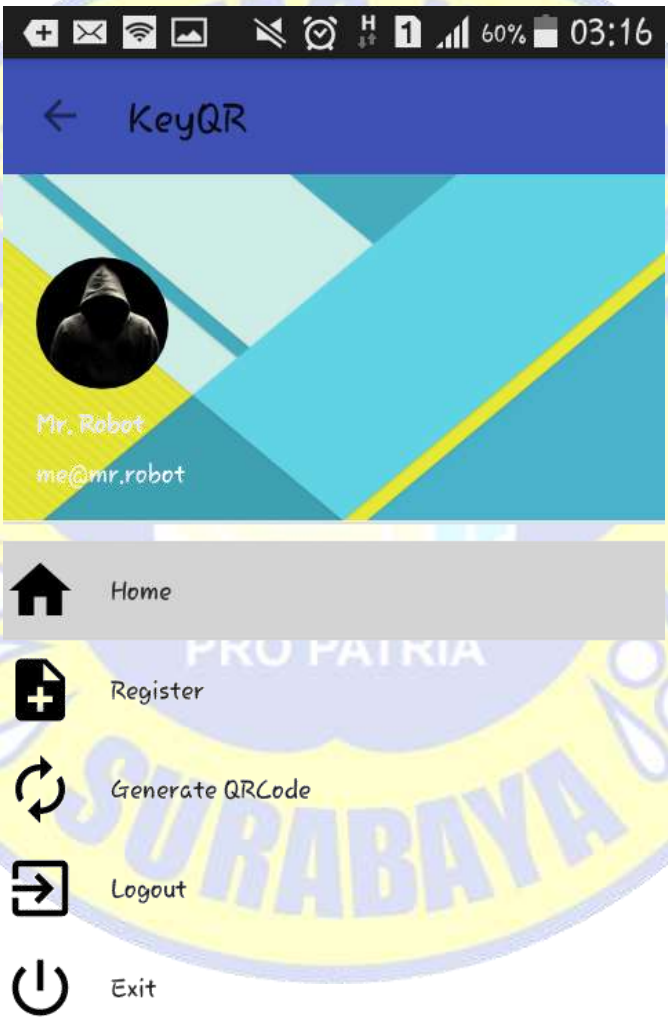
Merupakan halaman tampilan awal setelah user berhasil melakukan autentikasi pada halaman login. Home dashboard android dapat dilihat pada gambar berikut.

Selamat Datang

Gambar 4.7 *Home Dashboard Android*

6. *User Menu*

Pada halaman ini tersedia tampilan menu yang dapat diakses oleh user. Pilihan menu yang tersedia dapat digunakan oleh user yang memiliki hak akses terhadap aplikasi. *User menu* dapat dilihat pada gambar berikut.



Gambar 4.8 *User Menu*

7. Hasil *Generate QR Code*

Hasil *generate QR Code* dapat dilihat pada gambar berikut dimana aplikasi menampilkan *QR Code* yang telah *digenerate* dari database. QR Code yang diterima user telah dimuat didalamnya *username* dan *password* milik user dan siap untuk digunakan.

Gambar 4.9 *Hasil Generate QR Code*

4.4 Pengujian Sistem

Setelah tahapan implementasi dilakukan dan menghasilkan aplikasi autentikasi login QR Code dengan memanfaatkan teknologi android, maka dilanjutkan dengan tahapan pengujian. Tahap ini bertujuan untuk menguji kesiapan sistem, Berdasarkan tabel hasil pengujian aplikasi yang telah dilakukan diperoleh hasil sebagai berikut:

Tipe	Spesifikasi	Pengujian	Hasil
Samsung Galaxy Core 2 SM-G355H	OS Android KitKat 4.4.2	Pengujian login dengan username dan password terenkripsi berwujud QR Code	Login Berhasil
		Akses menu dashboard website dan android	Akses Berhasil
		Generate QR Code dari android	Generate Sukses
		Menu logout, keluar dari website dan aplikasi.	Berhasil

Tabel 4.1 *Hasil Pengujian Aplikasi*

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan pembahasan yang terdapat pada bab-bab sebelumnya, maka dapat disimpulkan bahwa Pengembangan Prototipe *QR Code* Sebagai Autentikasi Keamanan Login Sistem Dengan Memanfaatkan Teknologi Android ini dapat:

1. Mempermudah pengguna dalam melakukan login pada sistem.
2. Dapat dikembangkan ke depan untuk proses autentikasi aplikasi yang membutuhkan PIN atau token.
3. QR Code dapat digunakan sebagai sistem login yang praktis dan aman karena tidak mudah dihafal/ditebak dan berubah-ubah setiap waktu.

