

**APLIKASI VERIFIKASI KEPULANGAN WARGA BINAAN  
PEMASYARAKATAN MENGGUNAKAN ALGORITMA KNUTH  
MORRIS PRATT**

Buku ini dibuat untuk memenuhi persyaratan kelulusan  
matakuliah Program Internship I



**Dibuat Oleh,  
1.16.4.031 Ajis Trigunawan**

**PROGRAM DIPLOMA IV TEKNIK INFORMATIKA  
POLITEKNIK POS INDONESIA  
BANDUNG  
2019**

## **KATA PENGANTAR**

Puji syukur kehadirat Allah SWT yang telah memberikan rahmat, hidayah serta kekuatan sehingga dapat menyelesaikan Buku dengan judul “Aplikasi Verifikasi Kepulangan Warga Binaan Pemasyarakatan Menggunakan Algoritma Knuth Morris Pratt”.

Banyak kendala yang dihadapi dalam penyusunan Buku ini dan penulis menyadari bahwa penyusunan laporan ini masih belum sempurna. Ini mengingat keterbatasan pengetahuan, pengalaman serta kemampuan penulis. Penulis megharapkan kritik dan saran yang sifatnya membangun dari pembaca. Untuk itu penulis mengucapkan terima kasih kepada:

1. Allah SWT, karena dengan Rahmat dan Ridho-Nya penulis dapat menyelesaikan laporan intership I.
2. Kedua orang tua dan keluarga penulis yang telah mendorong dan memberi semangat kepada penulis.
3. Syaeful Bachri selaku pembimbing external di Rumah Tahanan Negara Klas 1 Bandung Kebonwaru.
4. Woro Isti Rahayu, S.T., M.T. selaku dosen pembimbing Interhsip 1.
5. Nisa Hanum Harani, S.Kom., M.T.selaku Koordinator Internship I.
6. M. Yusril Helmi Setyawan, S.Kom., M.Kom. selaku Ketua Prodi DIV Teknik Informatika.

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga buku ini membawa manfaat bagi pengembangan ilmu.

Bandung, 17 Januari 2020

Penulis

## DAFTAR ISI

---

DAFTAR ISI.....	i
DAFTAR GAMBAR.....	iv
DAFTAR TABEL.....	viii
BAB I.....	1
PENGENALAN SUBLIME, XAMPP, MYSQL, PHP DAN DATABASE1	
1.1    PENGANTAR SUBLIME TEXT .....	1
1.2    Pengantar XAMPP .....	1
1.3    Pengantar MySql.....	8
1.4    Pengantar PHP .....	22
1.5    Database.....	28
BAB II.....	37
PENGENALAN STRING MATCHING, ALGORITMA KNUTH MORRIS PRATT, WATERFALL, DAN CODEIGNITER .....	37
2.1    Pengantar String Matching.....	37
2.2    Pengantar Algoritma Knuth Morris Pratt(KMP).....	38
2.3    Metode Waterfall .....	43
2.4    Pengantar Codeigniter.....	46
BAB III .....	50
PENJELASAN TOOLS DAN BAHASA PEMOGRAMAN YANG DIGUNAKAN.....	50
3.1    Tools Yang Digunakan .....	50
BAB IV .....	59
INSTALASI TOOLS YANG DIGUNAKAN .....	59
4.1    Tools Yang Digunakan .....	59
BAB V .....	72
TUTORIAL PEMBUATAN APLIKASI.....	72
5.1    Tutorial Pembuatan Aplikasi .....	72
5.1.1    Mempersiapkan Tools.....	72

5.1.2	Membuat Database.....	73
5.1.3	Konfigurasi .....	78
5.1.4	Konfigurasi Template CSS Boostrap .....	84
5.1.5	Memasukan Template Kedalam Codingan View Admin .....	85
5.1.6	Source Code Admin Pertanyaan .....	93
5.1.7	Source Code Edit Pertanyaan.....	101
5.1.8	Memasukan Template Kedalam codingan view Menu Utama ....	104
5.1.9	Sorce Code From Input Untuk Verifikasi .....	109
5.1.10	Souce Code Lihat Konfirmasi.....	121
5.1.11	Source Code Sudah Di Konfirmasi .....	130
5.1.12	Source Code Login.....	141
5.1.13	Source Code Kategori Admin .....	147
5.1.14	Source Code Edit Kategori .....	154
5.1.15	Bootstrap Top Menu Petugas Tahanan Masuk .....	157
5.1.16	Source Code Tampilan Data Tahanan.....	159
5.1.17	Source Code Input Data Tahanan .....	171
5.1.18	From Input Jawaban.....	182
5.1.19	Source Code Edit Data Tahanan .....	185
BAB VI .....	192	
PENERAPAN ALGORITMA KNUTH MORRIS PRATT .....	192	
6.1	Alasan Menggunakan Algoritma Knuth Morris Pratt .....	192
6.2	Simulasi Algoritma Knuth Morris Pratt.....	192
BAB VII.....	198	
TAMPILAN ANTARMUKA.....	198	
7.1	Halaman Menu Utama .....	198
7.2	Halaman Data Tahanan .....	198
7.3	Halaman Login.....	199
7.4	Halaman Konfirmasi .....	200

7.5	Halaman Data User .....	200
BAB VIII .....		201
	KESIMPULAN.....	201
8.1	Kesimpulan .....	201
8.2	Kesimpulan Metode .....	201
8.3	Kesimpulan Pengujian Sistem .....	202

## **DAFTAR GAMBAR**

---

Gambar 1. 1 MySql.....	18
Gambar 1. 2 Contoh Penggunaan HTML .....	26
Gambar 1. 3 Contoh Penggunaan PHP .....	27
Gambar 2. 1 Metode Waterfall.....	44
Gambar 2. 2 Codeigniter.....	46
Gambar 2. 3 Perbandingan PHP Biasa Dengan CI .....	49
Gambar 3. 1 Sublime Text.....	51
Gambar 3. 2 Konsep Aliran MVC .....	55
Gambar 3. 3 Alur Kerja Codeigniter.....	55
Gambar 3. 4 Model .....	56
Gambar 3. 5 View .....	57
Gambar 3. 6 Controller .....	58
Gambar 4. 1 Downlod file.exe.....	59
Gambar 4. 2 Double Klik File .exe .....	60
Gambar 4. 3 Proses Instalasi Sublime.....	60
Gambar 4. 4 Proses Instalasi Sublime.....	61
Gambar 4. 5 Proses Instalasi Sublime.....	61
Gambar 4. 6 Proses Instalasi Sublime.....	62
Gambar 4. 7 Proses Instalasi Sublime.....	62
Gambar 4. 8 Instalasi Xampp.....	63
Gambar 4. 9 langkah instalasi .....	63
Gambar 4. 10 Langkah Instalasi .....	64
Gambar 4. 11 Langkah Instalasi .....	64
Gambar 4. 12 Menetukan lokasi folder xampp.....	65
Gambar 4. 13 Instalasi Selesai .....	66
Gambar 4. 14 Jendela dialog.....	66
Gambar 4. 15 klik tombol Start.....	67
Gambar 4. 16 Menjalankan Xampp .....	68

Gambar 4. 17 Xampp dijalankan .....	68
Gambar 4. 18 Downlod File Codeigniter.....	69
Gambar 4. 19 Extract file Codeigniter .....	69
Gambar 4. 20 Extract file Codeigniter .....	70
Gambar 4. 21 Hasil Extract file Codeigniter.....	70
Gambar 4. 22 Menjalankan Codeigniter .....	71
Gambar 5. 1 Xampp.....	73
Gambar 5. 2 PhpMyAdmin.....	74
Gambar 5. 3 Membuat Database.....	74
Gambar 5. 4 Membuat Database datatahanan.....	75
Gambar 5. 5 Databse datatahanan.....	75
Gambar 5. 6 Struktur Tabel Claim.....	76
Gambar 5. 7 Struktur Tabel Data Tahanan .....	76
Gambar 5. 8 Struktur Tabel Define Pasal .....	76
Gambar 5. 9 Struktur Tabel Jawaban.....	77
Gambar 5. 10 Struktur Tabel Jawaban Post.....	77
Gambar 5. 11 Struktur Tabel Pasal .....	77
Gambar 5. 12 Struktur Tabel Pertanyaan.....	77
Gambar 5. 13 Struktur Tabel User .....	77
Gambar 5. 14 Langkah Pertama.....	78
Gambar 5. 15 Langkah Kedua .....	79
Gambar 5. 16 Langkah ketiga.....	79
Gambar 5. 17 Langkah Keempat .....	80
Gambar 5. 18 Hasil Pemanggilan .....	81
Gambar 5. 19 Langkah Kelima .....	82
Gambar 5. 20 Langkah Keenam .....	82
Gambar 5. 21 Langkah Ketujuh .....	83
Gambar 5. 22 Langkah Kedepalan.....	84
Gambar 5. 23 Download Template .....	85
Gambar 5. 24 Extract Bootsrap.....	85

Gambar 5. 25 Heading .....	86
Gambar 5. 26 Sidebar Admin .....	88
Gambar 5. 27 Halaman Utama Admin.....	90
Gambar 5. 28 From Pertanyaan .....	93
Gambar 5. 29 From Edit Pertanyaan.....	101
Gambar 5. 30 Halaman Utama.....	105
Gambar 5. 31 Header Halaman Utama .....	107
Gambar 5. 32 Top Menu User Kepulangan .....	107
Gambar 5. 33 From Input Verifikasi.....	109
Gambar 5. 34 From Lihat Konfirmasi .....	121
Gambar 5. 35 From Sudah Di Konfirmasi.....	130
Gambar 5. 36 Halaman Login.....	141
Gambar 5. 37 Kategori.....	147
Gambar 5. 38 From Edit Kategori Pasal .....	154
Gambar 5. 39 Top Menu Petugas Tahanan Masuk .....	157
Gambar 5. 40 Tampilan Data Tahanan .....	159
Gambar 5. 41 From Input Tahanan .....	171
Gambar 5. 42 From Input Jawaban.....	182
Gambar 5. 43 Edit Data Tahanan.....	185
Gambar 6. 1 Algoritma Knuth Morris Pratt.....	192
Gambar 6. 2 Input Data Identitas Tahanan .....	195
Gambar 6. 3 Hasil Pencocokan Algoritma Knuth Morris Pratt .....	196
Gambar 6. 4 Data Tahanan Yang Sudah Di Konfirmasi.....	196
Gambar 7. 1 Tampilan Antarmuka Halaman Menu Utama.....	198
Gambar 7. 2 Tampilan Antarmuka Halaman Data Tahanan.....	198
Gambar 7. 3 Tampilan Antarmuka Halaman Login.....	199
Gambar 7. 4 Tampilan Antarmuka Halaman konfirmasi .....	200
Gambar 7. 5 Tampilan Antarmuka Halaman Data User .....	200
Gambar 8. 1 From Input Identitas Untuk Pencocokan.....	202

Gambar 8. 2 From Hasil Pencocokan .....	203
Gambar 8. 3 From Yang Di Konfirmasi .....	203
Gambar 8. 4 Data Base Data Tahanan .....	203

## **DAFTAR TABEL**

---

Tabel 2. 1 Pergeseran Pertama.....	41
Tabel 2. 2 Pergeseran Kedua .....	41
Tabel 2. 3 Pergeseran Ketiga .....	41
Tabel 2. 4 Pergeseran Keempat.....	42
Tabel 2. 5 Pergeseran Kelima .....	42
Tabel 2. 6 Pergeseran Keenam.....	42
Tabel 2. 7 Pergeseran Ketujuh .....	43

# BAB I

---

## PENGENALAN SUBLIME, XAMPP, MYSQL, PHP DAN DATABASE

Sublime Text, Xampp, Mysql, PHP Dan Database adalah tool yang membantu untuk membuat sebuah aplikasi. Pada buku ini penulis akan memaparkan materi tentang tools tersebut.

### 1.1 PENGANTAR SUBLIME TEXT

#### 1.1.1 Definisi Sublime Text

*Sublime Text* adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai *platform* operating system dengan menggunakan teknologi *Python API*. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, Aplikasi ini sangatlah fleksibel dan powerfull. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan sublime-*packages*. Sublime *Text* bukanlah aplikasi *opensource* dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, akan tetapi beberapa fitur pengembangan fungsionalitas (*packages*) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki linsensi aplikasi gratis.

### 1.2 Pengantar XAMPP

#### 1.2.1 Definisi XAMPP

*XAMPP* ialah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program. Yang mempunyai fungsi sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri dari program *MySQL Database*, *Apache HTTP Server*, dan penerjemah ditulis dalam bahasa pemrograman *PHP* dan *Perl*.

XAMPP adalah sebuah paket perangkat lunak (software) komputer yang sistem penamaannya diambil dari akronim kata Apache, MySQL (dulu) / MariaDB (sekarang), PHP, dan Perl. Sementara imbuhan huruf “X” yang terdapat pada awal kata berasal dari istilah *cross platform* sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti OS Linux, OS Windows, Mac OS, dan juga Solaris.

Sesuai dengan namanya software yang satu ini merupakan gabungan dari beberapa software dengan fungsi yang sama yakni menunjang para pembuat web yang menginginkan adanya web server sendiri di PC atau laptopnya. Software ini juga berlisensi GNU dan dapat didownload secara gratis di internet mengingat peran vital yang dimilikinya terutama bagi pembuat web pemula.

Software XAMPP didirikan oleh suatu perusahaan bernama Apache Friends. Dengan adanya beberapa tools pemrograman seperti MySQL, PHP dan Perl yang dimilikinya tentu mengindikasikan jika anda menekuni salah satu atau semuanya berarti harus memiliki software yang bernama XAMPP ini. Maksud dari Apache yakni selain mengindikasikan nama pengembangnya juga merupakan suatu software yang menghadirkan web server pada komputer anda layaknya web server sesungguhnya.

### **1.2.2 Sejarah XAMPP**

XAMPP merupakan pengembangan dari LAMP (Linux Apache , MySQL, PHP and PERL), XAMPP ini merupakan project non-profit yang dikembangkan oleh Apache Friends yang didirikan Kai ‘Oswald’ Seidler dan Kay Vogelgesang pada

tahun 2002, project mereka ini bertujuan mempromosikan penggunaan Apache web server.

Sejarah mencatat, software XAMPP pertama kali dikembangkan oleh tim proyek bernama Apache Friends dan sampai saat ini sudah masuk dalam rilis versi **7.3.9** yang bisa didapatkan secara gratis dengan label GNU (General Public License).

Jika dijabarkan secara gamblang, masing-masing huruf yang ada di dalam nama XAMPP memiliki arti sebagai berikut ini:

1. **X = Cross Platform**

Merupakan kode penanda untuk software cross platform atau yang bisa berjalan di banyak sistem operasi.

2. **A = Apache**

Apache adalah aplikasi web server yang bersifat gratis dan bisa dikembangkan oleh banyak orang (open source).

3. **M = MySQL / MariaDB**

MySQL atau MariaDB merupakan aplikasi database server yang dikembangkan oleh orang yang sama. MySQL berperan dalam mengolah, mengedit, dan menghapus daftar melalui database.

4. **P = PHP**

Huruf “P” yang pertama dari akronim kata XAMPP adalah inisial untuk menunjukkan eksistensi bahasa pemrograman PHP. Bahasa pemrograman ini biasanya digunakan untuk membuat website dinamis, contohnya dalam website berbasis CMS WordPress.

5. **P = Perl**

Sementara itu, untuk huruf P selanjutnya merupakan singkatan dari bahasa pemrograman Perl yang kerap digunakan untuk memenuhi

berbagai macam kebutuhan. Perl ini bisa berjalan di dalam banyak sistem operasi sehingga sangat fleksibel dan banyak digunakan.

### **1.2.3 Fungsi Xampp**

Jika sebelumnya sudah dijelaskan bahwa XAMPP merupakan gabungan dari beberapa software maka kali ini kami akan menjelaskan satu – persatu fungsi atau kegunaannya, tentunya berhubungan dengan dunia developer web. Sebagai software yang cross platform tentunya dimaksudkan agar semua orang dapat menggunakannya. Apache adalah suatu software yang juga dikembangkan Apache Friends dengan tujuan untuk membuat web server pribadi sehingga anda dapat membuat tampilan web yang dinamis. Istilah ini biasa disebut Localhost. Banyak developer web yang terlebih dahulu mencoba menjalankan webnya di Localhost sebelum akhirnya diposting di web server yang sesungguhnya.

Program aplikasi XAMPP berfungsi sebagai server lokal untuk mengampu berbagai jenis data website yang sedang dalam proses pengembangan. Dalam prakteknya, XAMPP bisa digunakan untuk menguji kinerja fitur ataupun menampilkan konten yang ada didalam website kepada orang lain tanpa harus terkoneksi dengan internet, atau istilahnya website offline. XAMPP bekerja secara offline layaknya web hosting biasa namun tidak bisa diakses oleh banyak orang. Maka dari itu, XAMPP biasanya banyak digunakan oleh para mahasiswa maupun pelajar untuk melihat hasil desain website sebelum akhirnya dibuat online menggunakan web hosting yang biasa dijual dipasaran.

Selain sebagai web server, XAMPP juga menunjang beberapa Bahasa pemrograman khusus dalam website yakni PHP, MySQL dan Perl. PHP merupakan suatu Bahasa yang sering digunakan oleh programmer

khusus Back End karena memang lebih mengutamakan logika dibanding tampilan, beda halnya dengan HTML atau CSS. Oleh karena itu script PHP tidak akan terlihat dalam tampilan website.

MySQL merupakan suatu software yang digunakan untuk mengelola SQL (Structured Query Language). Bahasa ini biasa digunakan untuk keperluan database khusus pada website. Pengelolaan database yang dimaksudkan adalah untuk menambah data, mengubah, menghapus dan lain – lain. Keberadaan MySQL juga biasanya identic dengan Bahasa PHP.

Selanjutnya adalah tools bernama Perl. Bahasa pemrograman yang satu ini tidak hanya digunakan untuk pengelolaan website saja namun juga dalam berbagai hal, juga merupakan salah satu Bahasa pemrograman versi jadul namun tetap bisa eksis sampai sekarang. Perl pertama kali dikenalkan pada tahun 1987 dimana saat itu masih menggunakan Unix.

Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah untuk digunakan yang dapat menampilkan halaman web yang dinamis. Untuk mendapatkannya XAMPP anda dapat mendownload langsung dari web resminya. Dan berikut beberapa definisi program lainnya yang terdapat dalam XAMPP.

#### **1.2.4 Bagian – Bagian Penting Pada Xampp**

Sebagai suatu software yang bertindak sebagai web server layaknya hosting sesungguhnya tentu saja melibatkan banyak bagian – bagian penting yang ada pada XAMPP. Bagi mereka yang terbiasa menggunakan software ini pasti tidak akan asing dengan istilah Htdocs, phpmyadmin dan Control Panel. Baik pada web server yang asli maupun pada software XAMPP juga menggunakannya. Tiga hal tersebut juga menjadi bagian terpenting dalam XAMPP dan akan kami jelaskan satu – persatu disini.

## 1. Htdocs

Htdocs merupakan sebuah folder penyimpanan web server untuk halaman – halaman web yang sudah dibuat dan nantinya akan ditampilkan. Baik pada web server yang asli maupun XAMPP bentuk Htdocs-nya sama namun yang berbeda adalah di kapasitasnya. Karena XAMPP menggunakan penyimpanan internal komputer maka kapasitasnya menyesuaikan komputer anda. Sedangkan pada hosting berbayar kapasitas yang disediakan mengikuti ketentuan yang dibuat.

## 2. PhpMyAdmin

phpMyAdmin merupakan suatu software khusus untuk mengelola administrasi MySQL. Jika pada Htdocs menyimpan file – file tampilan web anda maka di phpMyAdmin ini terdapat semua database yang anda gunakan untuk keperluan website.

## 3. Control Panel

Sesuai dengan namanya, di Control Panel ini anda dapat mengontrol atau mengendalikan XAMPP dengan lebih efektif, mulai dari mengatur setting website, database, dan masih banyak lagi. Dalam dunia hosting lebih dikenal istilah CPanel.

### **1.2.5 Komponen Xampp**

Berikut ini terdapat beberapa komponen xampp, terdiri atas:

#### A. XAMPP 1.8.3 untuk Windows, Termasuk :

- Apache 2.4.4
- MySQL 6.5.11
- PHP 5.5.0
- phpMyAdmin 4.0.4

- FileZilla FTP Server 0.9.41
- Tomcat 7.0.41 (with mod\_proxy\_ajp as connector)
- Strawberry Perl 5.16.3.1 Portabel
- XAMPP Control Panel 3.2.1 (dari hackattack142)

B. XAMPP 1.8.3 untuk Linux, Termasuk :

- Apache 2.4.4
- MySQL 6.5.11
- PHP 5.5.0
- phpMyAdmin 4.0.4
- OpenSSL 1.0.1e

### **1.2.6 Kelebihan Dan Kekurangan Xampp**

Berikut ini terdapat beberapa kelebihan dan kekurangan XAMPP, terdiri atas:

- Kelebihan :
  1. Database Storage Engine ini banyak digunakan oleh programmer apalagi oleh web developer karena sifatnya yang free. Untuk yang expert sudah ada yang bayar.
  2. Kemampuannya sudah bisa diandalkan, mempunyai kapasitas yang cukup mumpuni sekitar 60.000 tabel dengan jumlah record mencapai 5.000.000.000 bahkan untuk yang terbaru sudah lebih.
  3. Keamanan datanya cukup aman walaupun tidak sehebat Postgre apalagi Oracle.

4. Engine ini multiplatform sehingga mampu diaplikasikan di berbagai sistem operasi. My Sql cocok diaplikasikan diaplikasi kelas kecil dan menengah.
  5. Kelebihan paling utama engine ini adalah kecepatannya.
- Kekurangan :
    1. Tidak cocok untuk menangani data dengan jumlah yang besar, baik untuk menyimpan data maupun untuk memproses data.
    2. Memiliki keterbatasan kemampuan kinerja pada server ketika data yang disimpan telah melebihi batas maksimal kemampuan daya tampung server karena tidak menerapkan konsep Technology Cluste.

### 1.3 Pengantar MySql

*MySQL* adalah sebuah program basis data *server* yang mampu menerima dan mengirim datanya dengan sangat cepat, multi user serta menggunakan perintah standar *SQL (structured query language)*. *MySQL* dapat juga berperan sebagai client sehingga sering disebut *database client/server*, yang *open source* dengan kemampuan dapat berjalan baik di OS (*operating system*) manapun.

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis. Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (Structured Query Language). SQL adalah sebuah konsep

pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Jadi MySQL adalah database server yang gratis dengan lisensi GNU General Public License (GPL) sehingga dapat Anda pakai untuk keperluan pribadi atau komersil tanpa harus membayar lisensi yang ada.

Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun demikian pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (wordpress), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

SQL dapat digunakan secara berdiri sendiri maupun dilekatkan pada bahasa pemograman seperti C, dan Delphi.

## 1. Elemen Sql

Elemen dari SQL yang paling dasar antara lain pernyataan, nama, tipe data, ekspresi, konstanta dan fungsi bawaan.

- Pernyataan

Perintah dari SQL yang digunakan untuk meminta sebuah tindakan kepada DBMS.

Pernyataan dasar SQL antara lain :

1. ALTER : Merubah struktur tabel
2. COMMIT : Mengakhiri eksekusi transaksi
3. CREATE : Membuat tabel, indeks
4. DELETE : Menghapus baris pada sebuah tabel
5. DROP : Menghapus tabel, indeks
6. GRANT : Menugaskan hak terhadap basis data kepada user
7. INSERT : Menambah baris pada tabel
8. REVOKE : Membatalkan hak kepada basis data
9. ROLLBACK : Mengembalikan pada keadaan semula apabila transaksi gagal dilaksanakan
10. SELECT : Memilih baris dan kolom pada sebuah tabel
11. UPDATE : Mengubah value pada baris sebuah table

- Nama

Nama digunakan sebagai identitas, yaitu identitas bagi objek pada DBMS. Misal : tabel, kolom dan pengguna.

- Tipe Data

Tipe data yang ada dalam MYSQL :

- a. Tipe data numerik antara lain :

1. TINYINT : Nilai integer yang sangat kecil

2. SMALLINT : Nilai integer yang kecil
3. MEDIUMINT : Nilai integer yang sedang
4. INT : Nilai integer dengan nilai standar
5. BEGINT : Nilai integer dengan nilai besar
6. FLOAT : Bilangan decimal dengan single-precision
7. DOUBLE : Bilangan decimal dengan double-precision
8. DECIMAL(M,D) : Bilangan float yang dinyatakan sebagai string.

