BAB VII

HASIL DAN PEMBAHASAN

Pada tahap ini implementasi dan pengujian sistem menjadi salah satu tahapan yang penting dalam pembuatan suatu sistem agar suatu program dapat berjalan sesuai dengan proses perencanaan yang sebelumnya sudah di lakukan.

4.1 Sistem Operasi Ubuntu Server

Sebelum membangun sebuah sistem monitoring file pada root directory web server, di butuhkan sistem operasi untuk menjalankan program bot yang sudah di buat untuk memonitoring dan juga di gunakan untuk tempat berjalannya aplikasi berbasis web untuk dashboard monitoring file.

4.2 Dependency / Modul Python 3

Agar program bot python dapat berjalan baik di butuhkan modul pyhon 3 untuk melengkapi reqruitment sistem, berikut modul yang di butuhkan untuk menjalankan bot monitoring file pada root directory web server.

- Python 3
- WatchDog
- Hash File
- Sql Lite 3
- Request

4.3 Apache Webserver & Maria DB

Agar dashboard berbasis Web UI monitoring file pada root web server dapat berjalan di butuhkan aplikasi webserver Apache & database MariaDB.

4.4 Konfigurasi Bot Monitor

Berikut merupakan cara lengkap untuk mengkonfigurasi bot monitoring file pada root web server.