5.2 Saran

Pengembangan Prototipe *QR Code* Sebagai Autentikasi Keamanan Login Sistem Dengan Memanfaatkan Teknologi Android ini masih jauh dari kesempurnaan dan perlu dikembangkan lebih lanjut. Adapun saran untuk pengembangan aplikasi ini selanjutnya agar dapat dikembangkan sebagai alternatif lain metode autentikasi login pada *website* atau sistem. Selain menggunakan *username* dan *password* atau penggunaan token dan PIN.

DAFTAR PUSTAKA

Masdito Bachtiar, Ary Mazharuddin. (2012). *Smart Login Pada Situs Web Menggunakan QR-Code*. Surabaya: Teknik Informatika Institut Teknologi Sepuluh Nopember.

Frengky Tedy, (2013). *Pengembangan Aplikasi Ticketing Berbasis QR Code Dengan Data Terenkripsi Untuk Stadion Utama Gelora Bung Karno*. Yogyakarta: Teknik Informatika Universitas Atma Jaya.

Meier. (2010). *Professional Android 2 Application Development*. California: Wiley.

Gargenta. (2011). *Learning Android*. California: O'Rilley Media.

Rosari, R.W. (2008). *PHP Dan MYSQL Untuk Pemula*. Yogyakarta: ANDI.

Rouillard. (2008). *Multimodality in Mobile Computing and Mobile Devices. Hershey, New York: IGI Global.*

Agus Saputra. (2011). *Trik Dan Solusi Jitu Pemrograman PHP*. Jakarta: Elex Media Komputindo.

Ariadi. (2011). *Analisis dan Perancangan Kode Matriks Dua Dimensi Quick Response QR Code*. Medan: Universitas Sumatera Utara.

Nababan. (2011). *Studi Perbandingan Antara Metode Probabilistic Encryption dengan Metode Rivest Shamir Adleman*. Medan: Universitas Sumatra Utara.

M. Salahuddin dan Rosa, (2010). *Pemograman J2ME Belajar Cepat Pemograman Perangkat Telekomunikasi Mobile*. Bandung: Informatika.

Aghus Sofwan, Agung Budi P, Toni Susanto. (2006). *Aplikasi Kriptografi Dengan Algoritma Message Digest 5 (MD5).* Semarang*:* Universitas Diponegoro.

LAMPIRAN

AUTHENTICATION SCRIPT

Merupakan *code*/*script* untuk menjalankan proses *authentication*.

```
<?php require_once("mod.inc.php");

/** Clean input **/
$clean      = new GUMP();
$kripto = new Crypton();
$data = array();
$_POST      = $clean->sanitize($_POST);
extract($_POST);

/** If QRCode **/
if(isset($send) && $send=="qrcode"){

      /** Query Insert DB **/
      $stm  = "SELECT
                  t.*
                  FROM
                    qr_users t
                  WHERE user_qrcode=?
                  AND TIMESTAMPDIFF(MINUTE,t.user_exptime,NOW())
<= 1";
      $exec = $db->GetRow($stm,array($credential));

      if($exec){
            session_regenerate_id(TRUE);
            $_SESSION["login"]            = "qrcode";
```

```
            $_SESSION["user_id"]    = $exec['user_id'];
            $_SESSION["user_name"]  = $exec['user_name'];

            /** Generate new qrcode **/
            $time = str_replace(".","",microtime(true));
            $user = $exec['user_name'];
            $qTxt = $kripto->myCrypt($user,$time);

            /** Update qrtext in db **/
            $stm  = "UPDATE qr_users SET user_qrcode='$qTxt',
user_exptime=NOW() WHERE user_id='" . $exec['user_id'] . "'";
            $exec = $db->Execute($stm);

            $data['AUTH'] = TRUE;
      }else{
            $data['AUTH'] = FALSE;
      }

      header('Content-Type: application/json');
      print(json_encode($data));
      exit;

}else{
      $data['AUTH'] = FALSE;
      header('Content-Type: application/json');
      print(json_encode($data));
      exit;
}

?>
```

KICKASS SCRIPT

Merupakan *script*/*code* yang digunakan untuk logout dari sistem.

```
<?php session_start();

if(isset($_POST['kick'])) {
	unset($_SESSION['login']);
	session_destroy();
	session_regenerate_id(TRUE);
	header('Content-Type: application/json');
	print(json_encode(array('AUTH'=>'kick')));
	exit;
}

?>
```