M: jumlah digit yang disimpan,

D : jumlah angka dibelakang koma

b. Tipe data String antara lain :

1. CHAR : Karakter yang memiliki panjang tetap yaitu sebanyak n
2. VARCHAR : Karakter yang memiliki panjang tidak tetap yaitu maksimum n
3. TINYBLOB : BLOB dengan ukuran sangat kecil
4. BLOB : BLOB yang memiliki ukuran kecil
5. MEDIUMBLOB : BLOB yang memiliki ukuran sedang
6. LONGBLOB : BLOB yang memiliki ukuran besar
7. TINYTEX : teks dengan ukuran sangat kecil
8. TEXT : teks yang memiliki ukuran kecil
9. MEDIUMTEXT : teks yang memiliki ukuran sedang
10. LONGTEXT : teks yang memiliki ukuran besar

- 11. ENUM : kolom diisi dengan satu member enumerasi
  - 12. SET : Kolom dapat diisi dengan beberapa nilai anggota himpunan
- c. Tipe data tunggal dan jam :
1. DATE : date memiliki format tahun-bulan-tanggal
  2. TIME : time memiliki format jam-menit-detik
  3. DATETIME : gabungan dari format date dan time
- Ekspresi
- Ekspresi digunakan untuk menghasilkan/menghitung nilai.
- Misalnya : jumlah=harga-diskon
- Ekspresi aritmatika antara lain :
1. + : tambah
  2. - : kurang
  3. / : bagi
  4. \* : kali
- Konstanta
  - Nilai yang tetap
  - Fungsi Bawaan
- Fungsi adalah subprogram yang dapat menghasilkan suatu nilai apabila fungsi tersebut dipanggil. Fungsi Agregat adalah fungsi yang digunakan untuk melakukan summary, statistik yang dilakukan pada suatu tabel/query.
1. AVG(ekspresi) : digunakan untuk mencari nilai rata-rata dalam kolom dari tabel.
  2. COUNT(x) : digunakan untuk menghitung jumlah baris dari sebuah kolom dari tabel

3. MAX(ekspresi) : digunakan untuk mencari nilai yang paling besar dari suatu kolom dari tabel
  4. MIN(ekspresi) : digunakan untuk mencari nilai yang paling kecil dari suatu kolom dari tabel
  5. SUM(ekspresi) : digunakan untuk mengitung jumlah keseluruhan dari suatu kolom dari table
2. Kelompok Pernyataan Sql

Pernyataan SQL dapat dikelompokan menjadi 3 kelompok, yaitu : DDL, DML dan DCL.

a. Pernyataan SQL kelompok DDL (Data Definition Language)

DDL berfungsi untuk mendefinisikan atribut basis data, table, atribut(kolom), batasan-batasan terhadap suatu atribut, serta hubungan antar tabel. Perintah yang digunakan biasanya : CREATE, ALTER, dan DROP

1. Membuat Database

Syntax yang digunakan : CREATE DATABASE  
namadatabase;

2. Menghapus Database

Syntax yang digunakan : DROP DATABASE namadatabase;

3. Membuat Tabel

Syntax yang digunakan : CREATE TABLE namatabel;

4. Menghapus Database

Syntax yang digunakan : DROP TABLE namatabel;

5. Menghapus Tabel

Syntax yang digunakan : DROP TABLE namatabel;

6. Mendefinisikan null/not null

Syntax yang digunakan : CREATE TABLE namatabel

( Field1 TipeData1 NOT NULL,Field2 TipeData2);

7. Mendefinisikan Nilai Default

Syntax yang digunakan : CREATE TABLE namatabel

( Field1 TipeData1,Field2 TipeData2 DEFAULT nilai);

8. Mendefinisikan Primary Key pada Tabel

Dapat dilakukan dengan 3 Syntax :

- CREATE TABLE namatabel

( Field1 TipeData1 NOT NULL PRIMARY KEY,Field2  
TipeData2);

- CREATE TABLE namatabel

( Field1 TipeData1,Field2 TipeData2,PRIMARY  
KEY(Field1));

- ALTER TABEL namatabel ADD CONSTRAINT  
namaconstraint PRIMARY KEY (NAMAKOLOM)

9. Menghapus Primary Key pada Tabel

Syntax yang digunakan ada 2 :

- ALTER TABLE namatabel DROP CONSTRAINT  
NAMACONSTRAINT;

- ALTER TABLE NAMATABEL DROP PRIMARY  
KEY;

10. Menambah Kolom Baru pada Tabel

Syntax yang digunakan : ALTER TABEL NAMATABEL  
ADD newfield tipe;

11. Mengubah Tipe Data atau Lebar Kolom pada Tabel

Syntax yang digunakan : ALTER TABEL namatabel  
MODIFY COLUMN field tipe;

12. Mengubah Nama Kolom

Syntax yang digunakan : ALTER TABEL namabel  
CHANGE COLUMN namakolomlama namakolombaru tipe;

13. Menghapus Kolom pada Tabel

Syntax yang digunakan : ALTER TABEL namabel DROP  
COLUMN namakolom;

14. Mendefinisikan Foreign Key pada Tabel

Dapat dilakukan dengan 2 Syntax :

- CREATE TABLE namabel( Field1 TipeData1,Field2 TipeData2,FOREIGN KEY (Field2) REFERENCES namabelinduk (namakolominduk) ON UPDATE CASCADE ON DELETE NO ACTION)
- ALTER TABEL namabel ADD CONSTRAINT namaconstraint FOREIGN KEY (namakolom) REFERENCES namabelinduk (namakolominduk) ON UPDATE CASCADE ON DELETE NO ACTION;

15. Menghapus Foreign Key

Syntak yang digunakan : ALTER TABEL namabel DROP  
FOREIGN KEY namaconstraint;

b. DML (Data Manipulation Language)

DML berfungsi untuk memanipulasi data yang ada di dalam basis data, contohnya untuk pengambilan data, penyisipan data, pengubahan data dan penghapusan data.

Perintah yang digunakan biasanya adalah : INSERT, DELETE, UPDATE, dan SELECT.

1. INSERT

menambah baris pada tabel. Syntax yang paling sering digunakan : INSERT INTO namatabel VALUES (nilai1, nilai2, nilai-n);

## 2. DELETE

Menghapus baris pada tabel. Syntax : DELETE FROM namatabel [where kondisi];

## 3. UPDATE

Mengubah isi beberapa kolom pada tabel. Syntax : UPDATE namatabel SET kolom1=nilai1, kolom2=nilai2 [where kondisi];

## 4. SELECT

Menampilkan isi dari suatu tabel yang bisa dihubungkan dengan tabel yang lainnya;

### 1.3.1 Sejarah Mysql

MySQL pada awalnya diciptakan pada tahun 1979, oleh Michael "Monty" Widenius, seorang programmer komputer asal Swedia. Monty mengembangkan sebuah sistem database sederhana yang dinamakan UNIREG yang menggunakan koneksi low-level ISAM database engine dengan indexing. Pada saat itu Monty bekerja pada perusahaan bernama TcX di Swedia.

TcX pada tahun 1994 mulai mengembangkan aplikasi berbasis web, dan berencana menggunakan UNIREG sebagai sistem database. Namun sayangnya, UNIREG dianggap tidak cocok untuk database yang dinamis seperti web.

TcX kemudian mencoba mencari alternatif sistem database lainnya, salah satunya adalah mSQL (miniSQL). Namun mSQL versi 1 ini

juga memiliki kekurangan, yaitu tidak mendukung indexing, sehingga performanya tidak terlalu bagus.

Dengan tujuan memperbaiki performa mSQL, Monty mencoba menghubungi David Hughes (programmer yang mengembangkan mSQL) untuk menanyakan apakah ia tertarik mengembangkan sebuah konektor di mSQL yang dapat dihubungkan dengan UNIREG ISAM sehingga mendukung indexing. Namun saat itu Hughes menolak, dengan alasan sedang mengembangkan teknologi indexing yang independen untuk mSQL versi 2.

Dikarenakan penolakan tersebut, David Hughes, TcX (dan juga Monty) akhirnya memutuskan untuk merancang dan mengembangkan sendiri konsep sistem database baru. Sistem ini merupakan gabungan dari UNIREG dan mSQL (yang source codenya dapat bebas digunakan). Sehingga pada May 1995, sebuah RDBMS baru, yang dinamakan MySQL dirilis.

David Axmark dari Detron HB, rekanan TcX mengusulkan agar MySQL di ‘jual’ dengan model bisnis baru. Ia mengusulkan agar MySQL dikembangkan dan dirilis dengan gratis. Pendapatan perusahaan selanjutnya di dapat dari menjual jasa “support” untuk perusahaan yang ingin mengimplementasikan MySQL. Konsep bisnis ini sekarang dikenal dengan istilah Open Source.

Pada tahun 1995 itu juga, TcX berubah nama menjadi MySQL AB, dengan Michael Widenius, David Axmark dan Allan Larsson sebagai pendirinya. Titel “AB” di belakang MySQL, adalah singkatan dari “Aktiebolag”, istilah PT (Perseroan Terbatas) bagi perusahaan Swedia

MySQL adalah pengembangan lanjutan dari proyek UNIREG yang dikerjakan oleh Michael Monty Widenius dan TcX (perusahaan perangkat lunak asal Swedia).



Gambar 1. 1 MySql

Akana tetapi, UNIREG belum terlalu kompatibel dengan database dinamis yang dipakai di website. TcX kemudian mencari alternatif lain dan menemukan perangkat lunak yang dikembangkan oleh David Hughes, yaitu miniSQL atau mSQL. Namun, ditemukan masalah lagi karena mSQL tidak mendukung indexing sehingga belum sesuai dengan kebutuhan TcX.

MySQL AB merupakan perusahaan komersial Swedia yang mensponsori dan yang memiliki MySQL. Pendiri MySQL AB adalah dua orang Swedia yang bernama David Axmark, Allan Larsson dan satu orang Finlandia bernama Michael “Monty”. Setiap pengguna MySQL dapat menggunakannya secara bebas yang didistribusikan gratis dibawah lisensi GPL(*General Public License*) namun tidak boleh menjadikan produk turunan yang bersifat komersial.

Pada saat ini MySQL merupakan database server yang sangat terkenal di dunia, semua itu tak lain karena bahasa dasar yang digunakan

untuk mengakses database yaitu SQL. SQL (*Structured Query Language*) pertama kali diterapkan pada sebuah proyek riset pada laboratorium riset San Jose, IBM yang bernama system R. Kemudian SQL juga dikembangkan oleh Oracle, Informix dan Sybase. Dengan menggunakan SQL, proses pengaksesan database lebih user-friendly dibandingkan dengan yang lain, misalnya dBase atau Clipper karena mereka masih menggunakan perintah-perintah pemrograman murni.

Pada akhirnya muncul kerjasama antara pengembang UNIREG (Michael Monty Widenius), mSQL (David Hughes), dan TcX. Kerjasama ini bertujuan untuk mengembangkan sistem database yang baru, dan pada 1995 dirilislah MySQL seperti yang dikenal saat ini. Saat ini pengembangan MySQL berada di bawah Oracle.

### **1.3.2 Kelebihan Dan Kekurangan**

Meskipun menjadi database yang cukup populer, MySQL tentu mempunyai beberapa kelebihan dan kekurangan dibandingkan dengan database server lainnya. Salah satu kekurangan MySQL adalah performanya turun di saat beberapa database manajemen sistem mampu bekerja baik pada pengelolaan database yang besar.

#### **A. Kelebihan**

MySQL mempunyai beberapa kelebihan yang bisa Anda manfaatkan untuk mengembangkan perangkat lunak yang andal seperti:

##### **1. Mendukung Integrasi Dengan Bahasa Pemrograman Lain.**

Website atau perangkat lunak terkadang dikembangkan dengan menggunakan berbagai macam bahasa pemrograman, jadi Anda tidak perlu khawatir jika menggunakan MySQL. Maka dari itu, MySQL bisa membantu Anda untuk mengembangkan perangkat lunak yang lebih

efektif dan tentu saja lebih mudah dengan integrasi antara bahasa pemrograman.

## **2. Tidak Membutuhkan RAM Besar.**

MySQL dapat dipasang pada server dengan spesifikasi kecil. Jadi tidak perlu khawatir jika Anda hanya mempunyai server dengan kapasitas 1 GB karena Anda masih bisa menggunakan MySQL sebagai database Anda.

## **3. Mendukung Multi User.**

MySQL dapat dipakai oleh beberapa user dalam waktu bersamaan tanpa membuatnya crash atau berhenti bekerja. Ini dapat Anda manfaatkan ketika mengerjakan proyek yang sifatnya tim sehingga seluruh tim dapat bekerja dalam waktu bersamaan tanpa harus menunggu user lain selesai.

## **4. Bersifat Open Source**

MySQL adalah sistem manajemen database gratis. Meskipun gratis, bukan berarti database ini mempunyai kinerja buruk. Apalagi lisensi gratis yang dipakai adalah GPL di bawah pengelolaan Oracle sehingga kualitasnya termasuk baik. Selain itu, Anda juga tidak perlu khawatir jika terjadi masalah karena banyak komunitas dan dokumentasi yang membahas soal MySQL.

## **5. Struktur Tabel yang Fleksibel.**

MySQL mempunyai struktur tabel yang mudah dipakai dan fleksibel. Contohnya saat MySQL memproses ALTER TABLE dan lain sebagainya. Jika dibandingkan dengan database lain seperti Oracle dan PostgreSQL, MySQL tergolong lebih mudah.

## **6. Tipe Data yang Bervariasi.**

Kelebihan lain dari MySQL adalah mendukung berbagai macam data yang bisa Anda gunakan di MySQL. Contohnya float, integer, date, char, text, timestamp, double, dan lain sebagainya. Jadi manajemen database sistem ini sangat membantu Anda untuk mengembangkan perangkat lunak yang berguna untuk pengelolaan database di server.

## **7. Keamanan yang Terjamin.**

Open source bukan berarti MySQL menyediakan keamanan yang buruk. Malah sebaliknya, MySQL mempunyai fitur keamanan yang cukup apik. Ada beberapa lapisan keamanan yang diterapkan oleh MySQL, seperti level nama host, dan subnetmask. Selain itu MySQL juga dapat mengatur hak akses user dengan enkripsi password tingkat tinggi.

## **B. Kekurangan**

Sayangnya, meskipun memiliki segudang kelebihan, masih ada beberapa kelemahan yang dimiliki oleh MySQL sehingga perlu mempertimbangkannya juga sebelum memakainya.

### **1. Kurang Cocok untuk Aplikasi Game dan Mobile**

Anda yang ingin mengembangkan aplikasi game atau perangkat mobile ada baiknya jika mempertimbangkan lagi jika ingin menggunakan MySQL. Kebanyakan pengembang game maupun aplikasi mobile tidak menggunakan MySQL karena memang database manajemen sistem ini masih kurang bagus dipakai untuk sistem aplikasi tersebut.

## **2. Sulit Mengelola Database yang Besar**

Jika Anda ingin mengembangkan aplikasi atau sistem di perusahaan dengan database yang cukup besar, ada baiknya jika menggunakan database manajemen sistem selain MySQL. MySQL dikembangkan supaya ramah dengan perangkat yang mempunyai spesifikasi rendah, itulah mengapa MySQL tidak memiliki fitur yang lengkap seperti aplikasi lainnya

## **3. Technical Support yang Kurang Bagus**

Sifatnya yang open source terkadang membuat aplikasi tidak menyediakan technical support yang memadai. Technical support MySQL diklaim kurang bagus. Hal ini membuat pengguna kesulitan. Apalagi jika pengguna mengalami masalah yang berhubungan dengan pengoperasian perangkat lunak tersebut dan membutuhkan bantuan technical support.

## **1.4 Pengantar PHP**

### **1.4.1 Penjelasan PHP**

PHP adalah bahasa pemrograman yang sering disisipkan ke dalam HTML. PHP sendiri berasal dari kata *Hypertext Preprocessor*. Sejarah PHP pada awalnya merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama Form Interpreted (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web.

Bahasa pemrograman ini menggunakan sistem server-side. *Server-side programming* adalah jenis bahasa pemrograman yang nantinya script/program tersebut akan dijalankan/diproses oleh server.

Kelebihannya adalah mudah digunakan, sederhana, dan mudah untuk dimengerti dan dipelajari.

Semenjak PHP menjadi bahasa pemrograman yang open source, pengembang tidak perlu menunggu sampai dengan update terbaru rilis.

Pengguna PHP akan lebih baik jika menggunakan versi terbaru. Sehingga jika ada rilis terbaru Anda harus menyesuaikan sistem Anda dengan versi PHP yang paling baru. Meskipun harus menggunakan versi terbaru, biaya untuk maintenance dan web development sangat terjangkau.

Bahasa pemrograman PHP membantu Anda untuk mengembangkan aplikasi berbasis web yang cukup kompleks, handal, dan cepat. Tergantung dari spesifikasi bisnis, penggunaan hosting, tingkat pengalaman, kebutuhan aplikasi, dan pengembangan timeframe. Selain itu ada banyak PHP frameworks yang dapat Anda pilih.

PHP juga ialah *bahasa pemrograman script server-side yang didesain untuk pengembangan web*. Selain itu, PHP juga bisa digunakan sebagai *bahasa pemrograman umum*. PHP di kembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group. Situs resmi PHP beralamat di PHP disebut bahasa pemrograman *server side* karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti *JavaScript* yang diproses pada web *browser (client)*.

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

PHP dapat digunakan dengan gratis (*free*) dan bersifat *Open Source*. PHP dirilis dalam lisensi *PHP License*, sedikit berbeda dengan

lisensi *GNU General Public License (GPL)* yang biasa digunakan untuk proyek *Open Source*.

#### 1.4.2 Sejarah PHP

Pada awalnya PHP merupakan singkatan dari *Personal Home Page*. PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama Form Interpreted (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisan kode sumber ini menjadi sumber terbuka, maka banyak pemrograman yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang *PHP: Hypertext Preprocessing*.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Server web bawaan ditambahkan pada versi 5.4 untuk mempermudah pengembang menjalankan kode PHP tanpa menginstall software server.

Sesuai dengan namanya, PHP digunakan untuk membuat *website* pribadi. Dalam beberapa tahun perkembangannya, PHP menjelma menjadi bahasa pemrograman web yang *powerful* dan tidak hanya digunakan untuk membuat halaman web sederhana, tetapi juga website populer yang digunakan oleh jutaan orang seperti wikipedia, wordpress, joomla, dan lain lain. Saat ini PHP adalah singkatan dari PHP: Hypertext Preprocessor, sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri: **PHP: Hypertext Preprocessor**.

### 1.4.3 Fungsi PHP

Untuk membuat halaman web sebenarnya PHP bukanlah bahasa pemograman yang wajib digunakan. Kita bisa saja membuat website dengan menggunakan HTML saja. Web yang dihasilkan dengan HTML(dan CSS) ini dikenal dengan website statis dimana kontenn dan halaman web bersifat tetap.

Sebagai perbandingan website dinamis yang bisa dibuat menggunakan PHP adalah situs web yang bisa menyesuaikan tampilan konten tergantung situasi. Untuk pembuatan web kode PHP biasanya disispkan dalam dokument HTML. Karena PHP disebut juga sebagai Scripting Language atau bahasa pemograman script.

Untuk pembuatan web, kode PHP biasanya di sisipkan ke dalam dokumen HTML. Karena fitur inilah PHP disebut juga sebagai Scripting Language atau bahasa pemrograman script.

Sebagai contoh penggunaan PHP, misalkan kita ingin membuat list dari nomor 1 sampai nomor 10. Dengan menggunakan HTML murni, kita bisa membuatnya secara manual seperti kode berikut ini:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Contoh list dengan HTML</title>
    </head>
    <body>
        <h2>Daftar Absensi Mahasiswa</h2>
        <ol>
            <li>Nama Mahasiswa ke-1</li>
            <li>Nama Mahasiswa ke-2</li>
            <li>Nama Mahasiswa ke-3</li>
            <li>Nama Mahasiswa ke-4</li>
            <li>Nama Mahasiswa ke-5</li>
            <li>Nama Mahasiswa ke-6</li>
            <li>Nama Mahasiswa ke-7</li>
            <li>Nama Mahasiswa ke-8</li>
            <li>Nama Mahasiswa ke-9</li>
            <li>Nama Mahasiswa ke-10</li>
        </ol>
    </body>
</html>
```

Gambar 1. 2 Contoh Penggunaan HTML

Halaman HTML tersebut dapat digunakan dengan mudah dengan mengcopy tag <li> sebanyak 7 kali dan mengubah sedikit angka dan

nomernya namun jika yang diinginkan adalah menambah list tersebut menjadi 100 atau 1000 list cara tersebut menjadi tidak efektif.

Jika menggunakan PHP, Kita tinggal membuat perulangan for sebanyak 100 kali dengan perintah singkat seperti berikut:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Contoh list dengan PHP</title>
    </head>
<body>
<h2>Daftar Absensi Mahasiswa</h2>
<ol>
    <?php
        for ($i= 1; $i <= 1000; $i++)
        {
            echo "<li>Nama Mahasiswa ke-$i</li>";
        }
    ?>
</ol>
</body>
</html>
```

Gambar 1. 3 Contoh Penggunaan PHP

Dengan menggunakan kode baris yang bahkan lebih sedikit, kita dapat membuat list tersebut menjadi 1000 kali, bahkan 100.000 kali dengan hanya mengubah sebuah variabel **\$i**.

PHP tidak hanya dapat melakukan pengulangan tersebut masih banyak hal lain yang bisa dilakukan dengan PHP seperti menginputkan data ke database menghasilkan gambar mengkonversi halaman txt menjadi PDF dan hal lainnya.

## **1.5 Database**

### **1.5.1 Pengertian Database**

Database (Basis data) adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur data dan juga batasan-batasan pada data yang kemudian disimpan.

Basis data merupakan aspek yang sangat penting dalam sistem informasi karena berfungsi sebagai gudang penyimpanan data yang akan diolah lebih lanjut. Basis data menjadi penting karena dapat mengorganisasi data, menghindari duplikasi data, menghindari hubungan antar data yang tidak jelas dan juga update yang rumit.

Proses untuk memasukkan dan juga mengambil data ke dan dari media penyimpanan data memerlukan perangkat lunak (software) yang disebut dengan sistem manajemen basis data (database management system atau DBMS). DBMS merupakan sistem perangkat lunak yang memungkinkan pengguna basis data (database user) untuk memelihara, mengontrol dan mengakses data secara praktis dan efisien.

Dengan kata lain, semua akses ke basis data akan ditangani oleh DBMS. DBMS ini menjadi lapisan yang menghubungkan basis data dengan program aplikasi untuk memastikan bahwa basis data tetap terorganisasi secara konsisten dan dapat diakses dengan mudah.

Ada beberapa fungsi yang harus ditangani DBMS seperti pendefinisian data, menangani permintaan pengguna untuk mengakses data, memeriksa sekuriti dan integriti data yang didefinisikan oleh DBA (Database Administrator), menangani kegagalan dalam pengaksesan data

yang disebabkan oleh kerusakan sistem maupun media penyimpanan (disk) dan juga menangani unjuk kerja semua fungsi secara efisien.

Tujuan utama DBMS adalah untuk memberikan tinjauan abstrak data kepada pengguna. Jadi sistem menyembunyikan informasi tentang bagaimana data disimpan, dipelihara dan juga bisa diakses secara efisien. Pertimbangan efisien di sini adalah rancangan struktur data yang kompleks tetapi masih bisa digunakan oleh pengguna awam tanpa mengetahui kompleksitas strukturnya.

Secara sederhana *Database* ( basis data ) dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Dalam hal ini, pengertian akses dapat mencakup pemerolehan data maupun pemanipulasi data, seperti menambah dan menghapus data. Manajemen modern mengikutsertakan informasi sebagai sumber daya penting yang setara dengan sumber daya manusia, uang, mesin, dan material. Informasi adalah suatu bentuk penyajian data yang melalui mekanisme pemrosesan, yang berguna bagi pihak tertentu, misalnya manajer. Bagi pihak manajemen, informasi merupakan bahan untuk pengambilan keputusan.

### **1.5.2 Fungsi Database**

Mungkin database hanya diartikan sebagai kumpulan data dan informasi yang sepele. Padahal lebih daripada itu, database memiliki beberapa fungsi lain yang mampu mempermudah kinerja seseorang di era penyebarluasan informasi yang serba cepat ini. Beberapa fungsi lainnya diantaranya adalah sebagai berikut.

1. Sesuai dengan pengertian database yang mana memungkinkan kita untuk mengelompokkan data dengan tujuan untuk mempermudah identifikasi data. Pengelompokan bisa dilakukan melalui berbagai cara

seperti misalnya membuat beberapa tabel atau *field* yang berbeda. Sebagai contohnya, hal ini sangat berperan bagi pihak perusahaan bank untuk mengumpulkan dan mengakses kembali data-data dari para nasabahnya.

2. Database juga mampu menghindari data yang tersimpan lebih dari sekali. Pada dasarnya semua kata perintah telah terekam di komputer sehingga bisa ketahuan apabila ada nama file ada data yang ganda.
3. Database juga mempermudah penggunaan di segala sisi, seperti saat memasukkan data baru ataupun mengupdate data dan menghapusnya. Tentunya database juga memungkinkan kita untuk melakukan riset dari jarak jauh karena sudah memiliki akses untuk menelpon dan berkomunikasi melalui nomor yang tertera.
4. Menggunakan teknologi ini juga memungkinkan kita untuk menghemat daya penyimpanan pada produk tertentu. Seperti misalnya kita bisa menyimpan data secara langsung melalui server dan jaringan internet sehingga tidak perlu lagi memberdayakan sumber daya manusia untuk menghimpun data yang jumlahnya sangat banyak itu.
5. Mampu Menghemat Kertas

Penggunaan database juga memungkinkan kita untuk menyimpan file secara digital sehingga tidak perlu lagi menggunakan kertas dalam jumlah banyak. Hal ini pastinya sangat praktis karena kita bisa langsung mengcopy setiap data yang diperlukan tanpa harus menulisnya kembali di kertas.

### **1.5.3 Manfaat Database**

Pada perkembangan teknologi yang bergulir sangat cepat ini siapa yang tidak kenal dengan istilah basis data (database). Database sudah tidak asing lagi di telinga orang-orang generasi Y dan Z apalagi dikalangan

orang IT (Information Technology). Database diciptakan untuk menutupi kelemahan pada sistem berbasis file yang masih banyak terdapat redundansi data dan tingkat inkonsistensi data yang masih sangat tinggi.

Berkat perkembangan teknologi juga lah yang membuat software pengolah database bermunculan atau biasa di panggil DBMS (database management system). Seperti MySQL, Ms. Access, SQL Server, PostgreSQL, Oracle, SQLite, MongoDB, MariaDB, dan lainnya.

Dengan bermuculannya berbagai macam DBMS dengan keunggulan masing-masing, itu membuktikan bahwa kebutuhan akan database meningkat pesat dan bermanfaat untuk pengelolaan data yang banyak dan kompleks. berikut ini beberapa manfaat yang bisa didapatkan jika bekerja dengan sistem database:

#### 1.Tidak Terjadi Redudansi Basis Data

Seperti yang sudah disinggung pada pengertian database sebelumnya, database bisa membantu meminimalkan redudansi data. Redudansi sendiri merupakan terjadinya data-data ganda dalam berkas-berkas yang berbeda.

#### 2.Integritas Data Terjaga

Database memastikan integritas data yang tinggi dimana database akan memastikan keakuratan, aksesibilitas, konsistensi dan juga kualitas tinggi pada suatu data.

#### 3.Independensi Data Terjaga

Database menjaga independensi data dimana orang lain tidak dapat merubah data meskipun data bisa diakses.

#### 4.Kemudahan Berbagi Data

Menggunakan perangkat lunak database bisa digunakan untuk berbagi data atau informasi dengan sesama pengguna lainnya.

## **5.Menjaga Keamanan Data**

Database menjamin keamanan suatu informasi dan data, dimana Anda bisa menyisipkan kode akses untuk data-data tertentu yang tidak bisa diakses bersama.

## **6.Kemudahan Akses Data**

Dengan database bisa memudahkan untuk mengakses dan mendapatkan data karena semua data terorganisir dengan baik.

### **A. Model Database**

Database mempunyai dua varian model, yaitu model Post-relational database dan model Object database.

#### **1. Post-relational database models**

Sebuah produk yang menawarkan model data yang lebih umum dari model relasional dan dikenal sebagai post-relational. Model data dalam produk tersebut mencakup hubungan namun tidak dibatasi oleh Prinsip Informasi yang mana mewakili semua informasi dengan nilai-nilai data dalam kaitannya dengan hal itu.

#### **2. Object database models**

Dalam beberapa tahun terakhir,[update], paradigma yang berorientasi pada obyek telah diterapkan dalam bidang-bidang seperti teknik dan spasial database, telekomunikasi dan ilmu pilmiah lainnya. Para konglomerasi pemrograman berorientasi objek dan teknologi database mengarah pada model pemrograman baru yang dikenal sebagai Object database.