Berikut source code python config pada bot.

```
import sys
import time
import datetime
import sqlite3
import hashlib
import requests

from watchdog.observers import Observer
from watchdog.events import FileSystemEventHandler
from sqlite3 import Error
from filehash import FileHash

token = '';

class Watcher:
    DIRECTORY_TO_WATCH = "/var/www/html "

    def __init__(self):
        self.observer = Observer()

    def run(self):
        event_handler = Handler()
        self.observer.schedule(event_handler,
self.DIRECTORY_TO_WATCH, recursive=True)
        self.observer.start()
        try:
            while True:
                time.sleep(5)
        except:
            self.observer.stop()
            print("Error")
```

```
        self.observer.join()

class Penyimpanan:
    def sql_connect():
        try:
            con = sqlite3.connect('logs.db')
            #print("Data Logs Sudah Terhubung")
            return con
        except Error:
            #print("Data Log, belum terhubung. sistem akan berjalan
dengan tidak menyimpan LOGS")
            print(Error)

    def crt_tbl(con):
        cObj = con.cursor()
        cObj.execute("create table if not exists m_logs (id integer
primary key, mod text not null, desc text not null, enk text not
null, tgl text not null)")
        con.commit()
        print("Data Logs Sudah Terhubung")

    def tambah(con, data):
        cObj = con.cursor()
        cObj.execute("insert into m_logs(mod, desc, enk, tgl)
values(?,?,?,?)", data)
        #cObj.execute("insert into m_logs (key, enk, tgl) values
(?,?,?)", mon, enk, det)
        con.commit()

class Pecah:
    def linear_search(item, my_list):
        found = False
        position = 0
        while position < len(my_list) and not found:
            if my_list[position] == item:
                found = True
            position = position + 1
        return found


class Handler(FileSystemEventHandler):

    @staticmethod
    def on_any_event(event):
        if event.is_directory:
            return None

        elif event.event_type == 'created':
            # Take any action here when a file is first created.
            currentDT = datetime.datetime.now()
            hd = FileHash('md5')
```

```
            out_text = event.src_path
            itemfound = Pecah.linear_search('swp',
out_text.split('.'))
            if itemfound:
                print("...")
            else:
                print("Created: {}".format(out_text))
                print("Hash: {}".format(hd.hash_file(out_text)))
                print("Time: {}-{}-{}
{}:{}:{}".format(currentDT.day, currentDT.month, currentDT.year,
currentDT.hour, currentDT.minute, currentDT.second))
                c = Penyimpanan.sql_connect()
                tgl = "{}-{}-{} {}:{}:{}".format(currentDT.day,
currentDT.month, currentDT.year, currentDT.hour, currentDT.minute,
currentDT.second)
                Penyimpanan.tambah(c,("CREATED", out_text,
hd.hash_file(out_text), tgl))
                data = {'token': token, 'mode': 'CREATED', 'mon':
str(out_text), 'enk': str(hd.hash_file(out_text))}
                req =
requests.post('http://192.168.1.11/bagus/push_data.php', data,
headers={'Content-Type': 'application/x-www-form-urlencoded'})

            #print("Created: {}".format(event.src_path))

        elif event.event_type == 'modified':
            # Taken any action here when a file is modified.
            currentDT = datetime.datetime.now()
            hd = FileHash('md5')
            out_text = event.src_path
            itemfound = Pecah.linear_search('swp',
out_text.split('.'))
            if itemfound:
                print("...")
            else:
                print("Modified: {}".format(out_text))
                print("Hash: {}".format(hd.hash_file(out_text)))
                print("Time: {}-{}-{}
{}:{}:{}".format(currentDT.day, currentDT.month, currentDT.year,
currentDT.hour, currentDT.minute, currentDT.second))
                #print(token)
                c = Penyimpanan.sql_connect()
                tgl = "{}-{}-{} {}:{}:{}".format(currentDT.day,
currentDT.month, currentDT.year, currentDT.hour, currentDT.minute,
currentDT.second)
                mon = '{}'.format(out_text)
                enk = '{}'.format(hd.hash_file(out_text))
                Penyimpanan.tambah(c,("MODIFIED", out_text,
hd.hash_file(out_text), tgl))
                data = {'token': token, 'mode': 'MODIFIED', 'mon':
str(out_text), 'enk': str(hd.hash_file(out_text))}
```

```
                req =
requests.post('http://192.168.1.11/bagus/push_data.php', data,
headers={'Content-Type': 'application/x-www-form-urlencoded'})
                #print(data)
                #print(req.status_code)
                #print(req.text)

            #print("Modified: {}".format(event.src_path))
            #print("Time: {}-{}-{} {}:{}:{}".format(currentDT.day,
currentDT.month, currentDT.year, currentDT.hour, currentDT.minute,
currentDT.second))
        elif event.event_type == 'deleted':
            # Taken any action here when a file is modified.
            currentDT = datetime.datetime.now()
            hd = hashlib.md5()
            out_text = event.src_path
            itemfound = Pecah.linear_search('swp',
out_text.split('.'))
            if itemfound:
                print("...")
            else:
                hd.update(b"Data telah dihapus")
                print("Deleted: {}".format(out_text))
                print("Hash: {}".format(hd.hexdigest()))
                print("Time: {}-{}-{}
{}:{}:{}".format(currentDT.day, currentDT.month, currentDT.year,
currentDT.hour, currentDT.minute, currentDT.second))
                c = Penyimpanan.sql_connect()
                tgl = "{}-{}-{} {}:{}:{}".format(currentDT.day,
currentDT.month, currentDT.year, currentDT.hour, currentDT.minute,
currentDT.second)
                Penyimpanan.tambah(c,("DELETED", out_text,
hd.hexdigest(), tgl))
                data = {'token': token, 'mode': 'DELETED', 'mon':
str(out_text), 'enk': str(hd.hexdigest())}
                req =
requests.post('http://192.168.1.11/bagus/push_data.php', data,
headers={'Content-Type': 'application/x-www-form-urlencoded'})


if __name__ == '__main__':
    val = input("Masukan token anda: ")
    print("Token anda: ")
    print(val)

    data = {'token': str(val)}
    req =
requests.post('http://192.168.1.11/bagus/check_token.php', data)

    print(req.status_code)
    #print(req.text)
```

```
    if int(req.text) > 0:
        print("Api ditemukan")
        token = str(val)
        w = Watcher()
        c = Penyimpanan.sql_connect()
        Penyimpanan.crt_tbl(c)
        w.run()
    else:
        print("Api tidak ditemukan, Mohon periksa kembali")
        sys.exit()
```