MOD INC SCRIPT

Digunakan untuk pendukung library.

```
<?php session_start();
require_once("../config/config.inc.php");

/** Create Admin Base Path **/
$dir        = explode("\\", __DIR__);
$last_dir   = NULL;

/** Set Admin Path **/
require_once("../libs/adodb5/tohtml.inc.php");
require_once("../libs/adodb5/adodb.inc.php");
require_once("../config/conn.inc.php");
require_once("../libs/gump/gump.class.php");
require_once("../libs/crypto/Crypto.class.php");

?>
```

WEB API SCRIPT

Untuk menerima dan memproses *request* dari *client* (Aplikasi Android)

```
<?php  if ( ! defined('ADMPATH')) exit('No direct script access
allowed');

/** Start output buffering **/

ob_start();


/** Include require library **/

require_once("libs/gump/gump.class.php");

require_once("libs/crypto/Crypto.class.php");

require_once("libs/Endroid/QrCode/QrCode.php");

require_once("libs/adodb5/tohtml.inc.php");

require_once("libs/adodb5/adodb.inc.php");

require_once("config/conn.inc.php");


/** Using Endroid Qrode **/

use Endroid\QrCode\QrCode;


/** Clean input **/

$clean      = new GUMP();

$_POST      = $clean->sanitize($_POST);

extract($_POST);


/** Get the requested action **/

$data       = array();

$act  = (isset($act)) ? $act:NULL;


switch ($act) {
```

```
	case 'getAuth':
		ob_get_clean();

		/** Check session/cookies **/
		if(isLoggedIn()) {
			$data = array("msg"=>"loggedin");
			//$data = array("COOKIE"=>$_COOKIE,
"SESSION"=>$_SESSION);
			header('Content-Type: application/json');
			print(json_encode($data));
			break;
		}

		/** Encrypt password **/
		$kripto = new Crypton();
		$pass	= $kripto-
>myCrypt($user_password,$user_name);

		/** Matching user & pass **/
		$stm	= "SELECT * FROM qr_users WHERE
user_name=? AND user_password=?";
		$exec	= $db-
>GetRow($stm,array($user_name,$pass));

		if($exec){
			$_SESSION["LOGIN"] = $exec['user_id'];
			setcookie("LOGIN", $exec['user_id'],
time()+3600*24*1, "/");
			session_regenerate_id(TRUE);
			$data = array("msg"=>"success");
		}else{
			$data = array("msg"=>"failed");
		}
```

```
            header('Content-Type: application/json');
            print(json_encode($data));
            break;

      case 'regUser':
            ob_get_clean();
            # Register new user
            $clean->validation_rules(array(
                  'user_name' =>
'required|alpha_numeric|max_len,100',
                  'user_password' =>
'required|alpha_space|max_len,255',
                  'user_fullname' =>
'required|alpha_space|max_len,255',
                  'user_email' => 'required|valid_email'
            ));

            /** Validation **/
            $ver = $clean->run($_POST);
            $msg = $clean->get_readable_errors(true);

            if($ver===false){
                  $data = explode("#", $msg);
                  $data =
array('msg'=>"failed","error"=>array_filter($data));
            }else{

                  /** Encrypt password **/
                  $kripto = new Crypton();
                  $pass       = $kripto-
>myCrypt($user_password,$user_name);
                  $time   = str_replace(".","",microtime(true));
                  $qTxt   = $kripto->myCrypt($user_name,$time);
```

```
                    /** Insert to db **/

                    $stm  = "INSERT INTO qr_users
(user_name,user_password,user_fullname,user_email,user_qrcode)

                                    VALUES
('$user_name','$pass','$user_fullname','$user_email','$qTxt')";

                    $exec = $db->Execute($stm);

                    if($exec){

                            $data = array("msg"=>"success");

                    }else{

                            $data = array("msg"=>"$stm");

                    }


            }


            header('Content-Type: application/json');

            print(json_encode($data));

            break;


        case 'regenQR':

            ob_get_clean();

            # Regenerate QRCode

            /** Check session/cookies **/

            if(isLoggedIn()) {


                    /** Get user data **/

                    $uid  = (isset($_COOKIE['LOGIN'])) ?
$_COOKIE['LOGIN']: $_SESSION['LOGIN'];

                    $stm        = "SELECT * FROM qr_users WHERE
user_id=?";

                    $exec       = $db->GetRow($stm,array($uid));


                    if($exec){
```

```
                        /** Generate new qrtext **/
                        $kripto = new Crypton();
                        $time =
str_replace(".","",microtime(true));
                        $user = $exec['user_name'];
                        $qTxt = $kripto->myCrypt($user,$time);


                        /** Update qrtext in db **/
                        $stm  = "UPDATE qr_users SET
user_qrcode='$qTxt', user_exptime=NOW() WHERE user_id='$uid'";
                        $exec = $db->Execute($stm);
                        $data = array("msg"=>"renew");
                        header('Content-Type: application/json');
                        print(json_encode($data));
                        break;
                  }

            }else{
                  $data = array("msg"=>"denied");
                  header('Content-Type: application/json');
                  print(json_encode($data));
                  break;
            }
            break;

      case 'kickAss':
            ob_get_clean();
            # Log Out
            kickAss();
            $data = array("msg"=>"out");
            header('Content-Type: application/json');
            print(json_encode($data));
```

```
            break;


      default:
            ob_get_clean();
            # Default action is show qrcode when loggedin
            if(isLoggedIn()) {
                  /** Get user data **/
                  $uid  = (isset($_COOKIE['LOGIN'])) ?
$_COOKIE['LOGIN']: $_SESSION['LOGIN'];
                  $stm        = "SELECT user_qrcode FROM qr_users
WHERE user_id=?";
                  $exec       = $db->GetRow($stm,array($uid));


                  if($exec){
                        /** Get qr_code from db **/
                        $qTxt = $exec['user_qrcode'];
                        header('Content-Type: image/png');
                        //header('Content-
Disposition:attachment;filename="qrcode.png"');
                        $qrCode = new QrCode();
                        $qrCode
                              ->setText($qTxt)
                              ->setSize(200)
                              ->setPadding(10)
                              ->setErrorCorrection('high')
                              ->setForegroundColor(array('r'=>0,
'g'=>0, 'b'=>0, 'a'=>0))
                              ->setBackgroundColor(array('r'=>255,
'g'=>255, 'b'=>255, 'a'=>0))
                              ->setLabelFontSize(16)
                              ->render();
                        break;
                  }
```

```
            }else{
                  header('Content-Type: application/json');
                  $data['msg'] = "Hii buddies, have a great day!";
                  print(json_encode($data));
                  break;
            }

            break;
}

function isLoggedIn(){
      if(isset($_SESSION['LOGIN']) || isset($_COOKIE['LOGIN']))
return TRUE;
      else return FALSE;
}

function kickAss(){
      session_destroy();
      setcookie("LOGIN","",time()-3600*24*1,"/");
      unset($_COOKIE["LOGIN"]);
      unset($_SESSION["LOGIN"]);
}
?>
```