### **B. Tipe Database**

Terdapat 12 tipe database, antara lain:

1. Operational database: Database ini menyimpan data rinci yang diperlukan untuk mendukung operasi dari seluruh organisasi. Mereka juga disebut *subject-area databases* (SADB), transaksi database, dan produksi database. Contoh: database pelanggan, database pribadi, database inventaris, akuntansi database.
2. Analytical database: Database ini menyimpan data dan informasi yang diambil dari operasional yang dipilih dan eksternal database. Mereka terdiri dari data dan informasi yang dirangkum paling dibutuhkan oleh sebuah organisasi manajemen dan End-user lainnya. Beberapa orang menyebut analitis multidimensi database sebagai database, manajemen database, atau informasi database.
3. Data warehouse: Sebuah data warehouse menyimpan data dari saat ini dan tahun-tahun sebelumnya – data yang diambil dari berbagai database operasional dari sebuah organisasi.
4. Distributed database: Ini adalah database-kelompok kerja lokal dan departemen di kantor regional, kantor cabang, pabrik-pabrik dan lokasi kerja lainnya. Database ini dapat mencakup kedua segmen yaitu operasional dan user database, serta data yang dihasilkan dan digunakan hanya pada pengguna situs sendiri.
5. End-user database: Database ini terdiri dari berbagai file data yang dikembangkan oleh end-user di workstation mereka. Contoh dari ini adalah koleksi dokumen dalam spreadsheet, word processing dan bahkan download file.
6. External database: Database ini menyediakan akses ke eksternal, data milik pribadi online – tersedia untuk biaya kepada pengguna akhir dan organisasi dari layanan komersial. Akses ke kekayaan informasi dari database eksternal yang tersedia untuk biaya dari

layanan online komersial dan dengan atau tanpa biaya dari banyak sumber di Internet.

7. Hypermedia databases on the web: Ini adalah kumpulan dari halaman-halaman multimedia yang saling berhubungan di sebuah situs web. Mereka terdiri dari home page dan halaman hyperlink lain dari multimedia atau campuran media seperti teks, grafik, gambar foto, klip video, audio dll.
8. Navigational database: Dalam navigasi database, queries menemukan benda terutama dengan mengikuti referensi dari objek lain.
9. In-memory databases: Database di memori terutama bergantung pada memori utama untuk penyimpanan data komputer. Ini berbeda dengan sistem manajemen database yang menggunakan disk berbasis mekanisme penyimpanan. Database memori utama lebih cepat daripada dioptimalkan disk database sejak Optimasi algoritma internal menjadi lebih sederhana dan lebih sedikit CPU mengeksekusi instruksi.
10. Document-oriented databases: Merupakan program komputer yang dirancang untuk aplikasi berorientasi dokumen. Sistem ini bisa diimplementasikan sebagai lapisan di atas sebuah database relasional atau objek database. Sebagai lawan dari database relasional, dokumen berbasis database tidak menyimpan data dalam tabel dengan ukuran seragam kolom untuk setiap record. Sebaliknya, mereka menyimpan setiap catatan sebagai dokumen yang memiliki karakteristik tertentu. Sejumlah bidang panjang apapun dapat ditambahkan ke dokumen. Bidang yang dapat juga berisi beberapa bagian data.

11. Real-time databases Real-time: Database adalah sistem pengolahan dirancang untuk menangani beban kerja negara yang dapat berubah terus-menerus. Ini berbeda dari database tradisional yang mengandung data yang terus- menerus, sebagian besar tidak terpengaruh oleh waktu. Sebagai contoh, pasar saham berubah dengan cepat dan dinamis. Real-time processing berarti bahwa transaksi diproses cukup cepat bagi hasil untuk kembali dan bertindak segera. Real-time database yang berguna untuk akuntansi, perbankan, hukum, catatan medis, multi-media, kontrol proses, sistem reservasi, dan analisis data ilmiah.
12. Relational Database: Database yang paling umum digunakan saat ini. Menggunakan meja untuk informasi struktur sehingga mudah untuk mencari.

### C. Jenis – Jenis Software Database

berikut ini jenis-jenis software database terbaik yang bisa Anda gunakan:

#### 1. Microsoft Access

Salah satu software database ini adalah yang paling sering digunakan. Microsoft access sangat cocok digunakan untuk sebagian besar komputer yang relasional. Selain itu, jika Anda berbisnis dalam skala rumahan, bisa memilih sistem database ini karena sangat ringan digunakan dan format datanya sangat umum sehingga memudahkan pembacaan.

#### 2. Oracle

Salah satu software database ini sangat mampu untuk menyimpan data dengan ukuran yang maksimum hingga tera byte. Oracle

paling banyak digunakan pada perusahaan-perusahaan terutama yang sedang berkembang karena memang untuk mengaksesnya tersedia secara gratis.

### 3. Ms SQL Server

Software database ini merupakan manajemen basis data yang umum digunakan pada Microsoft dengan bahasa pemrograman yang digunakan adalah Transact-SQL. Tipe data yang digunakan cukup banyak sehingga sangat efektif untuk mendukung kinerja Anda.

### 4. MySQL

Salah satu software database yang open access untuk umum dan kompatibel pada sistem operasi Windows maupun Linux. Keunggulan yang bisa Anda gunakan dengan menggunakan program MySQL adalah bisa digunakan untuk multi user. Kelebihan lainnya dari MySQL yaitu tersedia gratis, query data yang cepat dan berlisensi resmi.

### 5. Firebird

Bisa dibilang software database ini memiliki fitur sistem yang standar dan ringan yaitu fitur ANSI SQL-99 dan SQL – 2003. Kompatibel untuk digunakan pada sistem operasi Windows, Linux maupun Unix.

### 6. PostgreSQL

Menawarkan sistem database opensource dengan lisensi GPL/ General Public License. Software ini menggunakan bahasa pemrograman C++, C, SQL, PHP dan lainnya. Jika digunakan untuk pekerjaan pribadi, maka software ini sangat recommended digunakan.

## **BAB II**

---

### **PENGENALAN STRING MATCHING, ALGORITMA KNUTH MORRIS PRATT, WATERFALL, DAN CODEIGNITER**

#### **2.1 Pengantar String Matching**

Pengolahan *String matching* atau sering disebut dengan pencocokan *String* adalah proses pencarian semua kemunculan *query* yang selanjutnya disebut *Pattern* ke dalam *String* yang lebih panjang (teks). *Pattern* yang dilambangkan dengan  $x=x[0..m-1]$  dan panjangnya adalah  $m$ . Teks dilambangkan dengan  $y=y[0..n-1]$  dan panjangnya adalah  $n$ . Kedua *String* terdiri dari sekumpulan karakter yang disebut alphabet yang di lambangkan dengan  $\Sigma$  dan mempunyai ukuran  $\sigma$ . *String matching* dibagi menjadi dua, yakni exact *matching* dan heuristic.

1. Exact string matching, merupakan pencocokan *String* secara tepat dengan susunan karakter dalam *string* yang dicocokkan memiliki jumlah maupun urutan karakter dalam *string* yang sama.
2. Inexact string matching atau Fuzzy string matching, merupakan pencocokan *String* secara samar, maksudnya pencocokan *string* dimana *string* yang dicocokkan memiliki kemiripan dimana keduanya memiliki susunan karakter yang berbeda (mungkin jumlah atau urutannya) tetapi *string*-*String* tersebut memiliki kemiripan baik kemiripan textual/penulisan (approximate string matching) atau kemiripan ucapan(phnetic string matching).

Algoritma string matching dapat diklasifikasikan menjadi 3 bagian menurut arah pencariannya, yaitu :

1. From left to right Dari arah yang paling alami, dari kiri ke kanan, yang merupakan arah untuk membaca. Algoritma yang termasuk kategori ini adalah algoritma brute force, algoritma knuth morris Pratt.
2. From right to left Dari arah kanan ke kiri, arah yang biasanya menghasilkan hasil terbaik secara partikular. Algoritma yang termasuk kategori ini adalah algoritma boyer-moore.
3. In a specific order Dari arah yang ditentukan secara spesifik oleh algoritma tersebut, arah ini menghasilkan hasil terbaik secara teoritis. Algoritma yang termasuk kategori ini adalah algoritma colossi dan algoritma crochemore-perrin.

## 2.2 Pengantar Algoritma Knuth Morris Pratt(KMP)

Algoritma Knuth-Morris-Pratt adalah salah satu algoritma pencarian string, dikembangkan secara terpisah oleh Donald E. Knuth pada tahun 1967 dan James H. Morris bersama Vaughan R. Pratt pada tahun 1966, namun keduanya mempublikasikannya secara bersamaan pada tahun 1977.

*Algoritma Knuth-Morris-Pratt* merupakan pengembangan dari algoritma pencarian *string* sebelumnya, yaitu algoritma *Brute Force*. Algoritma Brute-Force merupakan algoritma dasar yang paling sederhana dalam menyelesaikan persoalan pencocokan *string* yang melakukan pencarian pada setiap posisi di dalam teks antara 0 dan n-m, dimana n adalah panjang teks/banyaknya nama file yang tersimpan di komputer dan m adalah panjang karakter dari suatu *pattern* (kata yang akan dicari)

Dengan menggunakan algoritma *Knuth-Morris-Pratt*, secara rata - rata proses pencarian akan menjadi lebih cepat jika dibandingkan dengan algoritma lainnya dan masalah yang dicoba diselesaikan oleh algoritma KMP adalah melakukan pencocokan string.

Kompleksitas algoritma pencocokan string dihitung dari jumlah operasi perbandingan yang dilakukan. Kompleksitas waktu terbaik dari algoritma brute force adalah  $O(n)$ . Kasus terbaik terjadi jika pada operasi perbandingan, setiap huruf pattern yang dicocokkan dengan awal dari teks adalah sama. Kompleksitas waktu terburuk dari *brute force* adalah  $O(mn)$ . Jika dibandingkan dengan algoritma brute force, maka algoritma KMP mempunyai kompleksitas algoritma  $O(m+n)$ .

Algoritma KMP melakukan proses awal terhadap pattern dengan menghitung fungsi pinggiran Dengan adanya fungsi pinggiran ini, maka dapat dicegah pergeseran yang tidak berguna, seperti yang terjadi pada algoritma brute force. Fungsi pinggiran hanya bergantung pada karakter yang ada di dalam pattern, tidak bergantung kepada karakter di dalam teks. Berikut adalah algoritma untuk mencari fungsi pinggiran di dalam pattern.

### 2.2.1 Penggunaan Algoritma Knuth Morris Pratt

Perhitungan penggeseran pada algoritma ini adalah sebagai berikut, bila terjadi ketidak cocokan pada saat pattern sejajar dengan teks[i..i+n-1], kita bisa menganggap ketidak cocokan pertama terjadi di antara teks[i+j] dan pattern[j], dengan  $0 < j < n$ . Berarti, teks[i..i+j-1] = pattern [0..j-1] dan a=teks[i+j] tidak sama dengan b = pattern[j]. Ketika kita menggeser, sangat beralasan bila ada sebuah awalan u dari pattern akan sama dengan sebagian akhiran u dari sebagian teks. Sehingga kita bisa menggeser pattern agar awalan u tersebut sejajar dengan akhiran dari u.

Dengan kata lain, pencocokkan string akan berjalan secara efisien bila kita mempunyai tabel yang menentukan berapa panjang kita seharusnya menggeser seandainya terdeteksi ketidakcocokan di karakter ke- $j$  dari pattern. Tabel itu harus memuat  $\text{next}[j]$  yang merupakan posisi karakter  $\text{pattern}[j]$  setelah digeser, sehingga kita bisa menggeser pattern sebesar  $j-\text{next}[j]$  relatif terhadap teks.

Secara sistematis, langkah-langkah yang dilakukan algoritma *Knuth Morris Pratt* pada saat mencocokkan *string* yaitu :

1. Masukkan Query kata yang akan dicari. Dengan permisalan  $P = \text{Pattern}$  atau pola susunan kata yang dijadikan sebagai contoh atau pola teks yang akan dicari  $T = \text{Teks}$  atau judul dokumen.
2. Algoritma *Knuth Morris Pratt* mulai mencocokkan *pattern* atau pola susunan kata yang dijadikan sebagai contoh pada awal teks.
3. Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter *pattern* atau pola susunan kata yang dijadikan sebagai contoh dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi yaitu :
  - a. Karakter di *pattern* atau pola susunan kata yang dijadikan sebagai contoh dan di teks yang dibandingkan tidak cocok (mismatch).
  - b. Semua karakter di *pattern* atau pola susunan kata yang dijadikan sebagai contoh cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
4. Algoritma kemudian menggeser *pattern* atau pola susunan kata yang dijadikan sebagai contoh berdasarkan tabel next, lalu mengulangi langkah no. 2 sampai *pattern* atau pola susunan kata yang dijadikan sebagai contoh berada di ujung teks.

Sehingga prinsip dari algortima *Knuth Morris Pratt* ini yakni :

Contoh penggunaan algortima *Knuth Morris Pratt* Dalam melakukan pencarian dalam teks.

*String* : RISALAH

*Pattern* : SALAH

Langkah – Langkah :

- Langkah I : *Pattern* tidak cocok dengan *string*. Maka *pattern* akan bergeser satu posisi ke kanan.

Tabel 2. 1 Pergeseran Pertama

<i>String</i>	R	i	s	a	l	a	h
<i>Pattern</i>	S	a	l	a	h		

- Langkah II: *Pattern* tidak cocok dengan *string*. Maka *pattern* akan bergeser satu posisi ke kanan.

Tabel 2. 2 Pergeseran Kedua

<i>String</i>	R	i	s	a	l	a	h
<i>Pattern</i>		S	a	l	a	h	

- Langkah III: *Pattern* cocok dengan *string*. Karena ada kecocokan, maka algoritma *Knuth Morris Pratt* akan menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* dengan *string*.

Tabel 2. 3 Pergeseran Ketiga

<i>String</i>	R	i	s	a	l	a	h
<i>Pattern</i>			S	a	l	a	h

- Langkah IV : *Pattern* cocok dengan *string*. Karena ada kecocokan, maka algoritma *Knuth Morris Pratt* akan menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan

pencocokan *pattern* dengan *string*.

Tabel 2. 4 Pergeseran Keempat

<i>String</i>	R	i	s	a	l	a	h
<i>Pattern</i>			S	a	l	a	h

- e. Langkah V : *Pattern* cocok dengan *string*. Karena ada kecocokan, maka algoritma *Knuth Morris Pratt* akan menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* dengan *string*.

Tabel 2. 5 Pergeseran Kelima

<i>String</i>	R	i	s	a	l	a	h
<i>Pattern</i>			S	a	l	a	h

- f. Langkah VI : *Pattern* cocok dengan *string*. Karena ada kecocokan, maka algoritma *Knuth Morris Pratt* akan menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* dengan *string*.

Tabel 2. 6 Pergeseran Keenam

<i>String</i>	R	i	s	a	l	a	h
<i>Pattern</i>			S	a	l	a	h

- g. Langkah VII : *Pattern* cocok dengan *string*. Karena ada kecocokan, maka algoritma *Knuth Morris Pratt* akan menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* dengan *string*. Namun karena jumlah *pattern* hanya 6 huruf, maka pencarian akan dihentikan dan diperoleh hasil bahwa *pattern* P terdapat kecocokan dengan *string* S sebesar 100 persen.

Tabel 2. 7 Pergeseran Ketujuh

String	R	i	s	a	l	a	h
Pattern			S	a	l	a	h

### 2.2.2 Kelebihan dan Kekurangan

Algoritma ini pun memiliki kelebihan dan kekurangan yaitu sebagai berikut :

#### 1. Kelebihan

Kelebihan dari algoritma Knuth-Morris-Pratt selain cepat juga sangat baik digunakan pada file berukuran besar karena pencarian kecocokan tidak perlu kembali ke belakang pada input teks.

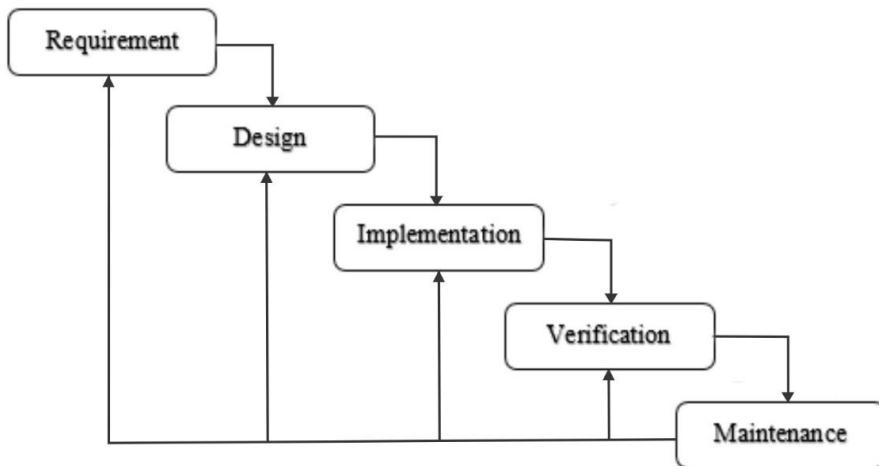
#### 2. kekurangan yakni efektifitas dari algoritma ini akan berkurang seiring dengan bertambahnya jumlah jenis karakter dari teks.

### 2.3 Metode Waterfall

Model air terjun (Waterfall Model) adalah metode pertama yang banyak digunakan dalam industri perangkat lunak. model ini merupakan pendekatan tradisional, dan jauh kurang fleksibel daripada metodologi gesit dengan pengembangan dipecah menjadi sprint tunggal, tetapi dapat dilengkapi dengan loop umpan balik dan loopback. Saat ini masih digunakan dalam berbagai versi jika persyaratan dan karakteristik suatu perangkat lunak dapat didefinisikan dengan jelas selama fase konseptual.

Metode air terjun atau yang sering disebut metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan

pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan. Tahapan metode *waterfall* dapat dilihat pada gambar di bawah ini.



Gambar 2. 1 Metode Waterfall

#### A. Tahapan Metode Waterfall

Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang berurut yaitu: *requirement* (analisis kebutuhan), *design system* (desain sistem), *Coding* (pengkodean) & *Testing* (pengujian), Penerapan Program, pemeliharaan. Tahapan tahapan dari metode *waterfall* adalah sebagai berikut :

##### 1. *Requirement Analysis*

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei

langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

## 2. *System Design*

Spesifikasi kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain Sistem membantu dalam menentukan perangkat keras(*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

## 3. *Implementation*

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut *unit*, yang terintegrasi dalam tahap selanjutnya. Setiap *unit* dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai *unit testing*.

## 4. *Integration & Testing*

Seluruh *unit* yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing *unit*. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

## 5. *Operation & Maintenance*

Tahap akhir dalam model *waterfall*. Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi *unit* sistem dan peningkatan jasa sistem sebagai kebutuhan baru

## B. Kelebihan Dan Kekurangan Metode Waterfall

1. Kelebihan menggunakan metode air terjun (*waterfall*) adalah metode ini memungkinkan untuk departementalisasi dan kontrol. proses pengembangan model fase *one by one*, sehingga meminimalis kesalahan yang mungkin akan terjadi. Pengembangan bergerak dari konsep, yaitu melalui desain, implementasi, pengujian, instalasi, penyelesaian masalah, dan berakhir di operasi dan pemeliharaan.
2. Kekurangan menggunakan metode *waterfall* adalah metode ini tidak memungkinkan untuk banyak revisi jika terjadi kesalahan dalam prosesnya. Karena setelah aplikasi ini dalam tahap pengujian, sulit untuk kembali lagi dan mengubah sesuatu yang tidak terdokumentasi dengan baik dalam tahap konsep sebelumnya.

## 2.4 Pengantar Codeigniter



Gambar 2. 2 Codeigniter

*CodeIgniter* adalah aplikasi open source yang merupakan *framework* dengan konsep model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP yang dapat mempercepat

pengembang untuk membuat sebuah aplikasi web. CodeIgniter memudahkan developer untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuatnya dari awal.

Selain ringan dan cepat, CodeIgniter juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih CodeIgniter sebagai framework pilihannya. Karena kelebihan-kelebihan yang dimiliki oleh CodeIgniter, pembuat PHP Rasmus Lerdorf memuji CodeIgniter di frOSCon (Agustus 2008) dengan mengatakan bahwa dia menyukai CodeIgniter karena “it is faster, lighter and the least like a framework.

Framework secara sederhana dapat diartikan kumpulan dari fungsi-fungsi/prosedur-prosedur dan class-class untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang pemrograman, tanpa harus membuat fungsi atau class dari awal.

#### A. Pengertian MVC

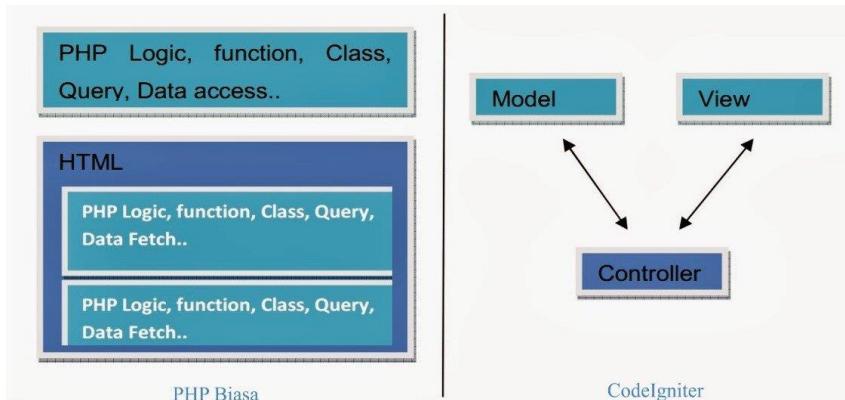
MVC adalah konsep dasar yang harus diketahui sebelum mengenal CodeIgniter. MVC ( Model, View, Controller) merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman Small Talk, yang memisahkan bisnis logic (alur pikir), data logic (penyimpanan data) dan presentation logic (antarmuka aplikasi) atau secara sederhana adalah memisahkan antara desain, data dan proses. Ada 3 komponen yang membangun suatu MVC yaitu :

- Model, biasanya berhubungan dengan data dan interaksi ke database atau webservice. Model juga merepresentasikan struktur data dari

aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks, file XML maupun webservice. Biasanya didalam model akan berisi class dan fungsi untuk mengambil, melakukan update dan menghapus data website. Sebuah aplikasi web biasanya menggunakan basis data dalam menyimpan data, maka pada bagian Model biasanya akan berhubungan dengan perintah-perintah query SQL.

- View, merupakan bagian yang menangani presentation logic. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh controller. View berfungsi untuk menerima dan merepresentasikan data hasil dari model dan controller kepada user. View tidak memiliki akses langsung terhadap bagian model.
- Controller, merupakan bagian yang mengatur hubungan antara bagian model dan bagian view. Pada controller terdapat class-class dan fungsi-fungsi yang memproses permintaan dari View ke dalam struktur data di dalam model. Controller juga tidak boleh berisi kode untuk mengakses basis data Karena tugas megakses data telah diserahkan kepada model. Tugas controller adalah menyediakan berbagai variable yang akan ditampilkan di view, memanggil model untuk melakukan akses ke basis data, menyediakan penanganan kesalahan/error, mengerjakan proses logika dari aplikasi serta melakukan validasi atau cek terhadap input.

## Perbandingan PHP Biasa dengan CodeIgniter



Gambar 2. 3 Perbandingan PHP Biasa Dengan CI

PHP native atau php murni adalah php yang digunakan oleh programmer merupakan penulisan yang dibuat sendiri. Pembuat program pada php murni ini membuat sendiri struktur program sesuai dengan konsep program yang diinginkannya. Sebagai contoh pada php murni untuk insert data ke database kita harus menggunakan perintah insert lengkap sesuai dengan bahasa SQL.

Sedangkan pada framework codeigniter untuk menyelesaikannya kita membutuhkan model dan view karena pada script contoh insert dengan codeigniter di atas biasanya di letakkan pada model. Pada php native terlihat bahwa pembuat program lebih biasa membuat program terpisah satu persatu, mulai dari membuat form, membuat file proses dan membuat file querinya, dan hal ini biasanya ditulis berurut untuk menyelesaikan setiap logika pemrograman. Dengan metode ini pengembangan web dinilai masih dengan cara lama dan seringkali kurang efisien dalam menyelesaikan alur program yang lebih kompleks.

## **BAB III**

---

### **PENJELASAN TOOLS DAN BAHASA PEMOGRAMAN YANG DIGUNAKAN**

#### **3.1 Tools Yang Digunakan**

##### **3.1.1 Sublime**

Sublime Text mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur syntax highlight hampir di semua bahasa pemrogramman yang didukung ataupun dikembangkan oleh komunitas seperti; C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile and XML. Biasanya bagi bahasa pemrograman yang didukung ataupun belum terdukung secara default dapat lebih dimaksimalkan atau didukung dengan menggunakan add-ons yang bisa didownload sesuai kebutuhan user.

Sublime Text bukanlah aplikasi opensource dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, namun beberapa fitur pengembangan fungsionalitas (packages) dari aplikasi ini merupakan hasil dari temuan dan memperoleh dukungan penuh dari komunitas serta memiliki lisensi aplikasi gratis. Aplikasi ini mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur syntax highlight hampir di semua bahasa pemrogramman yang didukung ataupun dikembangkan oleh komunitas.

```
190     const Tensor& value = VariableArgListVariable::value();
191     arg.type = value.dtype();
192     TF_RETURN_IF_ERROR(
193         TensorShapeToXLAShape(value.dtype(), value.shape(), &arg.shape));
194     arg.initialized = true;
195 } else {
196     // The values of uninitialized variables are not passed as inputs, since
197     // they are meaningless. However, it is legal to assign to a resource
198     // variable for the first time inside the XLA computation, so we do permit
199     // uninitialized variables.
200     arg.initialized = false;
201     arg.type = DT_INVALID;
202     arg.shape = xla::Shape();
203 }
204 ++input_num;
205 }
206 return Status::OK();
207 }
208 } // namespace
209
210 Status XlaCompilationCache::Compile(
211     const XlaCompiler::Options& options, const NameAttrList& function,
212     const ConstantArgs& constant_args, const std::vector<OptionalTensor>& variable_args,
213     OpKernelContext* ctx,
214     const XlaCompiler::CompilationResult** compilation_result,
215     xla::LocalExecutable** executable) {
216     VLOG(1) << "XlaCompilationCache::Compile " << DebugString();
217
218     if (VLOG_IS_ON(2)) {
219         VLOG(2) << "num_inputs=" << ctx->num_inputs()
220             << " num_constant_args=" << num_constant_args
221             << " num_variable_args=" << variable_args.size();
222         for (int i = 0; i < ctx->num_inputs(); i++) {
223             TensorShape shape = ctx->input(i).shape();
224             VLOG(2) << i << ": dtype=" << DatatypeString(ctx->input_dtype(i))
225                 << " present=" << ctx->has_input(i)
226                 << " shape=" << shape.DebugString();
227         }
228         for (const OptionalTensor& variable : variable_args) {
229             VLOG(2) << "variable present=" << variable.present
230                 << " type=" << DatatypeString(variable.value.dtype())
231                 << " shape=" << variable.value.shape().DebugString();
232     }
233 }
```

Gambar 3. 1 Sublime Text

Berikut beberapa fitur yang diunggulkan dari aplikasi Sublime Text:

- Goto Anything

Fitur yang sangat membantu dalam membuka file ataupun menjelajahi isi dari file hanya dengan beberapa keystrokes.