Berikut penjelasan source code diatas :

```
class Watcher:
    DIRECTORY_TO_WATCH = "/var/www/html "
```

Baris kode diatas merupakan konfigurasi dimana di mana berfungsi sebagai pengatur di rectory mana yang akan di monitoring, jadi untuk baris kode berikut bisa di sesuaikan dengan direktori mana yang akan monitoring.

```
elif event.event_type == 'created':
            # Take any action here when a file is first
created.
            currentDT = datetime.datetime.now()
            hd = FileHash('md5')
            out_text = event.src_path
            itemfound = Pecah.linear_search('swp',
out_text.split('.'))
```

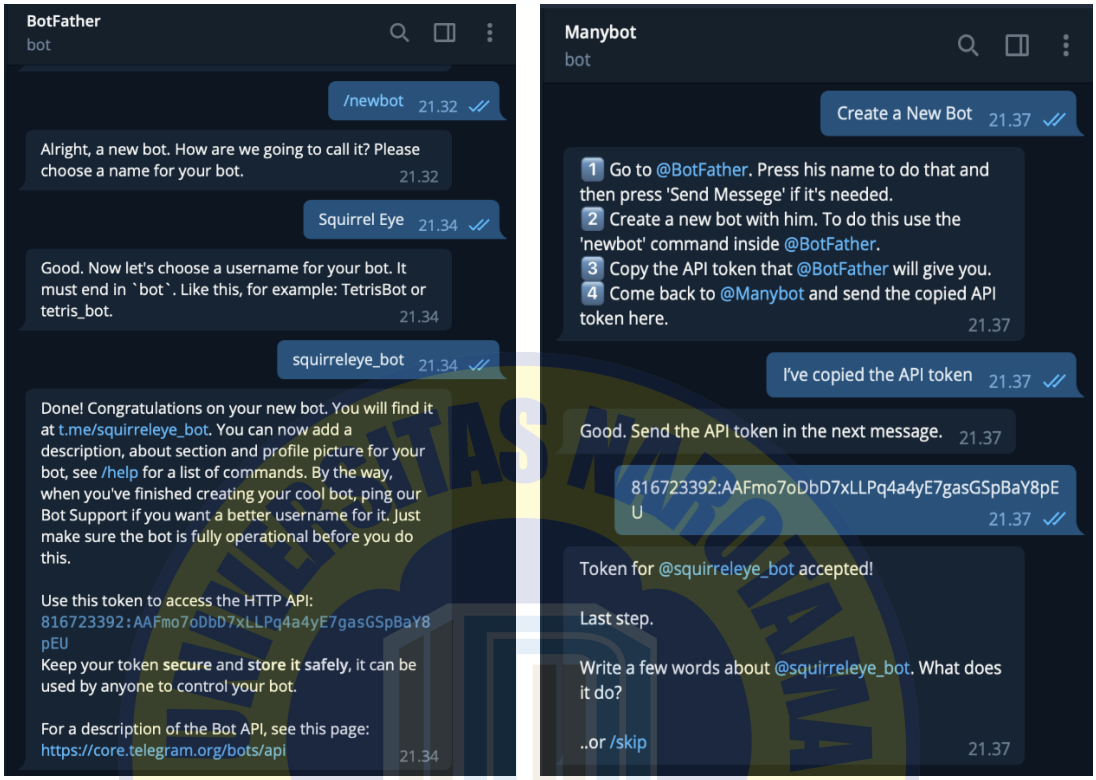
Baris kode diatas berfungsi sebagai penanda atau alert jika ada file Created, Modified, & Deleted, skrip diatas juga berfungsi sebagai generator hash pada seluruh file yang di monitoring.

```
requests.post('http://192.168.1.11/bagus/push_data.php', data,
headers={'Content-Type': 'application/x-www-form-urlencoded'})
```

Source code diatas merupakan baris kode yang berfungsi sebagai penghubung & pengirim data dari server bot ke server dashboard aplikasi berbasis web.