DAFTAR PUSTAKA

Masdito Bachtiar, Ary Mazharuddin. (2012). *Smart Login Pada Situs Web Menggunakan QR-Code*. Surabaya: Teknik Informatika Institut Teknologi Sepuluh Nopember.

Frengky Tedy, (2013). *Pengembangan Aplikasi Ticketing Berbasis QR Code Dengan Data Terenkripsi Untuk Stadion Utama Gelora Bung Karno*. Yogyakarta: Teknik Informatika Universitas Atma Jaya.

Meier. (2010). *Professional Android 2 Application Development*. California: Wiley.

Gargenta. (2011). *Learning Android*. California: O'Rilley Media.

Rosari, R.W. (2008). *PHP Dan MYSQL Untuk Pemula*. Yogyakarta: ANDI.

Rouillard. (2008). *Multimodality in Mobile Computing and Mobile Devices. Hershey, New York: IGI Global.*

Agus Saputra. (2011). *Trik Dan Solusi Jitu Pemrograman PHP*. Jakarta: Elex Media Komputindo.

Ariadi. (2011). *Analisis dan Perancangan Kode Matriks Dua Dimensi Quick Response QR Code*. Medan: Universitas Sumatera Utara.

Nababan. (2011). *Studi Perbandingan Antara Metode Probabilistic Encryption dengan Metode Rivest Shamir Adleman.* Medan: Universitas Sumatra Utara.

M. Salahuddin dan Rosa, (2010). *Pemograman J2ME Belajar Cepat Pemograman Perangkat Telekomunikasi Mobile*. Bandung: Informatika.

Aghus Sofwan, Agung Budi P, Toni Susanto. (2006). *Aplikasi Kriptografi Dengan Algoritma Message Digest 5 (MD5).* Semarang*:* Universitas Diponegoro.