- Multiple Selections

Fitur ini memungkinkan user untuk mengubah secara interaktif banyak baris sekaligus, mengubah nama variabel dengan mudah, dan memanipulasi file lebih cepat dari sebelumnya.

- Command Pallete

Dengan hanya beberapa keystrokes, user dapat dengan cepat mencari fungsi yang diinginkan, tanpa harus menavigasi melalui menu.

- Distraction Free Mode

Bila user memerlukan fokus penuh pada aplikasi ini, fitur ini dapat membantu user dengan memberikan tampilan layar penuh.

- Split Editing
- Dapatkan hasil yang maksimal dari monitor layar lebar dengan dukungan editing perpecahan. Melakukan editing di sisi file dengan sisi, atau mengedit dua lokasi di satu file. Anda dapat mengedit dengan banyak baris dan kolom yang user inginkan.
- Instant Project Switch  
Mengambil seluruh file yang dimasukkan kedalam project pada aplikasi ini terhubung dengan fitur Goto Anything untuk menjelajahi semua file yang ada ataupun untuk beralih ke file dalam project lainnya dengan cepat.
- Plugin API  
Aplikasi ini memiliki plugin API berbasis Phyton sehingga membuat aplikasi ini sangat tangguh.
- Customize Anything  
Aplikasi sublime mempunyai dan memberikan user fleksibilitas dalam hal pengaturan fungsional dalam aplikasi ini.
- Cross Platform  
Aplikasi ini dapat berjalan hampir disemua operating system modern seperti Windows, OS X, dan Linux based operating system.

### 3.1.2 Xampp

XAMPP merupakan salah satu paket installasi Apache, PHP dan MySQL instant yang dapat kita gunakan untuk membantu proses installasi ketiga produk tersebut. Selain paket installasi instant XAMPP versi 1.6.4 juga memberikan fasiltias pilihan pengunaan PHP4 atau PHP5.Untuk berpindah versi PHP yang ingin digunakan juga sangat mudah dilakukan dengan menggunakan bantuan PHP-Switch yang telah disertakan oleh

XAMPP, dan yang terpenting XAMPP bersifat free atau gratis untuk digunakan.

XAMPP merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web server Apache, PHP dan MySQL secara manual. XAMPP akan menginstalasi dan mengkonfigurasikannya secara otomatis. Merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkanya dapat mendownload langsung dari web resminya.

Fungsi XAMPP sendiri adalah sebagai server yang berdiri sendiri (localhost), yang terdiri beberapa program antara lain : Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP sendiri merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl.

### **3.1.3 Framework Codeigniter**

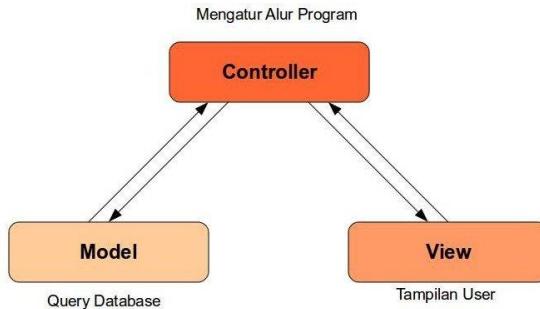
CodeIgniter pertamakali dikembangkan pada tahun 2006 oleh Rick Ellis. Dengan logo api yang menyala, CodeIgniter dengan cepat “membakar” semangat para web developer untuk mengembangkan web dinamis dengan cepat dan mudah menggunakan framework PHP yang satu ini.

Dalam situs resmi codeigniter, menyebutkan bahwa codeigniter merupakan framework PHP yang kuat dan sedikit bug. Codeigniter ini dibangun untuk para pengembang dengan bahasa pemrogram PHP yang membutuhkan alat untuk membuat web dengan fitur lengkap. kelebihan

dari framework codeigniter jika dibandingkan dengan framework lain adalah sebagai berikut :

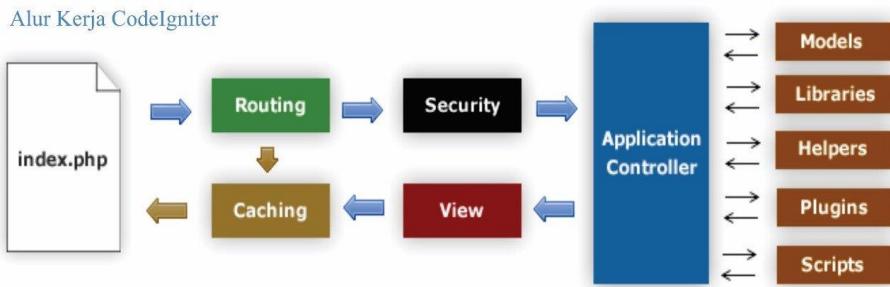
1. Gratis (Open-Source) Kerangka kerja Codeigniter memiliki lisensi dibawah Apache/BSD open-source sehingga bersifat bebas atau gratis.
2. Berukuran kecil Ukuran yang kecil merupakan keunggulan tersendiri jika dibandingkan framework lain yang berukuran besar dan membutuhkan resource yang besar dan juga dalam eksekusi maupun penyimpanannya.
3. Menggunakan konsep M-V-C Codeigniter merupakan konsep M-V-C (ModelView-Controller) yang memungkinkan pemisahan antara layer application-logic dan presentation. Dengan konsep ini kode PHP, query Mysql, Javascript dan CSS dapat saling dipisah-pisahkan sehingga ukuran file menjadi lebih kecil dan lebih mudah dalam perbaikan kedepannya atau maintenance.
  - a. Model Kode merupakan program (berupa OOP class) yang digunakan untuk berhubungan dengan database MySQL sekaligus untuk memanipulasinya (input-edit-delete).
  - b. View Merupakan kode program berupa template atau PHP untuk menampilkan data pada browser.
  - c. Controller merupakan Kode program (berupa OOP class ) yang digunakan untuk mengontrol aliran atau dengan kata lain sebagai pengontrol model dan view.

Adapun alur dari program aplikasi berbasis codeigniter yang menggunakan konsep M-V-C ditunjukkan pada gambar berikut :



Gambar 3. 2 Konsep Aliran MVC

### 3.1.4 Cara Kerja



Gambar 3. 3 Alur Kerja Codeigniter

- Index.php: Index.php disini berfungsi sebagai file pertama dalam program yang akan dibaca oleh program.
- The Router: Router akan memeriksa HTTP request untuk menentukan hal apa yang harus dilakukan oleh program.
- Cache File: Apabila dalam program sudah terdapat “cache file” maka file tersebut akan langsung dikirim ke browser. File cache inilah yang dapat membuat sebuah website dapat di buka dengan lebih cepat. Cache file dapat melewati proses yang sebenarnya harus dilakukan oleh program codeigniter.

- Security: Sebelum file controller di load keseluruhan, HTTP request dan data yang disubmit oleh user akan disaring terlebih dahulu melalui fasilitas security yang dimiliki oleh codeigniter.
- Controller: Controller akan membuka file model, core libraries, helper dan semua resources yang dibutuhkan dalam program tersebut.
- View: Hal yang terakhir akan dilakukan adalah membaca semua program yang ada dalam view file dan mengirimkannya ke browser supaya dapat dilihat. Apabila file view sudah ada yang di “cache” maka file view baru yang belum ter-cache akan mengupdate file view yang sudah ada.

## Contoh File untuk Model, View dan Controller

### 1. Model/Data\_tahanan

```

models > Data_tahanan_model.php
1  <?php
2  /*
3  * Generated by CRUDigniter v3.2
4  * www.crudigniter.com
5  */
6
7 class data_tahanan_model extends CI_Model
8 {
9     function __construct()
10    {
11        parent::__construct();
12        $this->load->library('session');
13    }
14
15    /*
16    * Get data_tahanan by id
17    */
18    function get_data_wbp($id)
19    {
20        $this->db->order_by('data_tahanan.id', 'desc');
21        if(isset($params) && !empty($params))
22        {
23            $this->db->limit($params['limit'], $params['offset']);
24        }
25        $this->db->select('*', data_tahanan.id AS id_wbp');
26        $this->db->from('data_tahanan');
27        $this->db->join('pasal', 'data_tahanan.pasal_id = pasal.id');
28        $this->db->join('user', 'data_tahanan.post_by = user.id');
29        $this->db->where('data_tahanan.id', $id);
30        return $this->db->get()->row_array();
31    }
32
33    function getKategori(){
34        $tes = $this->db->get('pasal');
35        return $tes;
36    }
37
38    /*
39    * Get all data_tahanan count

```

Gambar 3. 4 Model

## 2. View/data\_wbp

```
views > data_wbp > index.php
 1  <div class="row">
 2    <div class="col-md-12">
 3      <div class="box">
 4        <div class="box-header">
 5          <h3 class="box-title"><?= $title ?></h3>
 6          <div class="box-tools">
 7            | <!-- <a href=<?php // echo site_url('data_wbp/add'); ?>" class="btn btn-su
 8            | </div>
 9        </div>
10      <div class="box-body">
11        <table class="table table-striped table-bordered">
12          <tr>
13            <th>ID</th>
14            <th>Image</th>
15            <th>Nama</th>
16            <th>Pasal</th>
17            <th>Di Input Oleh</th>
18            <th>Alamat</th>
19            <!-- <th>Actions</th> -->
20          </tr>
21          <php foreach($data_tahanan as $b){ ?>
22            <tr>
23              <td><?= $b['id_wbp']; ?></td>
24              <td><img src=<?= base_url($b['image']); ?>" width="150px"></td>
25              <td><?= $b['nama_tahanan']; ?></td>
26              <td><?= $b['nama_pasal']; ?></td>
27              <td><?= $b['nama']; ?></td>
28              <td><?= $b['alamat_tahanan']; ?></td>
29              <!-- <td>
30                | <a href=<?php // echo site_url('data_wbp/edit/'.$b['id_wbp']); ?>
31                | <a href=<?php // echo site_url('data_wbp/remove/'.$b['id_wbp']); ?>
32              </td> -->
33            </tr>
34            <?php } ?>
35          </table>
36          <div class="pull-right">
37            | <?php echo $this->pagination->create_links(); ?>
38          </div>
39        </div>
40      </div>
```

Gambar 3. 5 View

### 3. Controller/data\_wbp

```
controllers > Data_wbp.php
1  <?php
2
3
4  class Data_wbp extends CI_Controller{
5      function __construct()
6      {
7          parent::__construct();
8          $this->load->model('Data_tahanan_model');
9          $this->load->model('Pertanyaan_model');
10     }
11
12
13     function index($tipe) //menampilkan data data tahanan
14     {
15         $params['limit'] = 9;
16         $params['offset'] = ($this->input->get('per_page')) ? $this->input->get('per_page') : 0;
17
18         $config = $this->config->item('pagination');
19         $config['base_url'] = site_url('data_wbp/index?');
20         $config['total_rows'] = $this->Data_tahanan_model->get_all_data_wbp_count();
21         $this->pagination->initialize($config);
22
23         $data['data_tahanan'] = $this->Data_tahanan_model->get_all_data_wbp($params, $tipe);
24         if ($tipe == 'lost'){
25             $data['title'] = 'Data Tahanan';
26         } else {
27             $data['title'] = 'Data Tahanan';
28         }
29
30         $data['_view'] = 'data_wbp/index';
31         $this->load->view('layouts/main',$data);
32     }
33
34
35     function add() // input data wbp tahanan
36     {
37         $this->load->library('form_validation');
38
39         $this->form_validation->set_rules('pasal_id','Pasal Id','required');
40         $this->form validation->set rules('nama_tahanan','Nama Tahanan','required');
```

Gambar 3. 6 Controller

# BAB IV

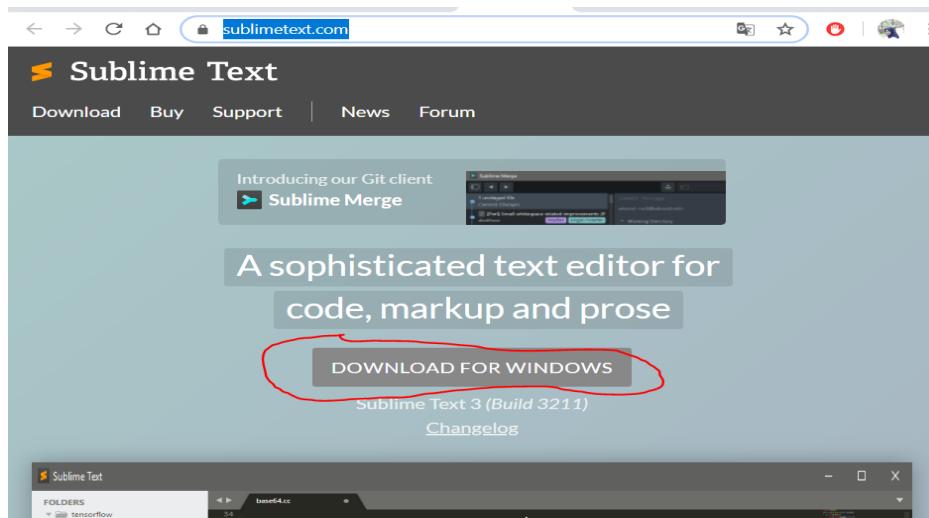
## INSTALASI TOOLS YANG DIGUNAKAN

### 4.1 Tools Yang Digunakan

#### 4.1.1 Instalasi Sublime

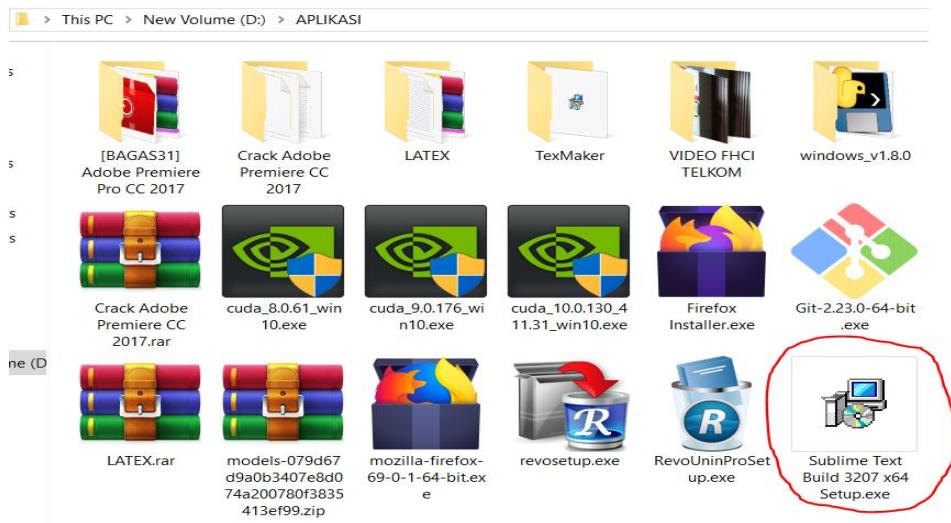
Berikut adalah langkah-langkah untuk melakukan instalasi Sublime :

1. Tahapan yang pertama untuk melakukan instalasi maka download dahulu file .exe pada link berikut :  
<https://www.sublimetext.com/>.



Gambar 4. 1 Download file .exe

2. Tahapan yang kedua setelah selesai mendownload maka buka directory dimana file .exe tersimpan, kemudian Double klik file .exe hasil download tadi :



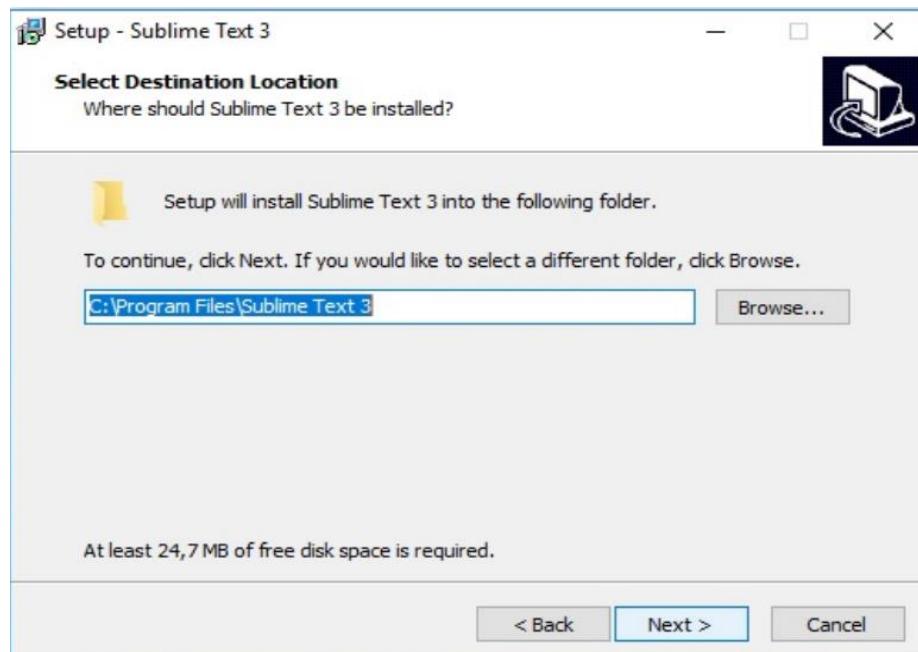
Gambar 4. 2 Double Klik File .exe

3. Tahapan ketiga jika sudah double klik file .exe maka akan muncul tampilan berikut kemudian klik next :



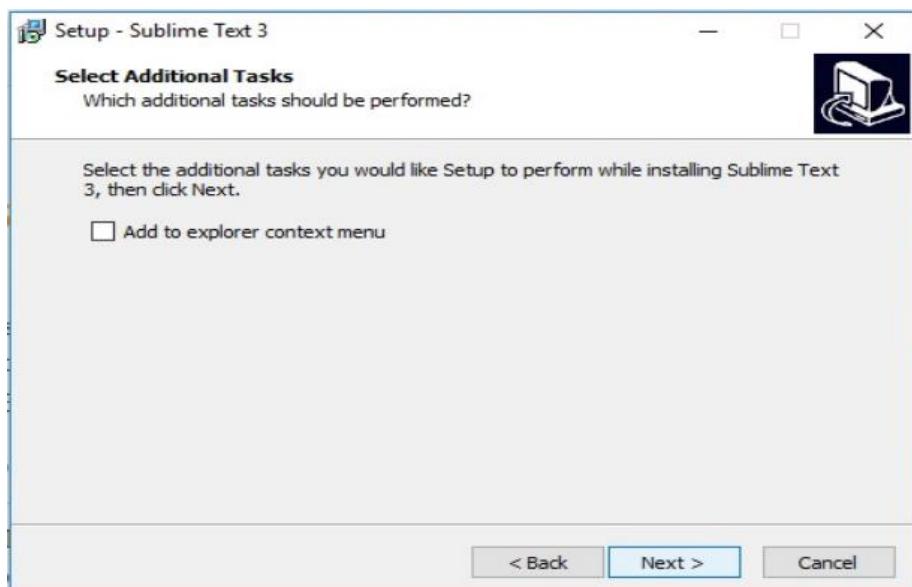
Gambar 4. 3 Proses Instalasi Sublime

4. Selanjutnya klik next



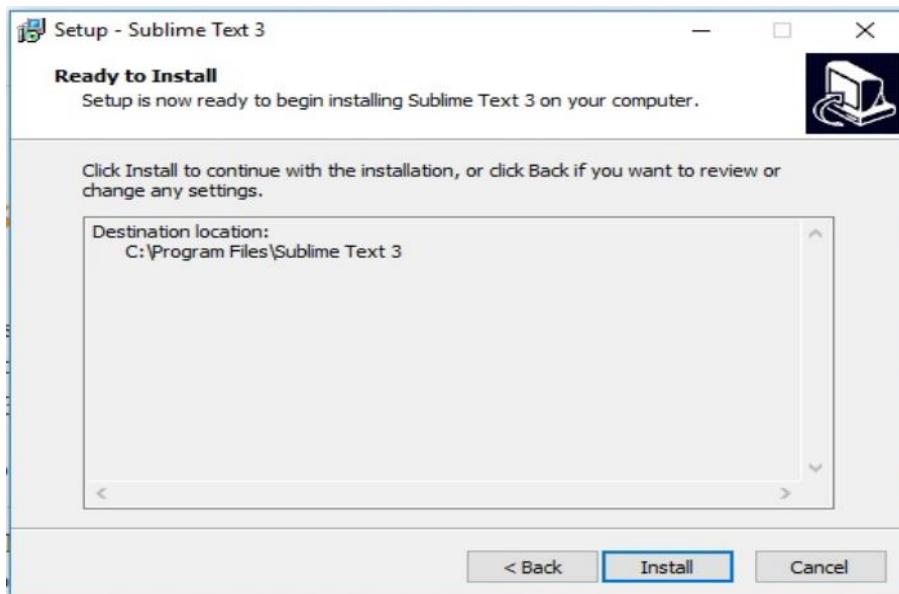
Gambar 4. 4 Proses Instalasi Sublime

5. Selanjutnya klik next



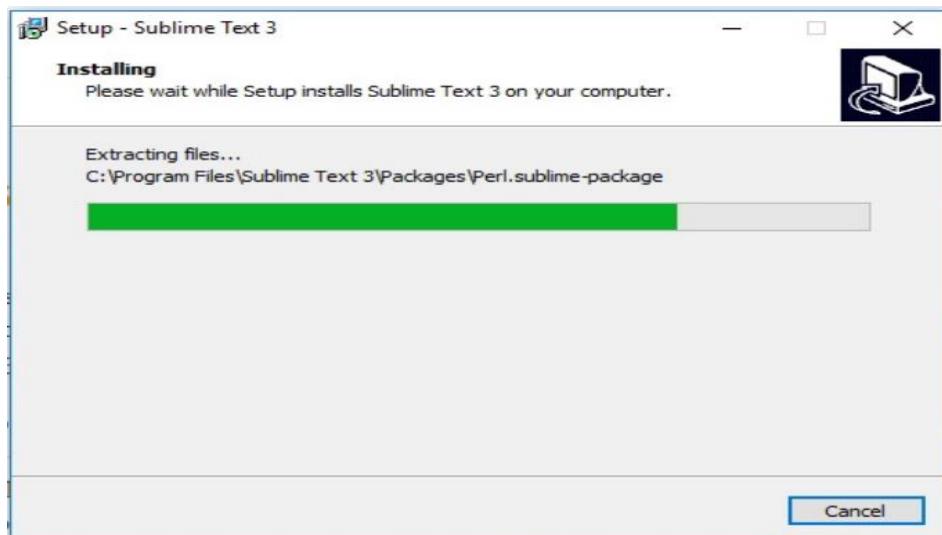
Gambar 4. 5 Proses Instalasi Sublime

6. Selanjutnya klik next



Gambar 4. 6 Proses Instalasi Sublime

7. Tunggu sampai proses instalasi selesai, Setelah proses ini selesai maka sublime telah bisa digunakan.

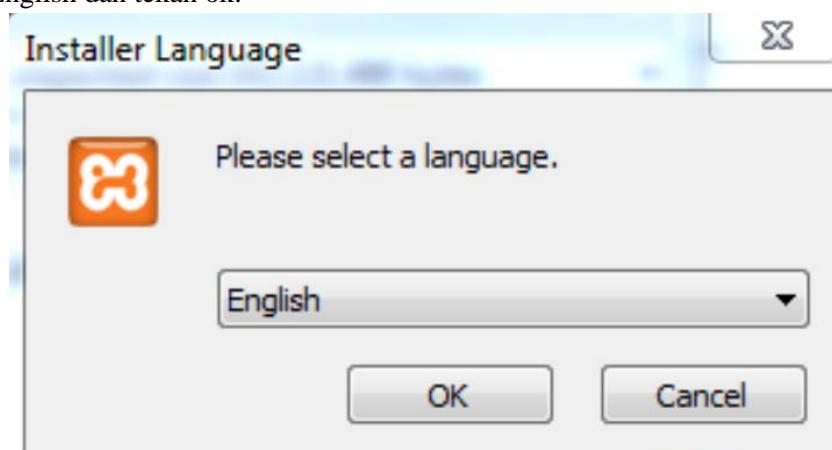


Gambar 4. 7 Proses Instalasi Sublime

#### 4.1.2 Instalasi Xampp

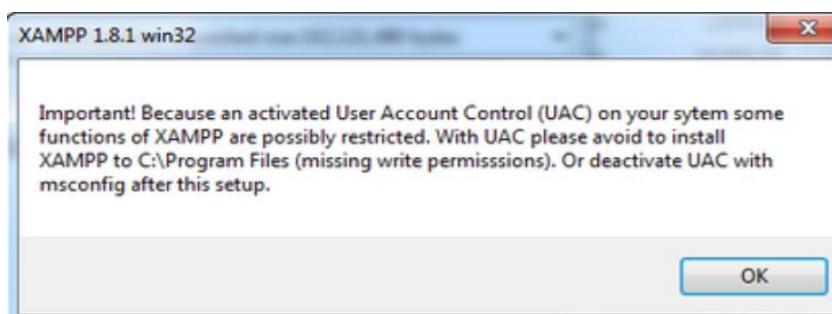
Berikut adalah langkah-langkah untuk melakukan instalasi Xampp:

1. Untuk melakukan instalasi Xampp maka pertama-tama melakukan download file .exe pada link berikut ini :  
<https://www.apachefriends.org/download.html>  
Sesuai kan dengan system operasi yang anda gunakan .
2. Dobel klik file XAMPP yang baru saja Anda download, nanti selanjutnya akan muncul jendela “installer language” seperti di bawah ini lalu pilih English dan tekan ok:



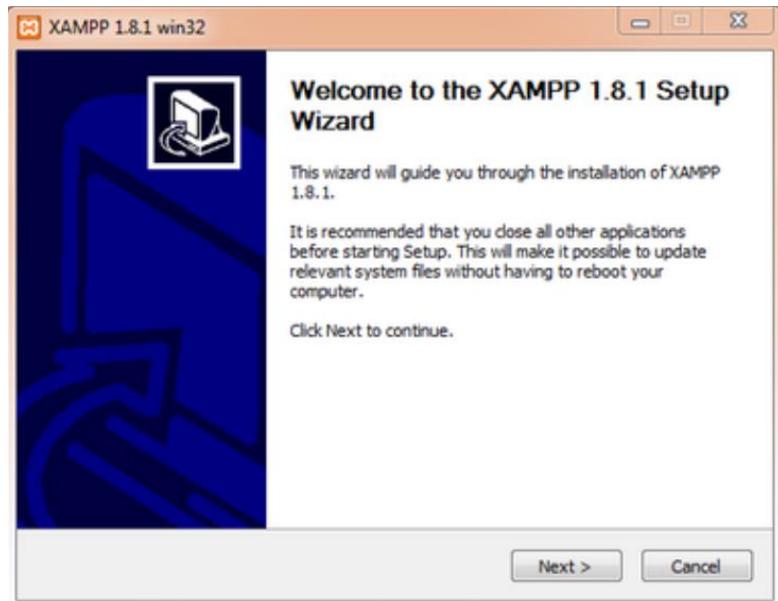
Gambar 4. 8 Instalasi Xampp

3. Kadang pada proses ini muncul pesan error. Jika ada, abaikan saja dan lanjutkan proses instalasi klik ok dan yes.



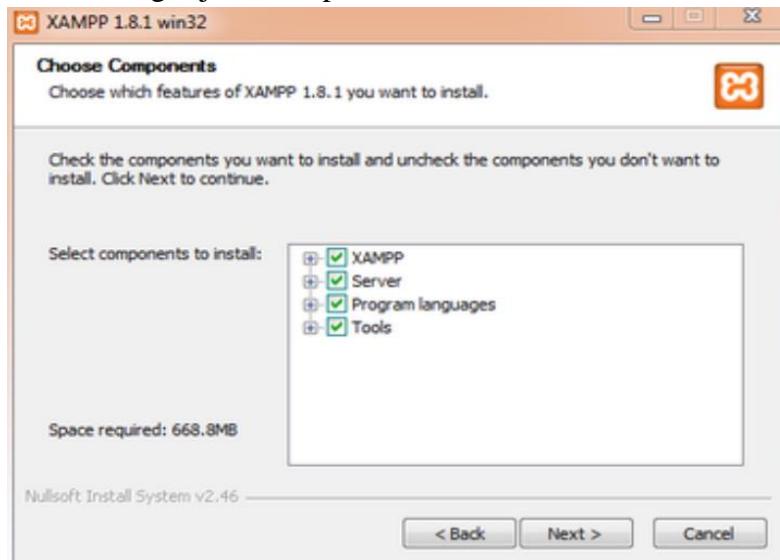
Gambar 4. 9 langkah instalasi

4. Berikutnya akan muncul jendela yang isinya meminta Anda menutup semua aplikasi yang sedang berjalan. Jika semua aplikasi sudah ditutup, maka klik tombol Next.