4.5 Konfigurasi BOT pada telegram

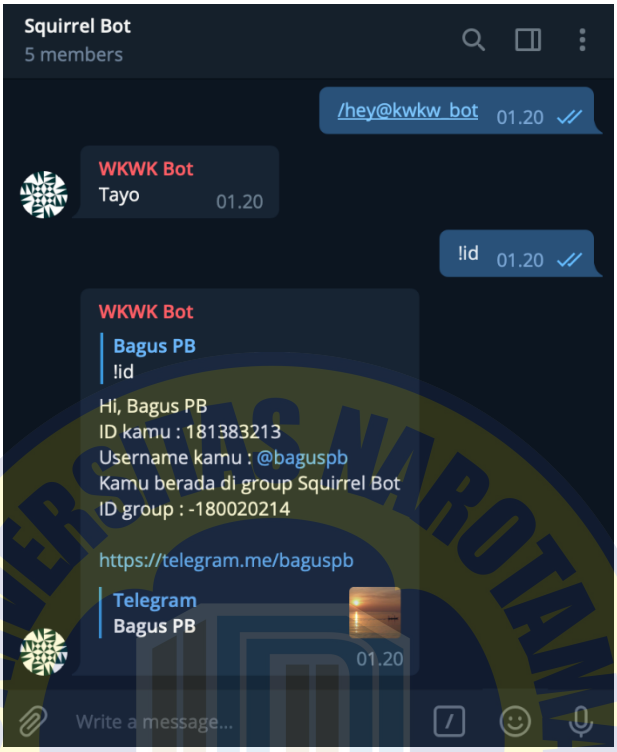
Agar program monitoring dapat memberikan inputan pada telegram maka di perlukan pembuatan bot baru, untuk registrasi dapat di lakukan di BotFather & Manybot untuk mendapatkan API Key, API key di gunakan untuk komunikasi antara dashboard webmonitoring dengan telegram notification. Berikut merupakan cara lengkap pembuatan bot pada telegram.



Gambar 4. 1 Generate token dari Manybot & BotFather

4.5.1 Konfigurasi mendapatkan id telegram & id group

Berikut merupakan mendapatkan id telegram atau id group menggunakan layanan bot @WKWK_Bot yang di gunakan untuk mendapatkan ID akun, ID akun ini berfungsi menghubungkan API Key Bot yang sudah di buat dari BotFather dan ManyBot dengan akun telegram yang akan di gunakan sebagai pusat monitoring..