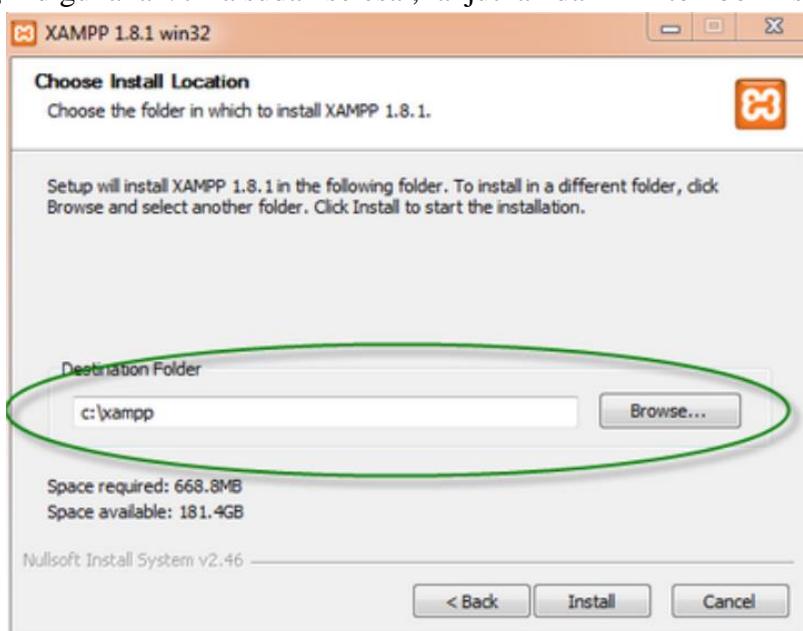
Gambar 4. 10 Langkah Instalasi

5. Selanjutnya Anda akan diminta untuk memilih aplikasi yang mau diinstal. Centang saja semua pilihan dan klik tombol Next.



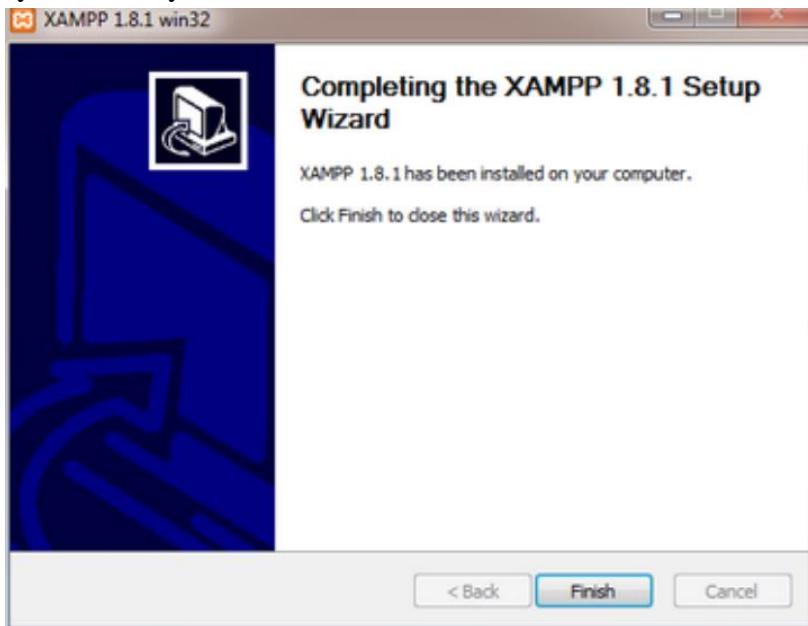
Gambar 4. 11 Langkah Instalasi

6. Kemudian Anda akan diminta untuk menentukan lokasi folder penyimpanan file-file dan folder XAMPP. Secara default akan diarahkan ke lokasi c:\xampp. Namun jika Anda ingin menyimpannya di folder lain bisa klik browse dan tentukan secara manual folder yang ingin digunakan. Jika sudah selesai, lanjutkan dan klik tombol Install.



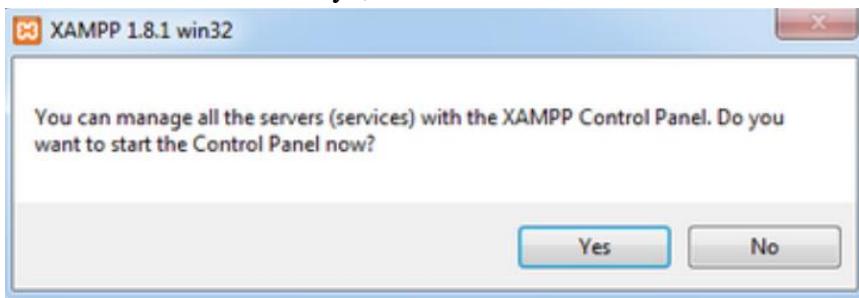
Gambar 4. 12 Menetukan lokasi folder xampp

7. Tunggu beberapa menit hingga proses instalasi selesai. Jika sudah muncul jendela seperti di bawah ini, klik tombol Finish untuk menyelesaiakannya.



Gambar 4. 13 Instalasi Selesai

8. Berikutnya, akan muncul jendela dialog seperti gambar di bawah ini yang menanyakan Anda apakah mau langsung menjalankan aplikasi XAMPP atau tidak. Jika ya, maka klik YES.

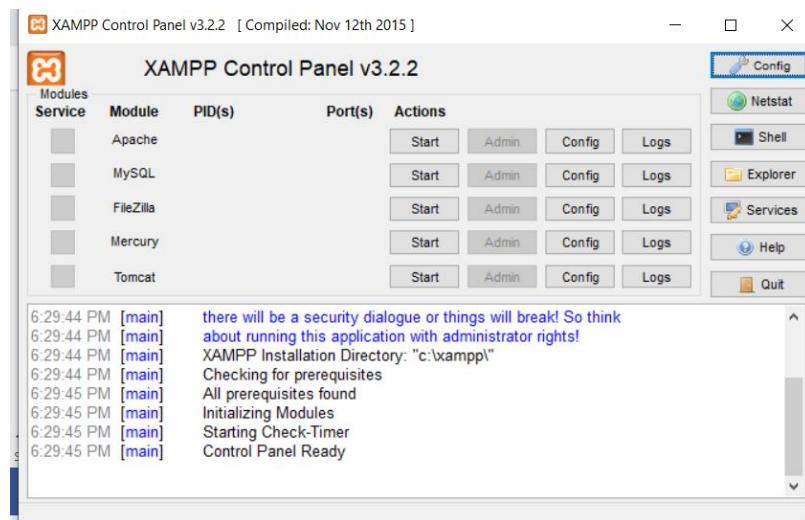


Gambar 4. 14 Jendela dialog

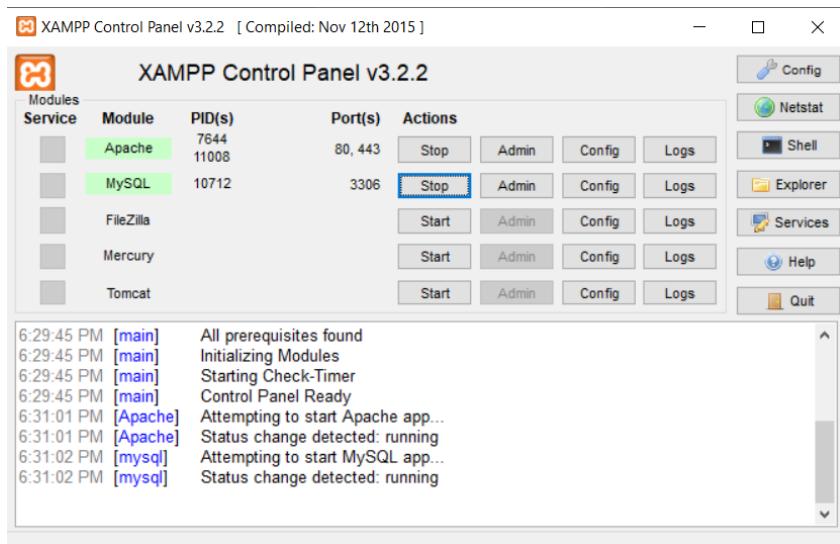
#### 4.1.3 Cara Menjalankan Xampp

Berikut adalah langkah-langkah untuk menjalankan Xampp :

1. Buka aplikasi XAMPP, bisa melalui Start Menu atau Desktop, dan klik icon XAMPP. Atau, jika Anda membukanya begitu proses instalasi selesai maka klik Yes seperti yang terlihat pada gambar di atas.
2. Setelah terbuka, silahkan klik tombol Start pada kolom Action sehingga tombol tersebut berubah menjadi Stop. Dengan mengklik tombol tersebut, artinya itulah aplikasi yang dijalankan. Biasanya jika saya menggunakan XAMPP, yang saya start hanyalah aplikasi Apache dan MySQL, karena saya tidak memerlukan aplikasi seperti Filezilla, dan lain-lain.



Gambar 4. 15 klik tombol Start



Gambar 4. 16 Menjalankan Xampp

3. Sekarang browser kesukaan, dan coba ketikkan <http://localhost/dashboard> di address bar. Jika muncul tampilan seperti gambar di bawah ini, instalasi telah berhasil.



Gambar 4. 17 Xampp dijalankan

#### 1.1.4 Instalasi Codeigniter

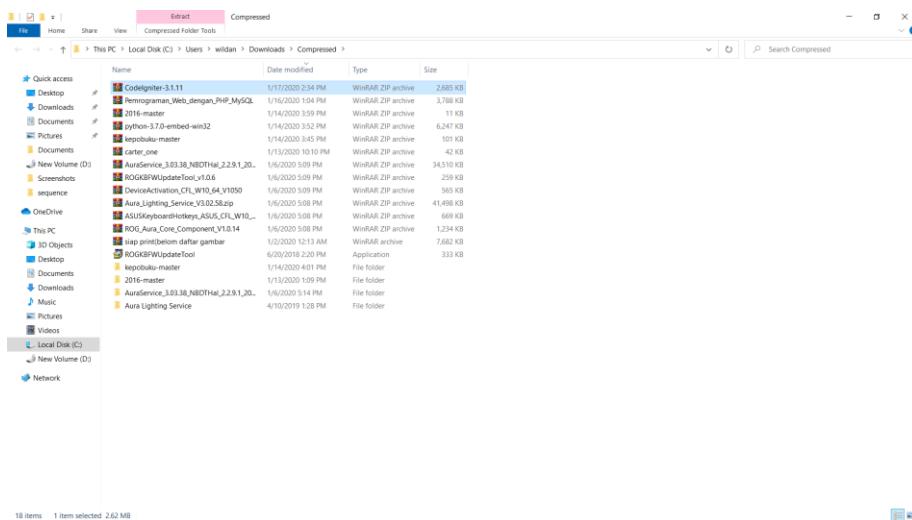
Berikut adalah langkah-langkah untuk melakukan instalasi Codeigniter:

##### 1 Langkah pertama yaitu Bukalah website <https://codeigniter.com/>.



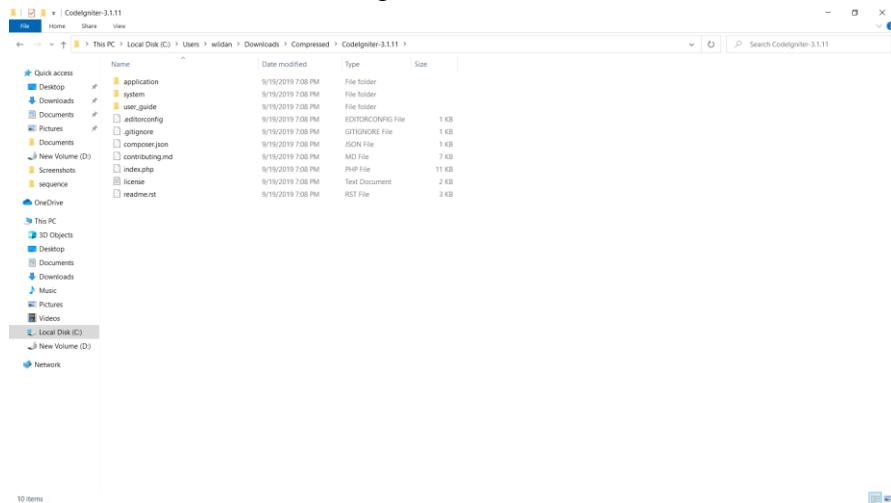
Gambar 4. 18 Download File Codeigniter

##### 2. Setelah didownload sudah selesai lakukan extract file codeigniter.



Gambar 4. 19 Extract file Codeigniter

### 3. Hasil dari extract file codeigniter



Gambar 4. 20 Extract file Codeigniter

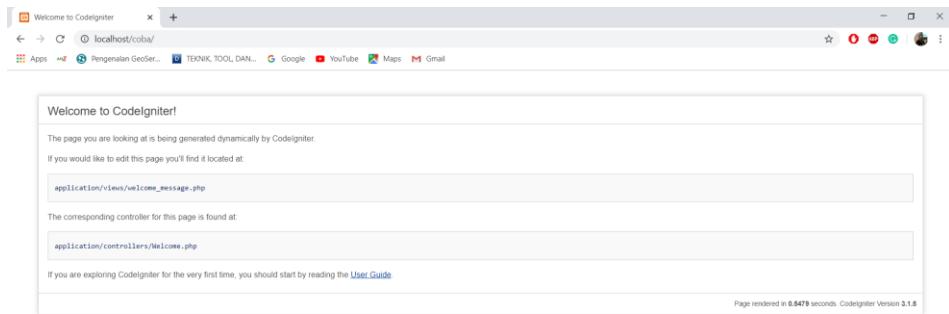
### 4. Pindahkan file tersebut kedalam folder C:\xampp\htdocs

This PC > Local Disk (C:) > xampp > htdocs

Name	Date modified	Type	Size
application	10/22/2019 12:42 ...	File folder	
codeigniter_v3	1/24/2020 6:32 PM	File folder	
crud	10/22/2019 12:43 ...	File folder	
dashboard	10/21/2019 8:23 PM	File folder	
img	10/21/2019 8:23 PM	File folder	
kepulangan	12/22/2019 3:10 PM	File folder	
kepulangan_server	12/2/2019 1:37 AM	File folder	
system	10/22/2019 12:42 ...	File folder	
tokobuah	10/22/2019 2:20 A...	File folder	
webalizer	10/21/2019 8:23 PM	File folder	
xampp	10/21/2019 8:23 PM	File folder	
applications	8/27/2019 9:02 PM	Chrome HTML Do...	4 KB
bitnami	8/27/2019 9:02 PM	Cascading Style Sh...	1 KB
codeigniter_v3	10/22/2019 12:42 ...	WinRAR archive	289 KB
contributing.md	6/19/2017 3:33 PM	MD File	7 KB
favicon	7/16/2015 10:32 PM	Icon	31 KB
index.php	7/16/2015 10:32 PM	PHP File	1 KB
KunjunganRutan1Bandung (5)	12/12/2019 2:22 PM	WinRAR ZIP archive	34,849 KB

Gambar 4. 21 Hasil Extract file Codeigniter

5. Lalu rename dengan sesuai kebutuhan, disini saya merubah menjadi “coba”



---

Gambar 4. 22 Menjalankan Codeigniter

# BAB V

---

## TUTORIAL PEMBUATAN APLIKASI

### 5.1 Tutorial Pembuatan Aplikasi

Sebelum melakukan tutorial penulis akan menjelaskan sedikit pemaparan latar belakang mengapa Aplikasi Verifikasi Kepulangan Warga Binaan Pemasyarakatan Menggunakan Algoritma *Knuth Morris Pratt* ini dibuat dan hasil yang diperoleh dari aplikasi yang sudah berhasil dibuat dan diuji. Salah satu permasalahan pada lapas yaitu dengan adanya verifikasi kepulangan tahanan tentunya tidak bisa dilupakan, oleh karena itu dibuatlah aplikasi yang menggunakan knuth morris pratt untuk meningkatkan proses verifikasi kepulangan dan melakukan penyaringan kembali terhadap tahanan yang akan dipulangkan

Knuth Morris Pratt menjadi salah satu metode yang sedang banyak diterapkan dalam String Matching.

Berdasarkan uraian diatas, maka peneliti membuat sebuah sistem Aplikasi Verifikasi Kepulangan Warga Binaan Pemasyarakatan Menggunakan Algoritma *Knuth Morris Pratt*. Bahasa pemograman yang digunakan adalah Framework Codeigniter dan Database MYSQL. Hasil penelitian ini Validasi berdasarkan String Matching yang dibuat dapat Melakukan verifikasi kunjungan secara otomatis berdasarkan rentan kesamaan string, video dan secara real time di webcam. Hasil penelitian menunjukkan bahwa dengan adanya aplikasi tersebut dapat mempermudah dan mempercepat pendaftaran via online dan tidak melupakan tingkat kemanan rutan.

#### 5.1.1 Mempersiapkan Tools

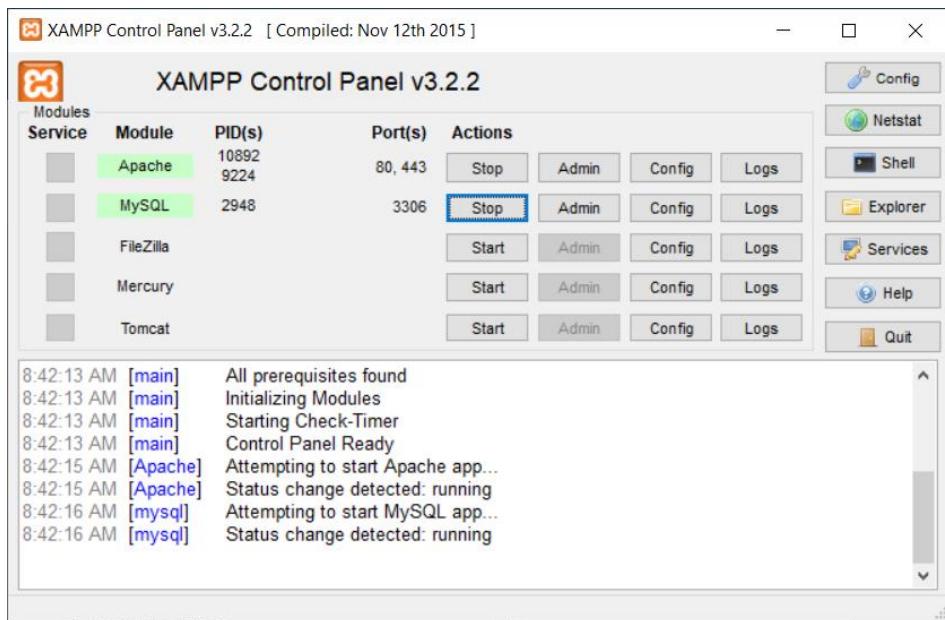
Untuk membuat Aplikasi pendaftaran Online kunjungan rutan maka diperlukan beberapa tools yang dapat mendukung dalam

pembuatannya dan tools ini sudah dibahas pada bab sebelumnya, berikut tools yang harus dipersiapkan dalam pembuatan system object deteksi :

- Xampp
- Sublime
- Codeigniter

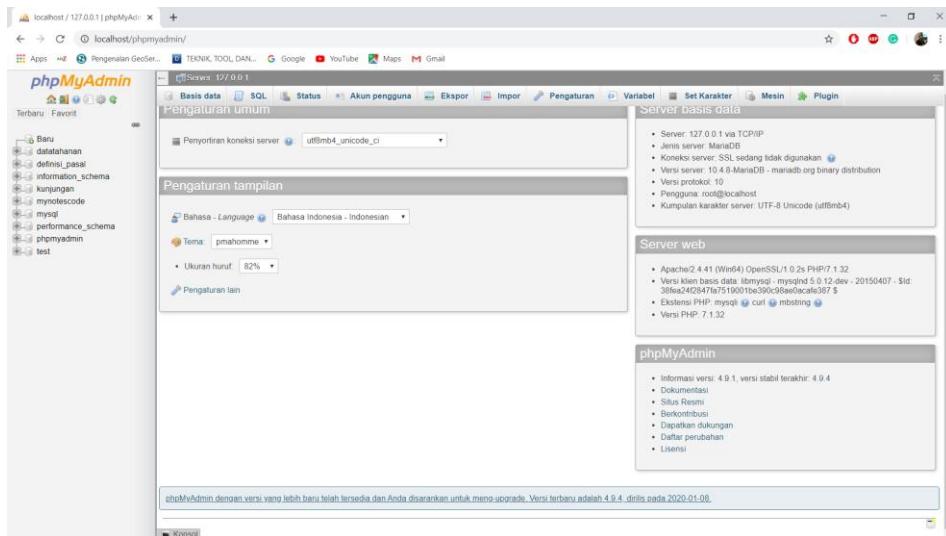
### 5.1.2 Membuat Database

1. Langkah awal ialah membuka aplikasi Xampp dan menjalankan APACHE dan MySQL.



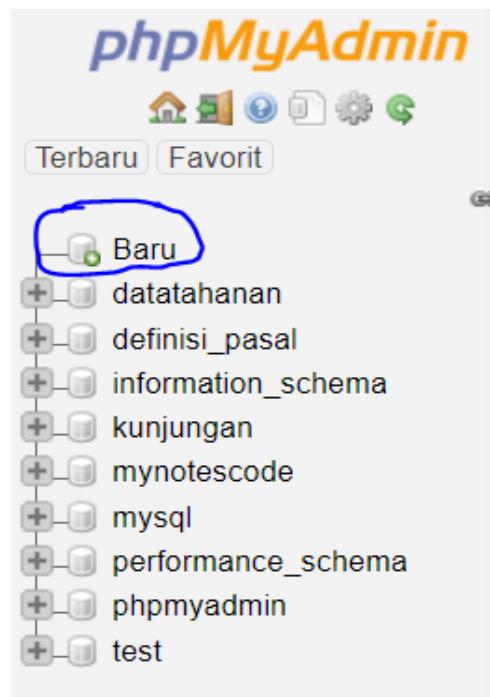
Gambar 5. 1 Xampp

2. Setelah itu buka web browser lalu ketik <http://localhost/phpmyadmin/>



Gambar 5. 2 PhpMyAdmin

3. Kemudian klik new untuk membuat database baru



Gambar 5. 3 Membuat Database

4. Untuk kasus pada kali ini saya memberikan nama database dengan datatahanan

Basis data	Penyortiran	Tindakan
datatahanan	utf8mb4_general_ci	Check privileges
definisi_pasal	utf8mb4_general_ci	Check privileges
information_schema	utf8_general_ci	Check privileges
kunjungan	utf8mb4_general_ci	Check privileges
mynotescode	utf8mb4_general_ci	Check privileges
mysql	utf8mb4_general_ci	Check privileges
performance_schema	utf8_general_ci	Check privileges
phpmyadmin	utf8_bin	Check privileges
test	latin1_swedish_ci	Check privileges
<b>Jumlah: 9</b>	<b>utf8mb4_general_ci</b>	

Gambar 5. 4 Membuat Database datatahanan

Tabel	Tindakan	Baris	Jenis	Penyortiran	Ukuran	Beban
claim	Cari	1	InnoDB	latin1_swedish_ci	64 KB	-
data_tahanan	Cari	2	InnoDB	latin1_swedish_ci	48 KB	-
definisi_pasal	Cari	5	InnoDB	utf8mb4_general_ci	16 KB	-
jawaban	Cari	8	InnoDB	latin1_swedish_ci	64 KB	-
jawaban_post	Cari	714	InnoDB	latin1_swedish_ci	48 KB	-
pasal	Cari	5	InnoDB	latin1_swedish_ci	16 KB	-
pertanyaan	Cari	48	InnoDB	latin1_swedish_ci	32 KB	-
user	Cari	3	InnoDB	latin1_swedish_ci	16 KB	-
<b>8 tabel</b>	<b>Jumlah</b>	<b>778</b>	<b>InnoDB</b>	<b>utf8mb4_general_ci</b>	<b>304 KB</b>	<b>0 B</b>