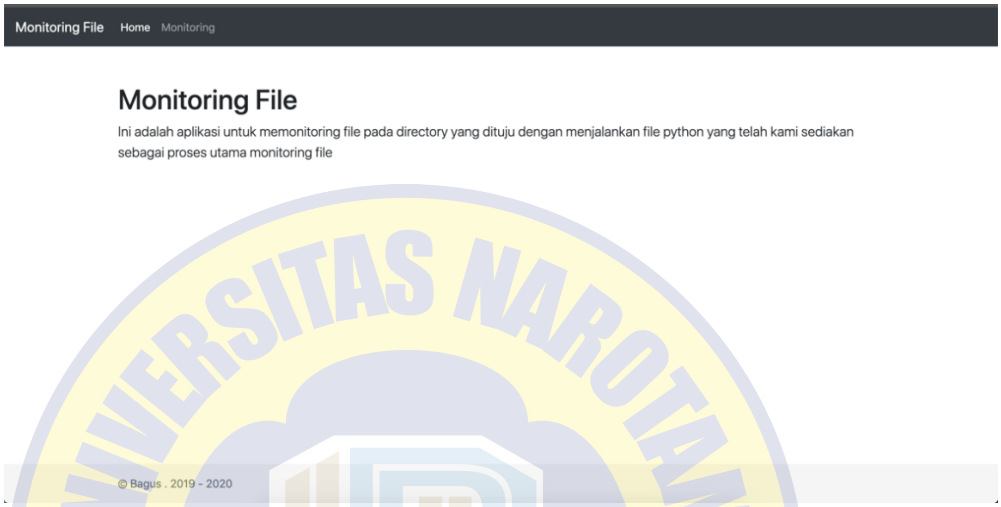


Gambar 4.5.1 Generate token id akun dan group

Gambar 4. 2 Notifikasi terhubung antara bot dengan server

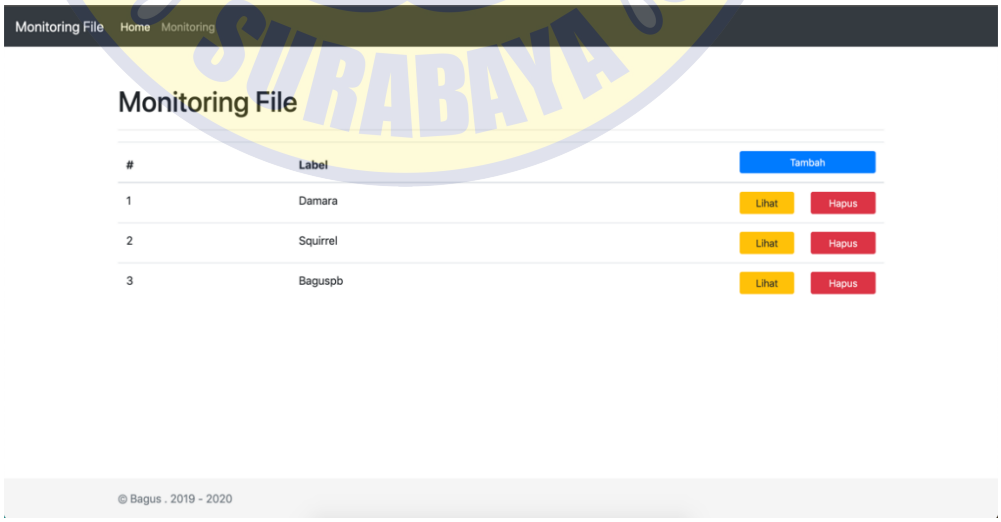
4.6 Konfigurasi tampilan antarmuka bot monitoring file