Gambar 5. 5 Databse datatahanan

Pada database data tahanan terdapat 8 tabel yaitu :

- Claim
- Data\_tahanan
- Definisi\_pasal

- Jawaban
- Jawaban\_post
- Pasal
- Pertanyaan
- User

## 5. Struktur data tabel claim

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	<b>id</b>	int(11)		Tidak	<i>Tidak ada</i>		AUTO_INCREMENT		Ubah  Hapus ▾ Lainnya
2	<b>data_tahanan_id</b>	int(11)		Tidak	<i>Tidak ada</i>				Ubah  Hapus ▾ Lainnya
3	<b>input_oleh</b>	int(11)		Tidak	<i>Tidak ada</i>				Ubah  Hapus ▾ Lainnya
4	<b>konfirmasi_oleh</b>	int(11)		Tidak	<i>Tidak ada</i>				Ubah  Hapus ▾ Lainnya
5	<b>status</b>	int(11)		Tidak	<i>Tidak ada</i>				Ubah  Hapus ▾ Lainnya
6	<b>berkas</b>	text	latin1_swedish_ci	Tidak					Ubah  Hapus ▾ Lainnya
7	<b>keterangan</b>	varchar(40)	latin1_swedish_ci	Tidak	<i>Tidak ada</i>				Ubah  Hapus ▾ Lainnya
8	<b>tanggal_konfirmasi</b>	timestamp		Tidak	current_timestamp()				Ubah  Hapus ▾ Lainnya
9	<b>created_at</b>	datetime		Tidak	<i>Tidak ada</i>				Ubah  Hapus ▾ Lainnya
10	<b>updated_at</b>	timestamp		Tidak	current_timestamp()				Ubah  Hapus ▾ Lainnya

Gambar 5. 6 Struktur Tabel Claim

## 6. Struktur Data Table Data Tahanan

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	<b>Id</b>	int(11)		Tidak	<i>Tidak ada</i>		AUTO_INCREMENT		Ubah  Hapus ▾ Lainnya
2	<b>nama_tahanan</b>	varchar(255)	latin1_swedish_ci	Tidak	<i>Tidak ada</i>				Ubah  Hapus ▾ Lainnya
3	<b>ciri_khusus</b>	text	latin1_swedish_ci	Tidak	<i>Tidak ada</i>				Ubah  Hapus ▾ Lainnya
4	<b>pasal_id</b>	int(11)		Tidak	<i>Tidak ada</i>				Ubah  Hapus ▾ Lainnya
5	<b>alamat_tahanan</b>	text	latin1_swedish_ci	Tidak	<i>Tidak ada</i>				Ubah  Hapus ▾ Lainnya
6	<b>post_by</b>	int(11)		Tidak	<i>Tidak ada</i>				Ubah  Hapus ▾ Lainnya
7	<b>image</b>	varchar(255)	latin1_swedish_ci	Tidak	<i>Tidak ada</i>				Ubah  Hapus ▾ Lainnya
8	<b>berkas</b>	text	latin1_swedish_ci	Tidak	<i>Tidak ada</i>				Ubah  Hapus ▾ Lainnya
9	<b>jenis</b>	varchar(6)	latin1_swedish_ci	Tidak	<i>Tidak ada</i>				Ubah  Hapus ▾ Lainnya
10	<b>status_claim</b>	int(11)		Tidak	<i>Tidak ada</i>				Ubah  Hapus ▾ Lainnya
11	<b>created_at</b>	datetime		Tidak	<i>Tidak ada</i>				Ubah  Hapus ▾ Lainnya
12	<b>updated_at</b>	timestamp		Tidak	current_timestamp()				Ubah  Hapus ▾ Lainnya

Gambar 5. 7 Struktur Tabel Data Tahanan

## 7. Struktur Data table Definisi Pasal

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	<b>id</b>	int(11)		Tidak	<i>Tidak ada</i>		AUTO_INCREMENT		Ubah  Hapus ▾ Lainnya
2	<b>id_pasal</b>	int(11)		Tidak	<i>Tidak ada</i>				Ubah  Hapus ▾ Lainnya
3	<b>isi</b>	text	utf8mb4_general_ci	Tidak					Ubah  Hapus ▾ Lainnya

Gambar 5. 8 Struktur Tabel Define Pasal

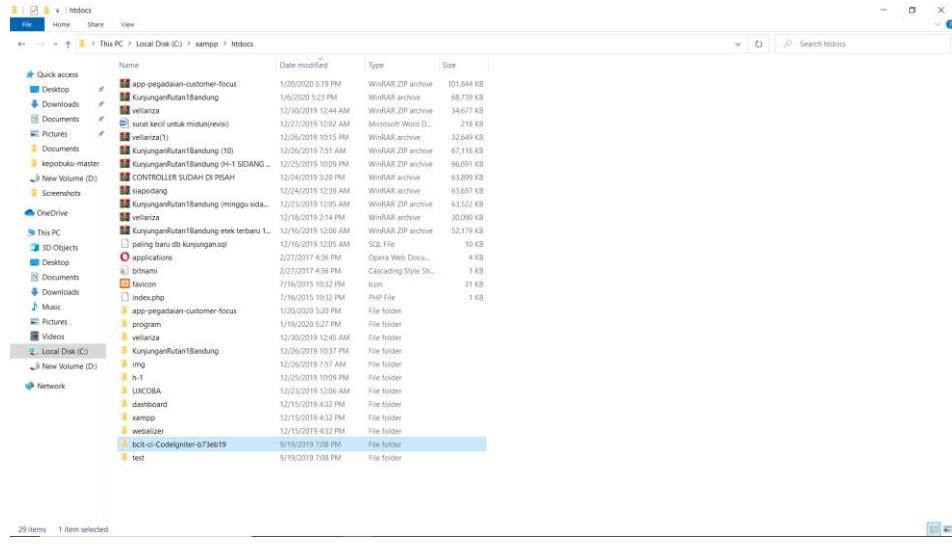


Pada gambar gambar diatas merupakan struktur database yang terdapat pada aplikasi verifikasi kepulangan dan akan dilanjutkan kedalam tahap selanjutnya yaitu configurasi Codeigniter sesuai dengan kebutuhan program yang akan dibuat.

### 5.1.3 Konfigurasi

Pada setiap pembuatan aplikasi harus melakukan configurasi terlebih dahulu supaya program yang kita buat dapat terhubung dengan database dari aplikasi tersebut, berikut merupakan konfigurasi Codeigniter:

1. Copykan folder Codeigniter yang sudah di download di <https://codeigniter.com/en/download> kedalam folder C:\xampp\htdocs



Gambar 5. 14 Langkah Pertama

2. Rename folder codeigniter yang sudah di download menjadi sesuai dengan nama aplikasi yang akan dibangun.

application	10/22/2019 12:42 ...	File folder
coba	1/24/2020 6:32 PM	File folder
crud	10/22/2019 12:43 ...	File folder
dashboard	10/21/2019 8:23 PM	File folder
img	10/21/2019 8:23 PM	File folder
kepulangan	12/22/2019 3:10 PM	File folder
kepulangan_server	12/2/2019 1:37 AM	File folder
system	10/22/2019 12:42 ...	File folder
tokobuah	10/22/2019 2:20 A...	File folder
webalizer	10/21/2019 8:23 PM	File folder
xampp	10/21/2019 8:23 PM	File folder

Gambar 5. 15 Langkah Kedua

3. Buka Tools Sublime, open folder codeigniter yang sudah di rename tadi

```

application
├── images
├── resources
├── system
├── uploads
└── vendor

controller.php
helper.php
model.php
view.php

function add() // input data wbp tahanan
{
    $this->load->library('form_validation');

    $this->form_validation->set_rules('pasal_id','Pasal Id','required');
    $this->form_validation->set_rules('nama_Tahanan','Nama Tahanan','required');
}

```

Gambar 5. 16 Langkah ketiga

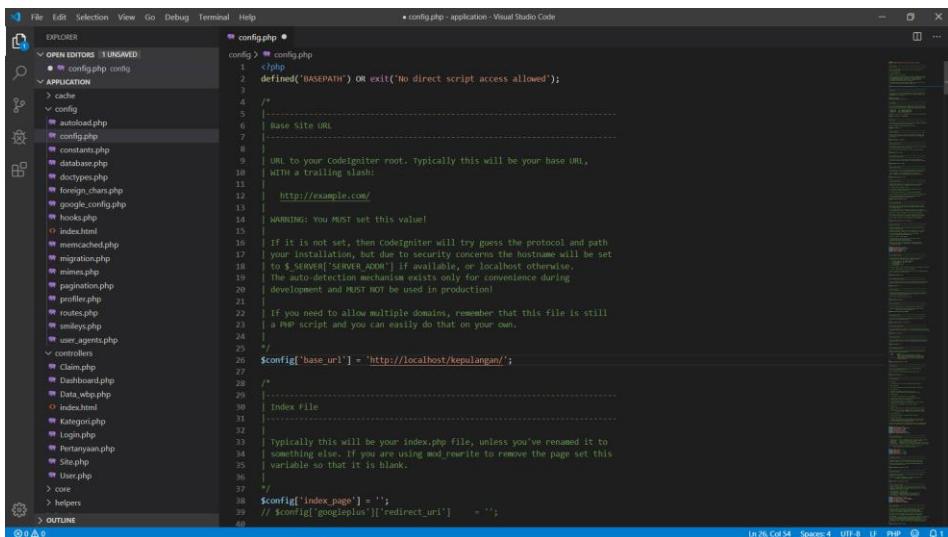
4. Lalu buka config>config.php, pada bagian :

```
$config['base_url'] =
```

isikan menjadi sesuai nama folder codeigniter

```
$config['base_url'] = 'http://localhost/kepulangan/';
```

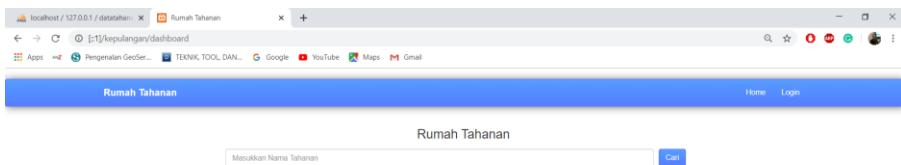
Yang berfungsi sebagai base url yang akan di panggil pada web browser



```
File Edit Selection View Go Debug Terminal Help
EXPLORER OPEN EDITORS 1 UNSAVED config > config.php
APPLICATION config > config.php
cache
config
autoload.php
constants.php
database.php
doctypes.php
foreign_chars.php
google_config.php
hooks.php
index.html
memcached.php
migration.php
mimes.php
pagination.php
profiles.php
routes.php
smileys.php
user_agents.php
controllers
Claim.php
Dashboard.php
Data_wdp.php
index.html
Kategori.php
Login.php
Pertanyaan.php
Site.php
User.php
core
helpers
OUTLINE
config.php
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 /*
5 | Base Site URL
6 |
7 |
8 |
9 | URL to your CodeIgniter root. Typically this will be your base URL,
10 | WITH a trailing slash!
11 |
12 | http://example.com/
13 |
14 | WARNING: You MUST set this value!
15 |
16 | If it is not set, then CodeIgniter will try guess the protocol and path
17 | to your installation, but due to security concerns the hostname will be set
18 | to $_SERVER[ SERVER_ADDR ] if available, or localhost otherwise.
19 | The auto-detection mechanism exists only for convenience during
20 | development and MUST NOT be used in production!
21 |
22 | If you need to allow multiple domains, remember that this file is still
23 | a PHP script and you can easily do that on your own.
24 |
25 */
26 $config['base_url'] = 'http://localhost/kepulangan/';
27
28 /*
29 | Index File
30 |
31 |
32 |
33 | Typically this will be your index.php file, unless you've renamed it to
34 | something else. If you are using mod_rewrite to remove the page set this
35 | variable so that it is blank.
36 |
37 */
38 $config['index_page'] = '';
39 // $config['googleplus']['redirect_url'] = '';
```

Gambar 5. 17 Langkah Keempat

5. Dan hasilnya jika kita memanggil localhost/kepulangan/ pada aplikasi web browser akan muncul tampilan program tersebut



---

Gambar 5. 18 Hasil Pemanggilan

6. Buka config>config.php, pada bagian:  
autoload['libraries']  
isikan menjadi :  
\$autoload['libraries'] = array('database', 'pagination','session', 'upload');  
Konfigurasi pada file ini bertujuan untuk mementukan sumber daya apa yang akan diload secara otomatis.

```

File Edit Selection View Go Debug Terminal Help
autoloader.php - application - Visual Studio Code

EXPLORER config.php autoload.php
config > autoload.php
  | config.php
  | config.php
  | constants.php
  | database.php
  | doctypes.php
  | foreign_chars.php
  | google_config.php
  | index.html
  | index.php
  | memcached.php
  | migration.php
  | mimes.php
  | pagination.php
  | profiler.php
  | routes.php
  | smileys.php
  | user_agents.php
  controllers
    | Cache.php
    | Dashboard.php
    | Data_whisp.php
    | index.html
    | Kategori.php
    | Login.php
    | Pertanyaan.php
    Site.php
    User.php
  core
    | OUTLINE

```

```

config > autoload.php
config > autoload.php
  | config.php
  | config.php
  | constants.php
  | database.php
  | doctypes.php
  | foreign_chars.php
  | google_config.php
  | index.html
  | index.php
  | memcached.php
  | migration.php
  | mimes.php
  | pagination.php
  | profiler.php
  | routes.php
  | smileys.php
  | user_agents.php
  controllers
    | Cache.php
    | Dashboard.php
    | Data_whisp.php
    | index.html
    | Kategori.php
    | Login.php
    | Pertanyaan.php
    Site.php
    User.php
  core
    | OUTLINE

```

```

36 | $autoload['packages'] = array(APPPATH.'third_party', '/user/local/shared');
37 | |
38 | | Prototype:
39 | |
40 | | $autoload['packages'] = array();
41 | |
42 | | $autoload['packages'] = array();
43 | |
44 | | Prototype:
45 | |
46 | | Auto-load Libraries
47 | |
48 | | These are the classes located in system/libraries/ or your
49 | | application/libraries/ directory, with the addition of the
50 | | "Database" library, which is somewhat of a special case.
51 | |
52 | | Prototype:
53 | |
54 | | $autoload['libraries'] = array('database', 'email', 'session');
55 | |
56 | | You can also supply an alternative library name to be assigned
57 | | in the controller:
58 | |
59 | | $autoload['libraries'] = array('user_agent' => 'ua');
60 | |
61 | | $autoload['libraries'] = array('database', 'pagination', 'session', 'upload');
62 | |
63 | | Prototype:
64 | |
65 | | Auto-load Drivers
66 | |
67 | | These classes are located in system/libraries/ or in your
68 | | application/libraries/ directory, but are also placed inside their
69 | | own subdirectory and they extend the CI_Driver_Library class. They
70 | | offer multiple interchangeable driver options.
71 | |
72 | | Prototype:
73 | |
74 | | $autoload['drivers'] = array('cache');
75 | |

```

In 61, Col 1 (78 selected) Spaces: 4 UTF-8 LF PHP Q X

Gambar 5. 19 Langkah Kelima

## 7. Buka config>config.php, pada bagian:

`$autoload['helper'] =`

isikan menjadi :

`$autoload['helper'] = array('url','security','form','file');`

Konfigurasi pada file ini bertujuan untuk mementukan sumber daya apa yang akan diload secara otomatis.

```

File Edit Selection View Go Debug Terminal Help
autoloader.php - application - Visual Studio Code

EXPLORER config.php autoload.php
config > autoload.php
  | config.php
  | config.php
  | constants.php
  | database.php
  | doctypes.php
  | foreign_chars.php
  | google_config.php
  | index.html
  | index.php
  | memcached.php
  | migration.php
  | mimes.php
  | pagination.php
  | profiler.php
  | routes.php
  | smileys.php
  | user_agents.php
  controllers
    | Cache.php
    | Dashboard.php
    | Data_whisp.php
    | index.html
    | Kategori.php
    | Login.php
    | Pertanyaan.php
    Site.php
    User.php
  core
    | OUTLINE

```

```

config > autoload.php
config > autoload.php
  | config.php
  | config.php
  | constants.php
  | database.php
  | doctypes.php
  | foreign_chars.php
  | google_config.php
  | index.html
  | index.php
  | memcached.php
  | migration.php
  | mimes.php
  | pagination.php
  | profiler.php
  | routes.php
  | smileys.php
  | user_agents.php
  controllers
    | Cache.php
    | Dashboard.php
    | Data_whisp.php
    | index.html
    | Kategori.php
    | Login.php
    | Pertanyaan.php
    Site.php
    User.php
  core
    | OUTLINE

```

```

77 | | You can also supply an alternative property name to be assigned in
78 | | the controller:
79 | | $autoload['drivers'] = array('cache' => 'cch');
80 | |
81 | | $autoload['drivers'] = array();
82 | |
83 | | Prototype:
84 | |
85 | | Auto-load Helper Files
86 | |
87 | | Prototype:
88 | | $autoload['helper'] = array('url', 'file');
89 | |
90 | | $autoload['helper'] = array('url', 'file');
91 | |
92 | | $autoload['helper'] = array('security', 'form', 'url', 'file');
93 | |
94 | | Prototype:
95 | | Auto-load Config files
96 | |
97 | | Prototype:
98 | | $autoload['config'] = array('config1', 'config2');
99 | |
100 | | NOTE: This item is intended for use ONLY if you have created custom
101 | | config files. Otherwise, leave it blank.
102 | |
103 | | Prototype:
104 | | $autoload['config'] = array('config');
105 | |
106 | | $autoload['config'] = array('pagination');
107 | |
108 | | Prototype:
109 | | Auto-load Language files
110 | |
111 | | Prototype:
112 | | $autoload['language'] = array('lang1', 'lang2');
113 | |

```

In 100, Col 1 (81 selected) Spaces: 4 UTF-8 LF PHP Q X

Gambar 5. 20 Langkah Keenam

8. Setelah itu pergi ke config>database, pada bagian :

\$db['default'] = array(

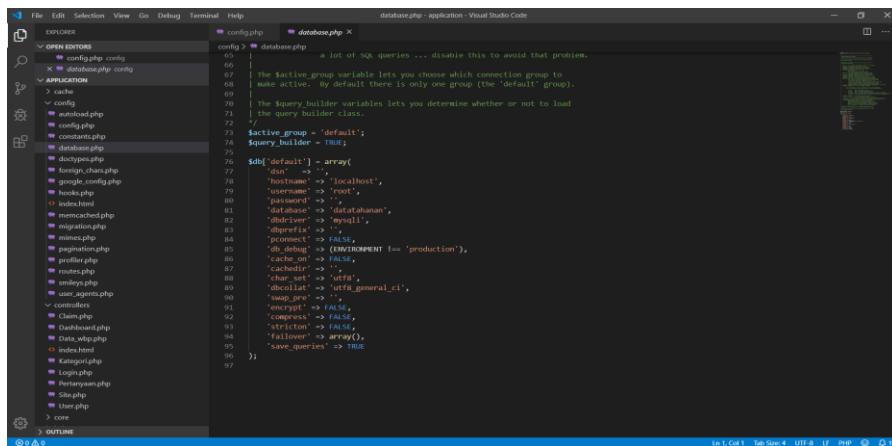
```
'dsn' => '',
'hostname' => 'localhost',
'username' => '',
'password' => '',
'database' => '',
'dbdriver' => 'mysqli',
```

Rubah menjadi :

\$db['default'] = array(

```
'dsn' => '',
'hostname' => 'localhost',
'username' => 'root',
'password' => '',
'database' => 'datatahanan',
'dbdriver' => 'mysqli',
```

Yang berfungsi untuk menghubungkan koneksi database yang sudah dibuat kedalam aplikasi web yang akan dibuat



```
database.php  application  Visual Studio Code
File Edit Selection View Go Debug Terminal Help
REPO  database.php ×
REPO  config.php ×
REPO  database.php config
REPO  database.php config
REPO  cache
REPO  config
REPO  download.php
REPO  config.php
REPO  constants.php
REPO  database.php
REPO  directory.php
REPO  download.php
REPO  footer.php
REPO  google_config.php
REPO  hooks.php
REPO  index.html
REPO  memcached.php
REPO  migration.php
REPO  minimes.php
REPO  pagination.php
REPO  profiler.php
REPO  routes.php
REPO  routes.php
REPO  user_agent.php
REPO  controllers
REPO  Claim.php
REPO  Dashboard.php
REPO  Debugging.php
REPO  index.html
REPO  Kategori.php
REPO  Logon.php
REPO  Pertanyaan.php
REPO  Site.php
REPO  User.php
> core
> OUTLINE
@ @ @

65  /* a lot of SQL queries ... disable this to avoid that problem.
66  */
67  | The active_group variable lets you choose which connection group to
68  | make active. By default there is only one group (the "default" group).
69  |
70  | The query_builder variables lets you determine whether or not to load
71  | the query builder class.
72  */
73  $active_group = 'default';
74  $query_builder = TRUE;
75
76  $db['default'] = array(
77      'dsn' => '',
78      'hostname' => 'localhost',
79      'username' => 'root',
80      'password' => '',
81      'database' => 'datatahanan',
82      'dbdriver' => 'mysqli',
83      'dbprefix' => '',
84      'pconnect' => FALSE,
85      'db_debug' => ENVIRONMENT === 'production',
86      'cache_on' => FALSE,
87      'cachedir' => '',
88      'char_set' => 'utf8',
89      'dbcollat' => 'utf8_general_ci',
90      'swap_pre' => '',
91      'encrypt' => FALSE,
92      'compress' => FALSE,
93      'stricton' => FALSE,
94      'failover' => array(),
95      'save_queries' => TRUE
96  );
97
```

Gambar 5. 21 Langkah Ketujuh

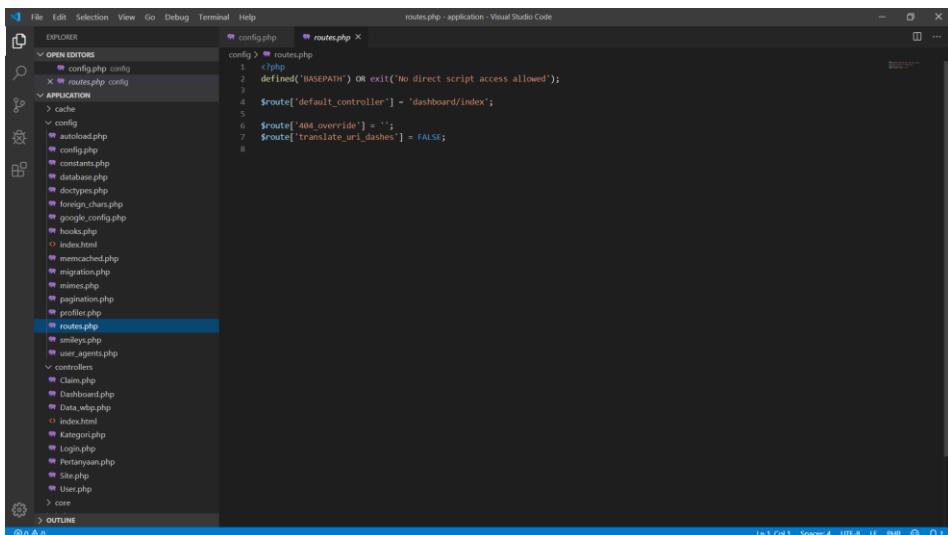
9. Buka config>routes.php, pada bagian:

```
$route['default_controller'] = 'welcome';
```

isikan menjadi :

```
$route['default_controller'] = 'dashboard/index';
```

Konfigurasi pada file ini bertujuan untuk mementukan default controller yang nantinya akan dipanggil.