1. Halaman Utama



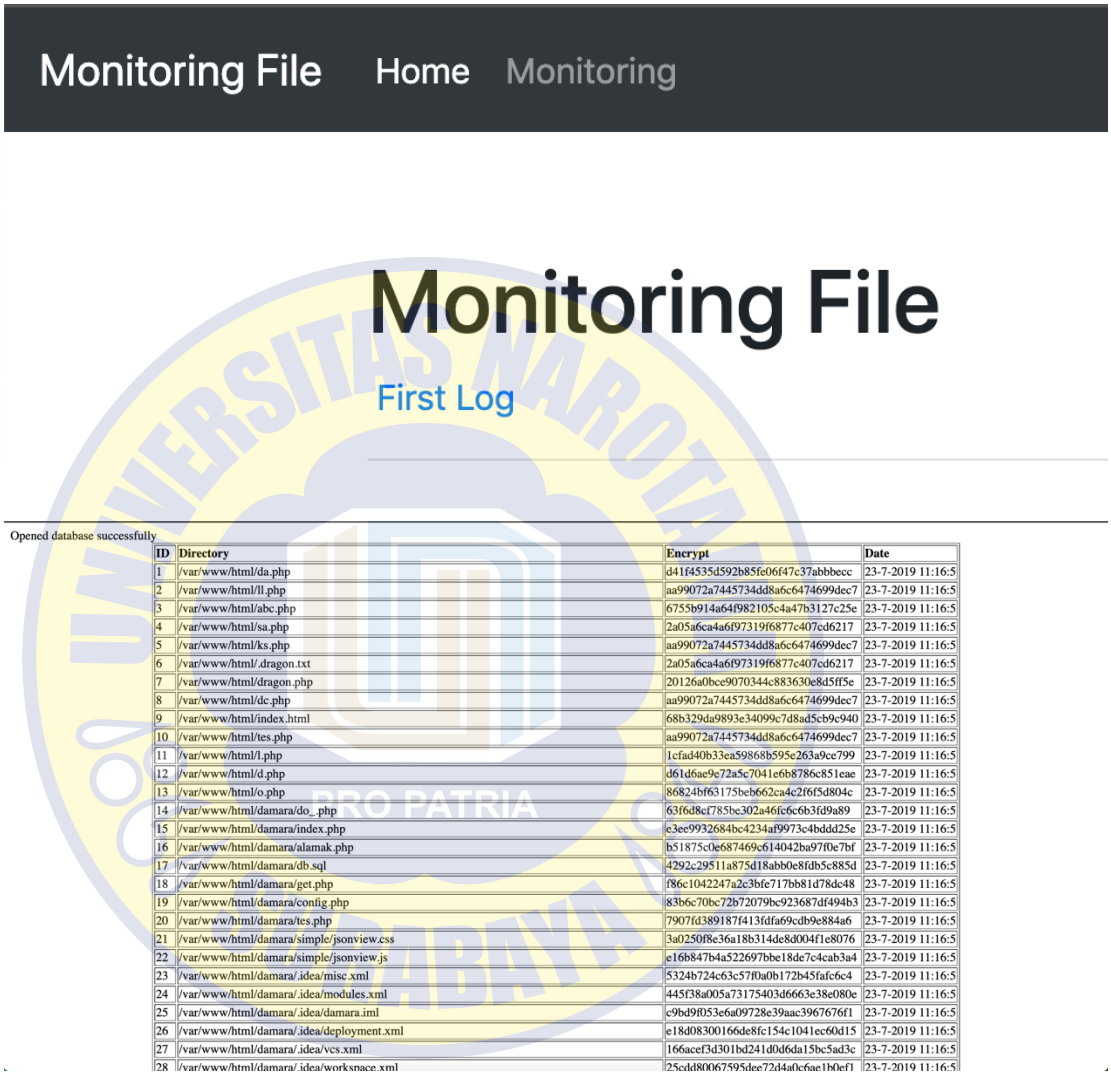
Gambar 4. 3 Halaman utama

2. Halaman manajemen label & token telegram



Gambar 4. 4 Generate token akun dan group

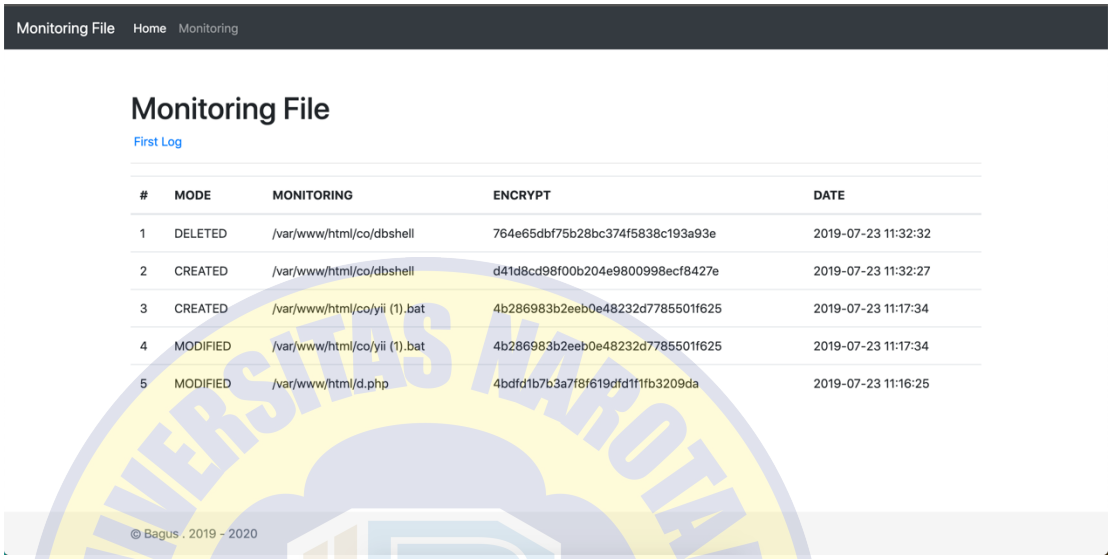
3. Halaman log awal



ID	Directory	Encrypt	Date
1	/var/www/html/da.php	d41f4535d592b85fe06f47c37abbbecc	23-7-2019 11:16:5
2	/var/www/html/ll.php	aa99072a7445734dd8a6c6474699dec7	23-7-2019 11:16:5
3	/var/www/html/abc.php	6755b914a64f982105c4a47b3127c25e	23-7-2019 11:16:5
4	/var/www/html/sa.php	2a05a6ca4a6f97319f6877c407cd6217	23-7-2019 11:16:5
5	/var/www/html/ks.php	aa99072a7445734dd8a6c6474699dec7	23-7-2019 11:16:5
6	/var/www/html/.dragon.txt	2a05a6ca4a6f97319f6877c407cd6217	23-7-2019 11:16:5
7	/var/www/html/dragon.php	20126a0bce9070344c883630e8d5ff5e	23-7-2019 11:16:5
8	/var/www/html/dc.php	aa99072a7445734dd8a6c6474699dec7	23-7-2019 11:16:5
9	/var/www/html/index.html	68b329da9893e34099c7d8ad5cb9c940	23-7-2019 11:16:5
10	/var/www/html/tes.php	aa99072a7445734dd8a6c6474699dec7	23-7-2019 11:16:5
11	/var/www/html/l.php	1cfad40b33ea59868b595e263a9ce799	23-7-2019 11:16:5
12	/var/www/html/d.php	d61d6ae9e72a5c7041e6b8786c851eae	23-7-2019 11:16:5
13	/var/www/html/o.php	86824bf63175beb662ca4c2f6f5d804c	23-7-2019 11:16:5
14	/var/www/html/damara/do_.php	63f6d8cf785bc302a46fc6c6b3fd9a89	23-7-2019 11:16:5
15	/var/www/html/damara/index.php	e3ee9932684bc4234af9973c4bddd25e	23-7-2019 11:16:5
16	/var/www/html/damara/alamak.php	b51875c0e687469c614042ba97f0e7bf	23-7-2019 11:16:5
17	/var/www/html/damara/db.sql	4292c29511a875d18abb0e8fdb5c885d	23-7-2019 11:16:5
18	/var/www/html/damara/get.php	f86c1042247a2c3bfe717bb81d78dc48	23-7-2019 11:16:5
19	/var/www/html/damara/config.php	83b6c70bc72b72079bc923687df494b3	23-7-2019 11:16:5
20	/var/www/html/damara/tes.php	7907fd389187f413fdfa69cdb9e884a6	23-7-2019 11:16:5
21	/var/www/html/damara/simple/jsonview.css	3a0250f8e36a18b314de8d004f1e8076	23-7-2019 11:16:5
22	/var/www/html/damara/simple/jsonview.js	e16b847b4a522697bbe18de7c4cab3a4	23-7-2019 11:16:5
23	/var/www/html/damara/.idea/misc.xml	5324b724c63c57f0a0b172b45fafc6c4	23-7-2019 11:16:5
24	/var/www/html/damara/.idea/modules.xml	445f38a005a73175403d6663e38e080e	23-7-2019 11:16:5
25	/var/www/html/damara/.idea/damara.iml	c9bd9f053e6a09728e39aac3967676f1	23-7-2019 11:16:5
26	/var/www/html/damara/.idea/deployment.xml	e18d08300166de8fc154c1041ec60d15	23-7-2019 11:16:5
27	/var/www/html/damara/.idea/vcs.xml	166acef3d301bd241d0d6da15bc5ad3c	23-7-2019 11:16:5
28	/var/www/html/damara/.idea/workspace.xml	25cdd80067595dee72d4a0c6ae1b0ef1	23-7-2019 11:16:5

Gambar 4. 5 Halaman Log hash awal

4. Halaman log perubahan file



Gambar 4. 6 Halaman perubahan file