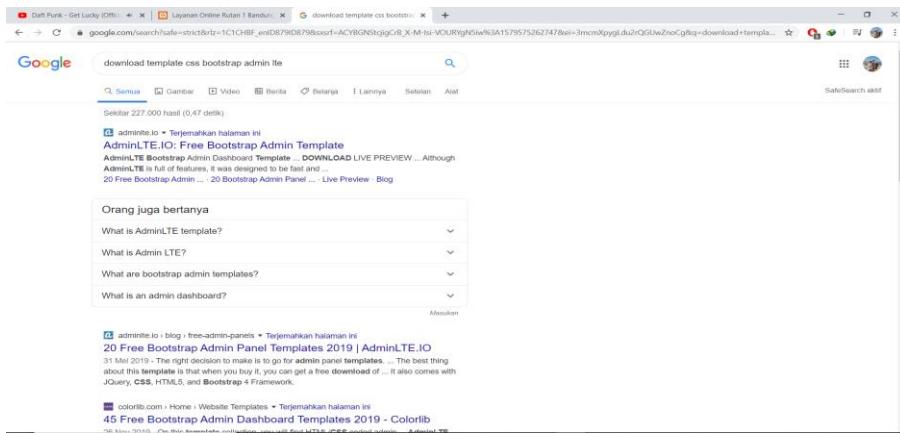


Gambar 5. 22 Langkah Kedepalan

#### 5.1.4 Konfigurasi Template CSS Bootstrap

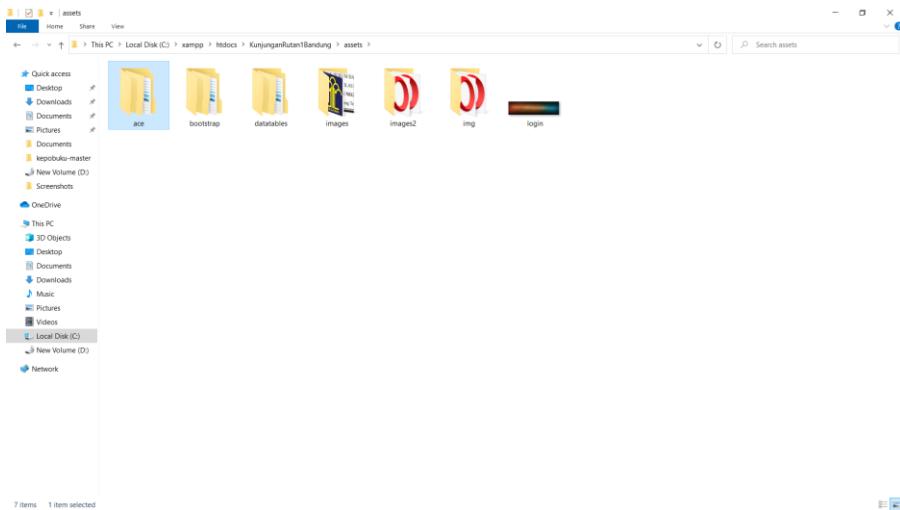
Pada setiap pembuatan aplikasi Tentunya harus memiliki tampilan yang menarik, maka dari itu diwajibkan untuk melakukan seting template pada codeigniter untuk mempermudah pemanggilan VIEW yaitu seperti:

1. Pilih template yang sesuai dengan keinginan atau sesuai dengan kebutuhan aplikasi yang dapat di unduh melalui web browser dengan mengetikan katakunci download template css bootsrap



Gambar 5. 23 Download Template

2. Jika sudah di download extract bootsrap tersebut dan masukan kedalam folder assets pada codeigniter projek.



Gambar 5. 24 Extract Bootsrap

### 5.1.5 Memasukan Template Kedalam Codingan View Admin

Untuk dapat menampilkan hasil dari CSS bootsrap harus dimasukan kedalam kodingan view pertemplate yang akan dipanggil, supaya dapat dengan mudah untuk memenggil tampilan tersebut ketika membangun tampilan view pada program yaitu dengan cara seperti berikut:

## 1. Bootstrap Top Menu Admin



Gambar 5. 25 Heading

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Rumah Tahanan</title>
    <!-- Tell the browser to be responsive to screen width -->
    <meta content="width=device-width, initial-scale=1,
maximum-scale=1, user-scalable=no" name="viewport">
    <!-- Bootstrap 3.3.6 -->
    <link rel="stylesheet" href="<?php echo
site_url('resources/css/bootstrap.min.css');?>">
    <!-- Font Awesome -->
    <link rel="stylesheet" href="<?php echo
site_url('resources/css/font-awesome.min.css');?>">
    <!-- Ionicons -->
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/ionicons/2.0.1/css/ionics.min.css">
    <!-- Datetimepicker -->
    <link rel="stylesheet" href="<?php echo
site_url('resources/css/bootstrap-datetimepicker.min.css');?>">
    <!-- Theme style -->
    <link rel="stylesheet" href="<?php echo
site_url('resources/css/AdminLTE.min.css');?>">
    <!-- AdminLTE Skins. Choose a skin from the css/skins
         folder instead of downloading all of them to reduce the
         load. -->
    <link rel="stylesheet" href="<?php echo
site_url('resources/css/_all-skins.min.css');?>">
  </head>

  <body class="hold-transition skin-blue sidebar-mini">
```

```

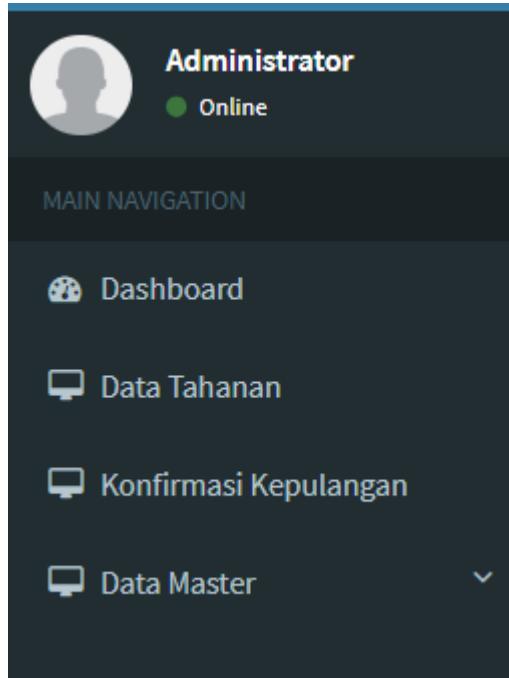
<div class="wrapper">
  <header class="main-header">
    <!-- Logo -->
    <a href="" class="logo">
      <!-- mini logo for sidebar mini 50x50 pixels -->
      <span class="logo-mini">RT</span>
      <!-- logo for regular state and mobile devices -->
      <span class="logo-lg">Rumah Tahanan</span>
    </a>
    <!-- Header Navbar: style can be found in header.less -->
    <nav class="navbar navbar-static-top">
      <!-- Sidebar toggle button-->
      <a href="#" class="sidebar-toggle" data-
        toggle="offcanvas" role="button">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </a>

      <div class="navbar-custom-menu">
        <ul class="nav navbar-nav">
          <!-- User Account: style can be found in
            dropdown.less -->
          <li class="dropdown user user-menu">
            <a href="#" class="dropdown-toggle" data-
              toggle="dropdown">
              <img src=<?php echo
                site_url('images/user.png');?>" class="user-image" alt="User
                Image">
              <span class="hidden-xs"><?= $this-
                >session->userdata('nama') ?></span>
            </a>
            <ul class="dropdown-menu">
              <li class="user-footer">
                <div class="pull-right">
                  <a href=<?= site_url('login/logout')?
                >" class="btn btn-default btn-flat">Sign out</a>
                </div>
              </li>
            </ul>
          </li>
        </ul>
      </div>
    </nav>
  </header>
  <div class="content-wrapper">
    <div class="content-header">
      <h1>Dashboard</h1>
      <small>Version 2.0.0</small>
    </div>
    <div class="content-body">
      <div class="row">
        <div class="col-md-3">
          <div class="card">
            <div class="card-block">
              <h3>Recent Activity</h3>
              <ul class="list-group">
                <li>John Doe - 2 hours ago</li>
                <li>Jane Doe - 3 hours ago</li>
                <li>Mike Doe - 4 hours ago</li>
                <li>Sarah Doe - 5 hours ago</li>
                <li>David Doe - 6 hours ago</li>
              </ul>
            </div>
          </div>
        </div>
        <div class="col-md-9">
          <div class="card">
            <div class="card-block">
              <h3>Recent Activity</h3>
              <ul class="list-group">
                <li>John Doe - 2 hours ago</li>
                <li>Jane Doe - 3 hours ago</li>
                <li>Mike Doe - 4 hours ago</li>
                <li>Sarah Doe - 5 hours ago</li>
                <li>David Doe - 6 hours ago</li>
              </ul>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```
        </ul>
    </li>
</ul>
</div>
</nav>
</header>
```

## 2. Bootstrap Sidebar Admin



Gambar 5. 26 Sidebar Admin

```
<!-- Left side column. contains the logo and sidebar -->
<aside class="main-sidebar">
    <!-- sidebar: style can be found in sidebar.less -->
    <section class="sidebar">
        <!-- Sidebar user panel -->
        <div class="user-panel">
            <div class="pull-left image">
                
    <div class="pull-left info">
        <p><?= $this->session->userdata('nama') ?></p>
        <a href="#"><i class="fa fa-circle text-
success"></i> Online</a>
    </div>
    </div>
    <!-- sidebar menu: : style can be found in sidebar.less -->
    <ul class="sidebar-menu">
        <li class="header">MAIN NAVIGATION</li>
        <li>
            <a href=<?php echo base_url();?>">
                <i class="fa fa-dashboard"></i>
            <span>Dashboard</span>
            </a>
        </li>
        <li>
            <a href=<?php echo
site_url('data_wbp/index/lost');?>">
                <i class="fa fa-desktop"></i> <span>Data
Tahanan</span>
            </a>
        </li>
        <li>
            <a href=<?php echo site_url('claim/index');?>">
                <i class="fa fa-desktop"></i> <span>Konfirmasi
Kepulangan</span>
            </a>
        </li>
        <li>
            <a href="#">
                <i class="fa fa-desktop"></i> <span>Data Master
<i style="float: right; margin-right: 10px" class="fa fa-angle-
down"></i></span>
            </a>
        </li>
    </ul>

```

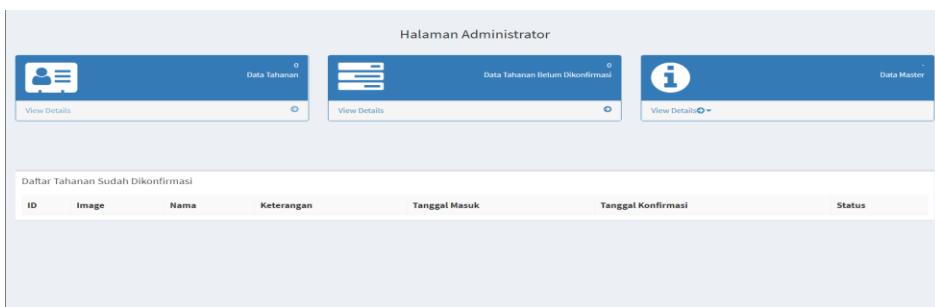
```

<ul class="treeview-menu">
    <li>
        <a href="php echo site_url('user/index');?&gt;"&gt;&lt;i class="fa fa-list-ul"&gt;&lt;/i&gt;
        Users&lt;/a&gt;
    &lt;/li&gt;
    &lt;li&gt;
        &lt;a href="<?php echo site_url('kategori/index');?&gt;"&gt;&lt;i class="fa fa-list-ul"&gt;&lt;/i&gt;
        Pasal&lt;/a&gt;
    &lt;/li&gt;
    &lt;li&gt;
        &lt;a href="<?php echo site_url('pertanyaan/index');?&gt;"&gt;&lt;i class="fa fa-list-ul"&gt;&lt;/i&gt;
        Pertanyaan&lt;/a&gt;
    &lt;/li&gt;
&lt;/ul&gt;
&lt;/li&gt;
&lt;/ul&gt;
&lt;/section&gt;
&lt;!-- /.sidebar --&gt;
&lt;/aside&gt;
</pre

```

### 3. Bootstrap Template Dashboard Admin

Berfungsi sebagai main content yang akan menampilkan data data berupa data gambar,table dll yang akan di panggil dari database



Gambar 5. 27 Halaman Utama Admin

```

<h3 align="center">Halaman Administrator</h3>
<br>
<div class="row">
    <div class="col-lg-4 col-md-6">
        <div class="panel panel-primary">
            <div class="panel-heading">
                <div class="row">
                    <div class="col-xs-3">
                        <i class="fa fa-address-card fa-5x"></i>
                    </div>
                    <div class="col-xs-9 text-right">
                        <div class="huge"><?php echo
$total;?></div>
                        <div>Data Tahanan</div>
                    </div>
                </div>
            </div>
            <a href="<?php echo
base_url();?>data_wbp/index/lost">
                <div class="panel-footer">
                    <span class="pull-left">View Details</span>
                    <span class="pull-right"><i class="fa fa-arrow-
circle-right"></i></span>
                    <div class="clearfix"></div>
                </div>
            </a>
        </div>
    </div>
    <div class="col-lg-4 col-md-6">
        <div class="panel panel-primary">
            <div class="panel-heading">
                <div class="row">
                    <div class="col-xs-3">
                        <i class="fa fa-tasks fa-5x"></i>
                    </div>
                    <div class="col-xs-9 text-right">
                        <div class="huge"><?= $claim?></div>
                        <div>Data Tahanan Belum
Dikonfirmasi</div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        </div>
    </div>
    <a href=<?= base_url();?>claim/index">
        <div class="panel-footer">
            <span class="pull-left">View Details</span>
            <span class="pull-right"><i class="fa fa-arrow-circle-right"></i></span>
            <div class="clearfix"></div>
        </div>
    </a>
</div>
<div class="col-lg-4 col-md-6">
    <div class="panel panel-primary">
        <div class="panel-heading">
            <div class="row">
                <div class="col-xs-3">
                    <i class="fa fa-info-circle fa-5x"></i>
                </div>
                <div class="col-xs-9 text-right">
                    <div class="huge">-</div>
                    <div>Data Master</div>
                </div>
            </div>
        </div>
        <a href="#">
            <div class="panel-footer">
                <span class="pull-left">View Details</span>
                <span class="pull-right">
                    <a class="dropdown-toggle-right" data-toggle="dropdown" href="#">
                        <i class="fa fa-arrow-circle-right"></i> <i class="fa fa-caret-down"></i>
                    </a>
                <ul class="dropdown-menu dropdown-alerts">
                    <li>
                        <a href=<?= base_url();?>user/index">
                            <div>
                                <i class="fa fa-users fa-fw"></i> Users
                            </div>

```

```

        </a>
    </li>
<li class="divider"></li>
<li>
    <a href=<?= base_url();?>kategori/index">
        <div>
            <i class="fa fa-ban fa-fw"></i> Pasal
        </div>
    </a>
</li>
<li class="divider"></li>
<li>
    <a href=<?= base_url();?>pertanyaan/index">
        <div>
            <i class="fa fa-question-circle fa-fw"></i> Pertanyaan
        </div>
    </a>
</li>
</ul>

```

### 5.1.6 Source Code Admin Pertanyaan

Berfungsi sebagai main content yang akan menampilkan data data berupa data table yang akan di panggil dari database, dan menambahkan data.

Pasal Listing			Add
ID	Pasal	Actions	
9	362 KUHP Tentang Pencurian	<a href="#">Edit</a>	<a href="#">Delete</a>
8	363 KUHP Tentang Pencurian	<a href="#">Edit</a>	<a href="#">Delete</a>
7	378 KUHP Tentang Penipuan	<a href="#">Edit</a>	<a href="#">Delete</a>
6	127 AVAT (1) Tentang Narkotika	<a href="#">Edit</a>	<a href="#">Delete</a>
5	372 KUHP Tentang Penggelapan	<a href="#">Edit</a>	<a href="#">Delete</a>

Gambar 5. 28 From Pertanyaan

View Pertanyaan
<div class="row"> <div class="col-md-12"> <div class="box"> <div class="box-header"> <h3 class="box-title">Pertanyaan Listing</h3> </div> </div> </div> </div>

```

<div class="box-tools">
    <a href="<?php echo site_url('pertanyaan/add'); ?>" class="btn btn-success btn-sm">Add</a>
</div>
</div>
<div class="box-body">
    <table class="table table-striped table-bordered">
        <tr>
            <th>ID</th>
            <th>Pasal</th>
            <th>Pertanyaan</th>
            <th>Actions</th>
        </tr>
        <?php foreach($pertanyaan as $p){ ?>
        <tr>
            <td><?php echo $p['id']; ?></td>
            <td><?php echo $p['nama_pasal']; ?></td>
            <td><?php echo $p['pertanyaan']; ?></td>
            <td>
                <a href="<?php echo site_url('pertanyaan/edit/'.$p['id']); ?>" class="btn btn-info btn-xs"><span class="fa fa-pencil"></span> Edit</a>
                <a href="<?php echo site_url('pertanyaan/remove/'.$p['id']); ?>" class="btn btn-danger btn-xs"><span class="fa fa-trash"></span> Delete</a>
            </td>
        </tr>
        <?php } ?>
    </table>
    <div class="pull-right">
        <?php echo $this->pagination->create_links(); ?>
    </div>
</div>
</div>
</div>

```

## Controller Pertanyaan

```
<?php

class Pertanyaan extends CI_Controller{
    function __construct()
    {
        parent::__construct();
        $this->load->model('Pertanyaan_model');
    }

    function index() // pertanyaan data wbp (admin)
    {
        $params['limit'] = RECORDS_PER_PAGE;
        $params['offset'] = ($this->input->get('per_page')) ? $this->input->get('per_page') : 0;

        $config = $this->config->item('pagination');
        $config['base_url'] = site_url('pertanyaan/index?');
        $config['total_rows'] = $this->Pertanyaan_model->get_all_pertanyaan_count();
        $this->pagination->initialize($config);

        $data['pertanyaan'] = $this->Pertanyaan_model->get_all_pertanyaan($params);

        $data['_view'] = 'pertanyaan/index';
        $this->load->view('layouts/main',$data);
    }

    function add() // menambahkan pertanyaan pada wbp yang di inputkan (admin)
    {
        $this->load->library('form_validation');

        $this->form_validation->set_rules('pertanyaan','Pertanyaan','required');
```

```

if($this->form_validation->run())
{
    $params = array(
        'pasal_id' => $this->input->post('pasal_id'),
        'pertanyaan' => $this->input->post('pertanyaan'),
    );

    $pertanyaan_id      =      $this->Pertanyaan_model-
>add_pertanyaan($params);
    redirect('pertanyaan/index');
}
else
{
    $this->load->model('Kategori_model');
    $data['all_kategori']      =      $this->Kategori_model-
>get_all_kategori();

    $data['_view'] = 'pertanyaan/add';
    $this->load->view('layouts/main',$data);
}
}

function edit($id) // edit data pertanyaan wbp (admin)
{
    $data['pertanyaan']      =      $this->Pertanyaan_model-
>get_pertanyaan($id);

    if(isset($data['pertanyaan']['id']))
    {
        $this->load->library('form_validation');

        $this->form_validation-
>set_rules('pertanyaan','Pertanyaan','required');

        if($this->form_validation->run())
        {
            $params = array(
                'pasal_id' => $this->input->post('pasal_id'),

```

```

'pertanyaan' => $this->input->post('pertanyaan'),
);

$this->Pertanyaan_model-
>update_pertanyaan($id,$params);
    redirect('pertanyaan/index');
}
else
{
    $this->load->model('Kategori_model');
    $data['all_kategori']      =      $this->Kategori_model-
>get_all_kategori();

    $data['_view'] = 'pertanyaan/edit';
    $this->load->view('layouts/main',$data);
}
else
    show_error('The pertanyaan you are trying to edit does not
exist.');
}

function remove($id) // delete pertanyaan (admin)
{
    $pertanyaan      =      $this->Pertanyaan_model-
>get_pertanyaan($id);

if(isset($pertanyaan['id']))
{
    $this->Pertanyaan_model->delete_pertanyaan($id);
    redirect('pertanyaan/index');
}
else
    show_error('The pertanyaan you are trying to delete does not
exist.');
}
}

```

## Model Pertanyaan

```
<?php
/*
 * Generated by CRUDigniter v3.2
 * www.crudigniter.com
 */

class Pertanyaan_model extends CI_Model
{
    function __construct()
    {
        parent::__construct();
    }

    /*
     * Get pertanyaan by id
     */
    function get_pertanyaan($id)
    {
        return $this->db->get_where('pertanyaan',array('id'=>$id))->row_array();
    }

    function getPertanyaan()
    {
        $this->db->join('pasal', 'pertanyaan.pasal_id = pasal.id');
        $this->db->select('pasal.nama_pasal,      pertanyaan.pasal_id,
pertanyaan.id, pertanyaan.pertanyaan');
        return $this->db->get('pertanyaan')->result_array();
    }

    /*
     * Get all pertanyaan count
     */
    function get_all_pertanyaan_count()
    {
        $this->db->from('pertanyaan');
        return $this->db->count_all_results();
    }
}
```

```

/*
 * Get all pertanyaan
 */
function get_all_pertanyaan($params = array())
{
    $this->db->order_by('id', 'desc');
    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'], $params['offset']);
    }
    $this->db->select('pertanyaan.*', 'pasal.nama_pasal');
    $this->db->from('pertanyaan');
    $this->db->join('pasal', 'pertanyaan.pasal_id = pasal.id');

    return $this->db->get()->result_array();
}

function get_all_pertanyaan_wbp($kategori)
{
    $query = $this->db->query("SELECT * FROM pertanyaan
WHERE pasal_id=" . $kategori)->result();
    // var_dump('<pre>', $query);exit;
    return $query;
}

function insert_jawaban($wbp, $user_id, $pertanyaan, $jawaban)
{
    return $this->db->query("INSERT INTO jawaban
(data_tahanan_id, user_id, pertanyaan_id, jawaban) VALUES (" .
$wbp . ", " . $user_id . ", " . $pertanyaan . ", " . $jawaban . ")");
}

function insert_jawaban_post($wbp, $pertanyaan, $jawaban)
{
    return $this->db->query("INSERT INTO jawaban_post
(data_tahanan_id, pertanyaan_id, jawaban) VALUES (" . $wbp . ",
. $pertanyaan . ", " . $jawaban . ")");
}
/*

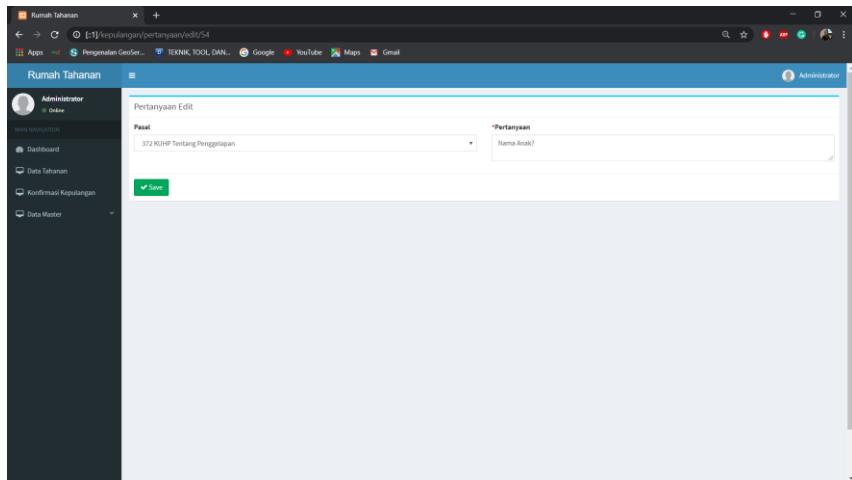
```

```
* function to add new pertanyaan
*/
function add_pertanyaan($params)
{
    $this->db->insert('pertanyaan',$params);
    return $this->db->insert_id();
}

/*
 * function to update pertanyaan
 */
function update_pertanyaan($id,$params)
{
    $this->db->where('id',$id);
    return $this->db->update('pertanyaan',$params);
}

/*
 * function to delete pertanyaan
 */
function delete_pertanyaan($id)
{
    return $this->db->delete('pertanyaan',array('id'=>$id));
}
```

### 5.1.7 Source Code Edit Pertanyaan



Gambar 5. 29 From Edit Pertanyaan

```
View Edit Pertanyaan
<div class="row">
    <div class="col-md-12">
        <div class="box box-info">
            <div class="box-header with-border">
                <h3 class="box-title">Pertanyaan Edit</h3>
            </div>
            <?php echo
form_open('pertanyaan/edit/'.$pertanyaan['id']); ?>
            <div class="box-body">
                <div class="row clearfix">
                    <div class="col-md-6">
                        <label
for="pasal_id" class="control-label">Pasal</label>
                        <div class="form-group">
                            <select
name="pasal_id" class="form-control">
                                <option value="">Pilih Pasal</option>
                            <?php
```

```

foreach($all_kategori as $kategori)
{
    $selected = ($kategori['id'] == $pertanyaan['pasal_id']) ?
    ' selected="selected"' : "";
    echo      '<option      value="'. $kategori['id']. '"'.
    '$selected.'>'. $kategori['nama_pasal']. '</option>';
}
?>
</select>
</div>
</div>
<div class="col-md-6">
    <label
for="pertanyaan" class="control-label"><span class="text-
danger">*</span>Pertanyaan</label>
    <div class="form-
group">
        <textarea
name="pertanyaan" class="form-control"
id="pertanyaan"><?php echo ($this->input->post('pertanyaan')
? $this->input->post('pertanyaan') : $pertanyaan['pertanyaan']);
?></textarea>
        <span
class="text-danger"><?php
echo
form_error('pertanyaan');?></span>
    </div>
    </div>
    </div>
    <div class="box-footer">
        <button type="submit" class="btn btn-success">
            <i      class="fa      fa-
check"></i> Save
        </button>
    </div>
    <?php echo form_close(); ?>

```

```

        </div>
    </div>
</div>
```

Model Edit Pertanyaan
<pre> function get_pertanyaan(\$id) {     return \$this-&gt;db-&gt;get_where('pertanyaan',array('id'=&gt;\$id))-&gt;row_array(); }  function getPertanyaan() {     \$this-&gt;db-&gt;join('pasal', 'pertanyaan.pasal_id = pasal.id');     \$this-&gt;db-&gt;select('pasal.nama_pasal, pertanyaan.pasal_id, pertanyaan.id, pertanyaan.pertanyaan');     return \$this-&gt;db-&gt;get('pertanyaan')-&gt;result_array(); } function update_pertanyaan(\$id,\$params) {     \$this-&gt;db-&gt;where('id',\$id);     return \$this-&gt;db-&gt;update('pertanyaan',\$params); }</pre>

Controller Edit Pertanyaan
<pre> function edit(\$id) // edit data pertanyaan wbp (admin) {     \$data['pertanyaan'] = \$this-&gt;Pertanyaan_model-&gt;get_pertanyaan(\$id);      if(isset(\$data['pertanyaan']['id']))     {         \$this-&gt;load-&gt;library('form_validation');          \$this-&gt;form_validation-&gt;set_rules('pertanyaan','Pertanyaan','required');</pre>

```

if($this->form_validation->run())
{
    $params = array(
        'pasal_id' => $this->input->post('pasal_id'),
        'pertanyaan' => $this->input->post('pertanyaan'),
    );

    $this->Pertanyaan_model-
>update_pertanyaan($id,$params);
    redirect('pertanyaan/index');
}
else
{
    $this->load->model('Kategori_model');
    $data['all_kategori']      =      $this->Kategori_model-
>get_all_kategori();

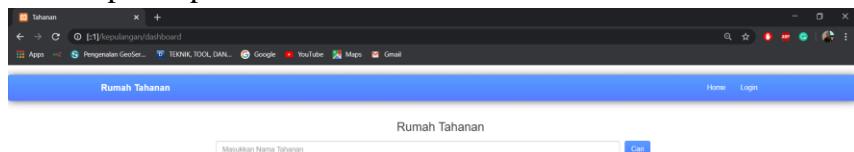
    $data['_view'] = 'pertanyaan/edit';
    $this->load->view('layouts/main',$data);
}
else
    show_error('The pertanyaan you are trying to edit does
not exist.');
}

```

### 5.1.8 Memasukan Template Kedalam codingan view Menu Utama

Untuk dapat menampilkan hasil dari CSS bootsrap harus dimasukan kedalam kodingan view pertemplate yang akan dipanggil, supaya dapat dengan mudah untuk memenggil tampilan tersebut ketika membangun tampilan view pada program yaitu dengan cara seperti berikut:

## 1. Bootstrap Template Menu Utama



Gambar 5. 30 Halaman Utama

```
<?php if (empty($this->session->userdata('role')) || $this->session->userdata('role') == 2) { ?>
<div class="container">
    <h3 align="center">Rumah Tahanan</h3>
    <div class="hilang-search">
        <div class="kiri">
            <div class="content">
                <input type="text" id="q" name="q" placeholder="Masukkan Nama Tahanan" class="form-control" required="required">
            </div>
        </div>

        <div class="kanan">
            <div class="content">
                <button id="tombol" type="submit" class="btn btn-primary">Cari</button>
            </div>
        </div>
        <script>
            $('#q').bind("enterKey",function(e){
```

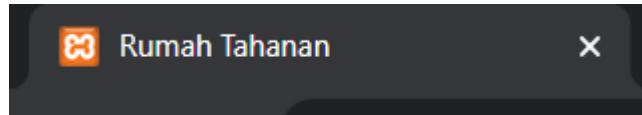
```

        window.location.href = "<?=
site_url('/dashboard/search/') ?>" + $('#q').val();
    });
    $('#q').keyup(function(e){
        if(e.keyCode == 13) {
            window.location.href = "<?=
site_url('/dashboard/search/') ?>" + $('#q').val();
        }
    });
    $('#tombol').click(function() {
        window.location.href = "<?=
site_url('/dashboard/search/') ?>" + $('#q').val();
    })
</script>
</div>
</div>
<?php foreach($data_wbp as $b){ ?>
<a href=<?php echo
site_url('site/lihatTahanan/'.$b['id_wbp']); ?>
class="wbp">
    <div class="content">
        <div style="background: url(<?=
base_url($b['image']) ?>); background-position: center
center; background-size: cover; background-repeat: no-
repeat; width: 100%; height: 220px;"></div>
        <div class="nama"><?= $b['nama_tahanan'];
?></div>
        <p align="center">Post By : Unknown </p>
    </div>
</a>
<?php } ?>

<div class="pull-right">
    <?php echo $this->pagination->create_links(); ?>
</div>

```

## 2. BOOTSRAP HEADER HOME



Gambar 5. 31 Header Halaman Utama

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible"
content="IE=edge">
        <title>Rumah Tahanan</title>
        <!-- Tell the browser to be responsive to screen width -->
        <meta content="width=device-width, initial-scale=1,
maximum-scale=1, user-scalable=no" name="viewport">
        <!-- Bootstrap 3.3.6 -->
        <link rel="stylesheet" href="<?php echo
site_url('resources/css/bootstrap.min.css');?>">
        <!-- Font Awesome -->
        <link rel="stylesheet" href="<?php echo
site_url('resources/css/font-awesome.min.css');?>">
        <!-- jQuery 2.2.3 -->
        <script src="<?php echo site_url('resources/js/jquery-
2.2.3.min.js');?>"></script>
    </head>
```

## 3. Bootsrap Top Menu Petugas Kepulangan

Home Konfirmasi Kepulangan Sudah Dikonfirmasi Logout (Johar)

Gambar 5. 32 Top Menu User Kepulangan

```
<div class="navbar-header">
    <button type="button" class="navbar-toggle
collapsed" data-toggle="collapse" data-target="#bs-
example-navbar-collapse-1" aria-expanded="false">
```

```

        <span class="sr-only">Toggle
navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
    </button>
    <a class="navbar-brand" href="/">Rumah
Tahanan</a>
</div>

        <!-- Collect the nav links, forms, and other
content for toggling -->
<div class="collapse navbar-collapse" id="bs-
example-navbar-collapse-1">
    <ul class="nav navbar-nav navbar-right">
        <li><a href=<?= base_url()
?>">Home</a></li>
            <?php if (!empty($this->session-
>userdata('nama'))) { ?>
                <?php if ($this->session-
>userdata('email') == 'johar@gmail.com') { ?>
                    <?php }else{ ?>
                    <li><a href=<?php echo
site_url('site/wbp/lost');?>">Data Tahanan</a></li>
                    <?php } ?>
                <?php if ($this->session-
>userdata('email') == 'rita5@gmail.com') { ?>
                    <?php }else{ ?>
                    <li><a href=<?php echo
site_url('site/konfirmasi');?>">Konfirmasi
Kepulangan</a></li>
                    <li><a href=<?php echo
site_url('site/sudah_konfirmasi');?>">Sudah
Dikonfirmasi</a></li>

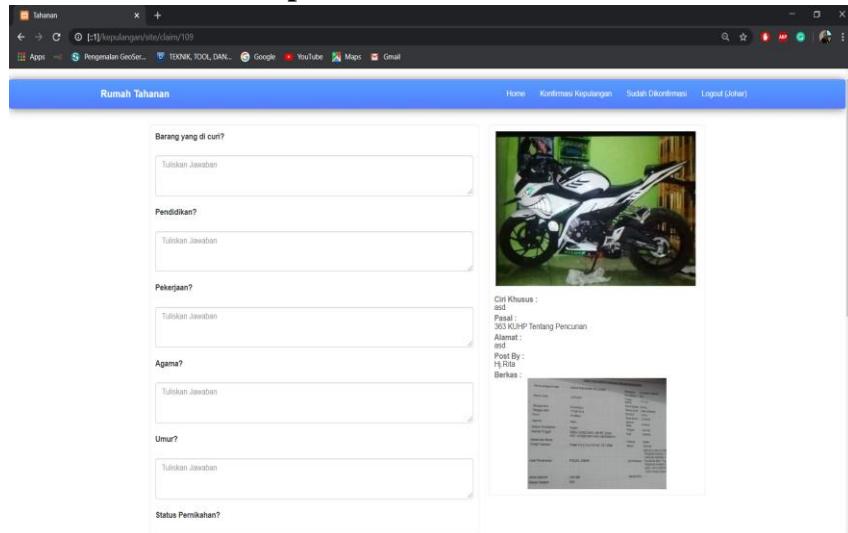
```

```

<?php } ?>
<li><a href="<?php echo
site_url('login/logout');?>">Logout (<?= $this->session-
>userdata('nama') ?>)</a></li>
<?php } else { ?>
<li><a href="<?php echo
site_url('login');?>">Login</a></li>
<?php } ?>

```

### 5.1.9 Sorce Code From Input Untuk Verifikasi



Gambar 5. 33 From Input Verifikasi

View Verifikasi
<pre> &lt;div class="form"&gt;   &lt;div class="kiri"&gt;     &lt;div class="content"&gt;       &lt;form action="&lt;?= site_url('site/sendClaim/' . \$id) ?&gt;" method="POST" enctype="multipart/form- data"&gt;         &lt;?php foreach (\$pertanyaan as \$key) { ?&gt;           &lt;label&gt;&lt;?= \$key-&gt;pertanyaan ?&gt;&lt;/label&gt; </pre>

```

<br>

<input type="hidden" name="pertanyaan[]"
class="form-control" value="<?= $key->id ?>">

<br>

<textarea name="jawaban[]" rows="3"
class="form-control" placeholder="Tuliskan Jawaban"
required></textarea>

<br>
<?php } ?>

<br>
<label>Berkas</label>
<br><br>
<input type="file" name="berkas"
class="form-control">
<br><br>
<label>Keterangan :</label>
<br>
<label>Berkas 1</label><br>
<input type="checkbox"
name="keterangan[]" value="Berkas Tahanan"> Berkas
Tahanan <br>
<label>Berkas 2</label><br>
<input type="checkbox"
name="keterangan[]" value="Berkas Penahanan">
Berkas Penahanan <br>
<label>Berkas 3</label><br>
<input type="checkbox"
name="keterangan[]" value="Berkas Asal Tahanan">
Berkas Asal Tahanan
<br>

```

```

<input type="hidden"
name="data_tahanan_id" class="form-control"
value=<?= $id ?>>
    <input type="hidden" name="input_oleh"
class="form-control" value=<?= $wbp['jenis'] == 'lost'
? $this->session->userdata('id') : $wbp['post_by'] ?>>
        <input type="hidden"
name="konfirmasi_oleh" class="form-control"
value=<?= $wbp['jenis'] == 'lost' ? $wbp['post_by'] :
$this->session->userdata('id') ?>>
    <br>
    <div class="form-group">
        <button type="submit" class="btn btn-primary submit">Save</button>
    </div>
    </form>
    </div>
</div>
<div class="kanan">
    <div class="content">
        <img src=<?= base_url($wbp['image']) ?>" alt=<?= $wbp['nama_tahanan'] ?>" style="width: 100%;">
            <p>Ciri Khusus :</p>
            <div class="value"><?= $wbp['ciri_khusus'] ?></div>
            <p>Pasal :</p>
            <div class="value"><?= $wbp['nama_pasal'] ?></div>
            <p>Alamat :</p>
            <div class="value"><?= $wbp['alamat_tahanan'] ?></div>
            <p>Post By :</p>
            <div class="value"><?= $wbp['nama'] ?></div>
            <p>Berkas :</p>

```

```

<center></center>
<!-- <div class="value"> -->
<!--  -->
</div>
</div>
</div>

```

### Model Verifikasi

```

.<?php
/*
 * Generated by CRUDigniter v3.2
 * www.crudigniter.com
 */

class Claim_model extends CI_Model
{
    function __construct()
    {
        parent::__construct();
        $this->load->library('session');
    }

    /*
     * Get claim by id
     */
    function get_claim($id)
    {
        $this->db->select('claim.*', 'a.nama as pemilik,
        b.nama as penemu', 'a.no_telp as telp_pemilik, b.no_telp
        as telp_penemu, data_tahanan.image,
        data_tahanan.berkas as berkas_tahanan,
        data_tahanan.nama_tahanan, data_tahanan.jenis AS
    
```

```

jenis, data_tahanan.ciri_khusus,
data_tahanan.created_at AS tanggal_upload,
definisi_pasal.isi, pasal.nama_pasal');
$this->db->from('claim');
$this->db->join('user a', 'claim.input_oleh = a.id');
$this->db->join('user b', 'claim.konfirmasi_oleh =
b.id');
$this->db->join('data_tahanan',
'claim.data_tahanan_id = data_tahanan.id');
$this->db->join('pasal', 'data_tahanan.pasal_id =
pasal.id');
$this->db->join('definisi_pasal',
'data_tahanan.pasal_id = definisi_pasal.id_pasal');
$this->db->where('claim.id', $id);

return $this->db->get()->row_array();

}

// function get_jawaban_claim($id)
// {
//   $this->db->select('*');
//   $this->db->from('jawaban');
//   $this->db->join('pertanyaan',
'jawaban.pertanyaan_id = pertanyaan.id');
//   $this->db->join('user', 'jawaban.user_id =
user.id');
//   $this->db->where('data_tahanan_id', $id);
//   $this->db->group_by('pertanyaan.id');

//   return $this->db->get()->result();
// }

function get_jawaban_claim($id)
{

```

```

        $this->db->select('*', jawaban.jawaban as
jawaban_johar, a.jawaban as jawaban_rita');
        $this->db->from('jawaban');
        $this->db->join('pertanyaan',
'jawaban.pertanyaan_id = pertanyaan.id');
        $this->db->join('jawaban_post a',
'jawaban.data_tahanan_id = a.data_tahanan_id and
jawaban.pertanyaan_id = a.pertanyaan_id');
        $this->db->join('user', 'jawaban.user_id = user.id');
        $this->db->where('jawaban.data_tahanan_id', $id);
        $this->db->group_by('pertanyaan.id');

        return $this->db->get()->result();
    }

    function get_jawaban_claim2($id)
    {
        $this->db->select('*', jawaban.jawaban as
jawaban_johar, jawaban_post.jawaban as
jawaban_rita');
        $this->db->from('jawaban');
        $this->db->join('pertanyaan',
'jawaban.pertanyaan_id = pertanyaan.id');
        $this->db->join('jawaban_post',
'jawaban.data_tahanan_id =
jawaban_post.data_tahanan_id');
        $this->db->join('user', 'jawaban.user_id = user.id');
        $this->db->where('jawaban.data_tahanan_id', $id);
        $this->db->group_by('pertanyaan.id');

        return $this->db->get()->result();
    }

    function get_jawaban_post($id)
    {

```

```

        $this->db->select('*');
        $this->db->from('jawaban_post');
        $this->db->join('pertanyaan',
'jawaban_post.pertanyaan_id = pertanyaan.id');
        $this->db->where('data_tahanan_id', $id);
        $this->db->group_by('pertanyaan.id');

        return $this->db->get()->result();
    }

/*
 * Get all claim count
 */
function get_all_claim_count()
{
    $this->db->select('claim.*', a.nama as pemilik,
b.nama as penemu, a.no_telp as telp_pemilik, b.no_telp
as telp_penemu, data_tahanan.image,
data_tahanan.nama_tahanan, data_tahanan.jenis AS
jenis, data_tahanan.ciri_khusus,
data_tahanan.created_at AS tanggal_upload');
    $this->db->from('claim');
    $this->db->join('user a', 'claim.input_oleh = a.id');
    $this->db->join('user b', 'claim.konfirmasi_oleh =
b.id');
    $this->db->join('data_tahanan',
'claim.data_tahanan_id = data_tahanan.id');
    if ($this->session->userdata('role') == 2) {
        $this->db->where('input_oleh', $this->session-
>userdata('id'));
        $this->db->or_where('konfirmasi_oleh', $this-
>session->userdata('id'));
    }
    return $this->db->count_all_results();
}

```

```

function get_all_claim_count2()
{
    $this->db->select('claim.* , a.nama as pemilik,
b.nama as penemu, a.no_telp as telp_pemilik, b.no_telp
as telp_penemu, data_tahanan.image,
data_tahanan.nama_tahanan, data_tahanan.jenis AS
jenis, data_tahanan.ciri_khusus,
data_tahanan.created_at AS tanggal_upload');

    $this->db->from('claim');
    $this->db->where('status', 0);
    $this->db->join('user a', 'claim.input_oleh = a.id');
    $this->db->join('user b', 'claim.konfirmasi_oleh =
b.id');
    $this->db->join('data_tahanan',
'claim.data_tahanan_id = data_tahanan.id');

    return $this->db->count_all_results();
}

/*
 * Get all claim
*/
function get_all_claim($params = array())
{
    $this->db->order_by('id', 'desc');
    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'],
$params['offset']);
    }
    $this->db->select('claim.* , a.nama as pemilik,
b.nama as penemu, a.no_telp as telp_pemilik, b.no_telp
as telp_penemu, data_tahanan.image,
data_tahanan.nama_tahanan, data_tahanan.jenis AS
jenis, data_tahanan.created_at AS tanggal_upload');
}

```

```

        $this->db->from('claim');
        $this->db->join('user a', 'claim.input_oleh = a.id');
        $this->db->join('user b', 'claim.konfirmasi_oleh =
b.id');
        $this->db->join('data_tahanan',
'claim.data_tahanan_id = data_tahanan.id');
        $this->db->where('claim.status', 0);
        if ($this->session->userdata('role') == 2) {
            $this->db->where('input_oleh', $this->session-
>userdata('id'));
            $this->db->or_where('konfirmasi_oleh', $this-
>session->userdata('id'));
        }
        return $this->db->get()->result_array();
    }

    function get_all_claim_konfirmasi($params =
array())
    {
        $this->db->order_by('id', 'desc');
        if(isset($params) && !empty($params))
        {
            $this->db->limit($params['limit'],
$params['offset']);
        }
        $this->db->select('claim.*', a.nama as pemilik,
b.nama as penemu, a.no_telp as telp_pemilik, b.no_telp
as telp_penemu, data_tahanan.image,
data_tahanan.nama_tahanan, data_tahanan.jenis AS
jenis, data_tahanan.created_at AS tanggal_upload');
        $this->db->from('claim');
        $this->db->join('user a', 'claim.input_oleh = a.id');
        $this->db->join('user b', 'claim.konfirmasi_oleh =
b.id');
    }

```

```

        $this->db->join('data_tahanan',
'claim.data_tahanan_id = data_tahanan.id');
        $this->db->where('claim.status', 1);
        if ($this->session->userdata('role') == 2) {
            $this->db->where('input_oleh', $this->session-
>userdata('id'));
            $this->db->or_where('konfirmasi_oleh', $this-
>session->userdata('id'));
        }
        return $this->db->get()->result_array();
    }

    function get_all_claim_konfirmasi_print()
    {
        $this->db->select('claim.*', a.nama as pemilik,
b.nama as penemu, a.no_telp as telp_pemilik, b.no_telp
as telp_penemu, data_tahanan.image,
data_tahanan.nama_tahanan, data_tahanan.jenis AS
jenis, data_tahanan.created_at AS tanggal_upload');
        $this->db->from('claim');
        $this->db->order_by('id', 'desc');
        $this->db->join('user a', 'claim.input_oleh = a.id');
        $this->db->join('user b', 'claim.konfirmasi_oleh =
b.id');
        $this->db->join('data_tahanan',
'claim.data_tahanan_id = data_tahanan.id');
        $this->db->where('claim.status', 1);
        if ($this->session->userdata('role') == 2) {
            $this->db->where('input_oleh', $this->session-
>userdata('id'));
            $this->db->or_where('konfirmasi_oleh', $this-
>session->userdata('id'));
        }
        return $this->db->get()->result_array();
    }

```

```

function add_claim($params)
{
    $this->db->insert('claim',$params);
    return $this->db->insert_id();
}

function update_claim($id,$params)
{
    $this->db->where('id',$id);
    return $this->db->update('claim',$params);
}

function delete_claim($id)
{
    return $this->db->delete('claim',array('id'=>$id));
}

```

### Controler Verifikasi

```

function sendClaim($id) // mengirimkan data wbp yang
di claim kepada si user yang melakukan upload
{
    $data['wbp'] = $this->Data_tahanan_model-
>get_data_wbp($id);
    $data['pertanyaan'] = $this->Pertanyaan_model-
>get_all_pertanyaan_wbp($data['wbp']['pasal_id']);

    $this->load->library('form_validation');

    $this->form_validation-
>set_rules('jawaban[]','Jawaban','required');

    if($this->form_validation->run()) {

```

```

        foreach ($this->input->post('jawaban') as $key
=> $value) {
            $this->Pertanyaan_model-
>insert_jawaban($data['wbp']['id_wbp'],$this->session-
>userdata('id'), $this->input->post('pertanyaan')[[$key],
$value);
        }

        $status = array(
            'status_claim' => 1,
        );
        $this->Data_tahanan_model-
>update_data_wbp($id,$status);

        // print_r($this->input-
>post('keterangan')) die;

        $config['upload_path']      =
'./uploads/berkas';
        $config['allowed_types']    = 'gif|jpg|png';
        $config['max_size']         = 100;
        $config['max_width']        = 1024;
        $config['max_height']       = 768;

        $this->load->library('upload', $config);

        $this->upload->initialize($config);

        if ($this->upload->do_upload('berkas')){
            $keterangan = $this->input-
>post('keterangan');
            $result = $this->upload->data();
            $params = array(
                'data_tahanan_id' => $this->input-
>post('data_tahanan_id'),

```

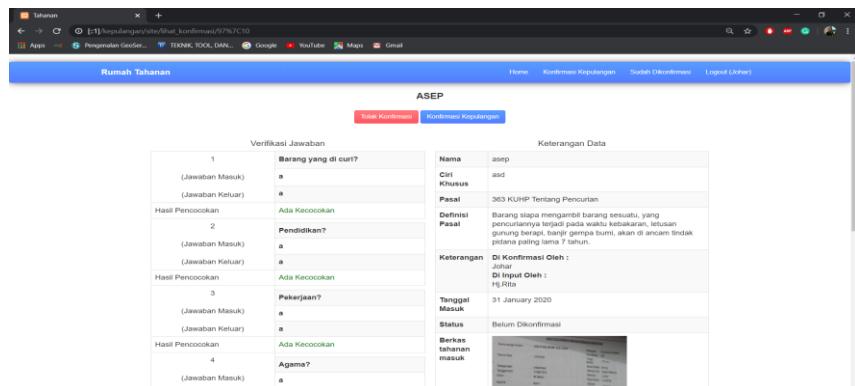
```

'input_oleh' => $this->input-
>post('input_oleh'),
'konfirmasi_oleh' => $this->input-
>post('konfirmasi_oleh'),
'status' => 0,
'created_at' => date('Y-m-d H:i:s'),
'berkas' => $result['file_name'],
'keterangan' => implode(", ", $keterangan)
);
// print_r($params);die;

$claim_id = $this->Claim_model-
>add_claim($params);
redirect(site_url('dashboard'));
} else {
$error = $this->upload->display_errors();
print_r($error);
}
} else {
redirect(site_url('site/claim'));
}
}

```

### 5.1.10 Souce Code Lihat Konfirmasi



Gambar 5. 34 From Lihat Konfirmasi

### View From Lihat Konfirmasi

```
<?php
    // Metode Knuth Morris Pratt
    function KnuthMorrisPratt($text, $pattern) { //fungsi
        ini butuh 2 parameter yaitu text dan pattern
        $text = strtolower($text); //ini membuat parameter
        yang isinya dari fungsi fungsi sebelumnya tapi dibuat
        lowercase semua
        $pattern = strtolower($pattern); //ini membuat
        parameter yang isinya dari fungsi fungsi sebelumnya
        tapi dibuat lowercase semua

        $patlen = strlen($pattern); //fungsi ini untuk
        mengembalikan panjang string, string yang di maksud
        adalah variabel $pattern
        $textlen = strlen($text); //fungsi ini untuk
        mengembalikan panjang string, string yang di maksud
        adalah variabel $text

        $table = makeCharTable($pattern);

        for ($i=$patlen-1; $i < $textlen;) { //fungsi ini
            untuk menghitung dimana $i hasilnya $patlen dikurang
            1, $i selalu lebih kecil dari variabel $textlen
            $t = $i;
            for ($j=$patlen-1; $pattern[$j]==$text[$i]; $j--
                ,$i--) {
                if($j == 0) return $i;
            }
            $i = $t;
            if(array_key_exists($text[$i], $table))
                $i = $i + max($table[$text[$i]], 1);
            else
                $i += $patlen;
        }
    }
}
```

```

        return -1;
    }

    function makeCharTable($string) { //fungsi ini
        membutuhkan parameter $string
        $len = strlen($string); //ini untuk membuat variabel
        $len dimana akan mengembalikan panjang kata dari
        $string
        $table = array(); //membuat variabel berisi array
        for ($i=0; $i < $len; $i++) { //perhitungan, jadi $i
            adalah 0, dan $i selalu lebih kecil dari $len
            $table[$string[$i]] = $len - $i - 1; //setelah
            perhitungan dijalankan, akan membuat lagi variabel
            $tabel yang array tadi di tampung oleh $string isinya
            $len-$i-1
        }
        return $table; //mengembalikan hasil array tadi
        berapa
    }
?>
<div class="detail">
    <div class="nama"><?= $claim['nama_tahanan'];
?></div>
    <center>
        <?php if ($user == $claim['input_oleh'] &&
$claim['status'] == 0) { ?>
            <a href="<?=
site_url('site/tolak_confirm/'.$claim['id']) ?>" class="btn
btn-danger">Tolak Konfirmasi</a>

            <a href="<?=
site_url('site/confirm_wbp/'.$claim['id']) ?>" class="btn
btn-primary">Konfirmasi Kepulangan</a>

        <?php } ?>
        <?php if ($claim['status'] == 1) { ?>

```

```

<div style="color: green; font-size:
20px;">Dikonfirmasi</div>
<?php } ?>
</center>
<br>
<?php // var_dump($claim);exit; ?>
<div class="gambar">
    <h4 align="center">Verifikasi Jawaban</h4>
    <table width="100%" class="table table-striped
table-bordered">
        <?php $no=1; foreach ($jawaban as $key =>
$value) { ?>
        <tr>

        </tr>
        <tr>
            <td align="center" rowspan="4"><?=

$no++ ?><br><br>(Jawaban Masuk)<br><br>(Jawaban
Keluar)</td>

        </tr>
        <tr>
            <td><b><?= $value->pertanyaan ?></b></td>
        </tr>
        <tr>
            <td><b><?=$value->jawaban_rita?></b></td>

        </tr>
        <tr>
            <td><b><?= $value->jawaban_johar ?> </b></td>
        </tr>

        <td>
            Hasil Pencocokan
        </td>

```

```

<td><?= KnuthMorrisPratt($value->jawaban_johar, $value->jawaban_rita) >= 0 ? '<span style="color: green">Ada Kecocokan</span>' : '<span style="color: red">Tidak Ada Kecocokan</span>'></td>
<?php } ?>
</table>
<!-- <img src=<?php // base_url($claim['image']) ?>" alt=<?php // $claim['nama_tahanan'] ?>" style="width: 100%;"> -->
</div>
<div class="gambar">
<h4 align="center">Keterangan Data</h4>
<table width="100%" class="table table-striped table-bordered">
<tr>
<th>Nama</th>
<td><?= $claim['nama_tahanan'] ?></td>
</tr>
<tr>
<th>Ciri Khusus</th>
<td><?= $claim['ciri_khusus'] ?></td>
</tr>
<tr>
<th>Pasal</th>
<td><?= $claim['nama_pasal'] ?></td>
</tr>
<tr>
<th>Definisi Pasal</th>
<td><?= $claim['isi'] ?></td>
</tr>
<tr>
<th>Keterangan</th>
<td>
```

```

<b><?= $claim['jenis'] == 'lost' ? 'Di
Konfirmasi Oleh :' : 'Di Input Oleh :' ?></b>
<br>
<?= $claim['jenis'] == 'lost' ?
$claim['pemilik'] : $claim['penemu'] ?>
<?php if ($claim['status'] == 1) { ?>
<br>

<?php } ?>
<br>
<b><?= $claim['jenis'] == 'lost' ? 'Di Input
Oleh :' : 'Di Konfirmasi Oleh :' ?></b>
<br>
<?= $claim['jenis'] == 'lost' ?
$claim['penemu'] : $claim['pemilik'] ?>
<?php if ($claim['status'] == 1) { ?>
<br>

<?php } ?>
</td>
</tr>
<tr>
<th>Tanggal Masuk</th>
<td><?=
date_format(date_create($claim['tanggal_upload']), 'd F
Y') ?></td>
</tr>
<tr>
<th>Status</th>
<td><?= $claim['status'] == 1 ? 'Dikonfirmasi'
: 'Belum Dikonfirmasi' ?></td>
</tr>
<?php
if ($claim['status'] == 1) {
?>

```

```

<tr>
<th>Tanggal Konfirmasi</th>
<td><?= date_format(date_create($claim['tanggal_konfirmasi']), 'd F Y') ?></td>
</tr>
<?php } else{ ?>

<?php } ?>
<tr>
<th>Berkas tahanan masuk</th>
<td><img style="width: 270px; height: 300px;" src=<?= base_url(); ?><?=$claim['berkas_tahanan']; ?>></td>
</tr>
<tr>
<th>Berkas tahanan keluar</th>
<td><img style="width: 270px; height: 300px;" src=<?= base_url(); ?>uploads/berkas/<?=$claim['berkas']; ?>></td>
</tr>

<tr>
<th>Keterangan Berkas</th>
<td><?= $claim['keterangan']?></td>
</tr>
</table>

</div>
</div>

```

### Controler Lihat Konfirmasi

```
function lihat_konfirmasi($param) // melihat hasil dari
konfirmasi data wbp
{
    $hasil = explode('%7C', $param);
    $id = $hasil[0];
    $user = $hasil[1];

    $claim = $this->Claim_model->get_claim($id);

    $jawaban = $this->Claim_model-
>get_jawaban_claim($claim['data_tahanan_id']);
    $jawaban_post = $this->Claim_model-
>get_jawaban_post($claim['data_tahanan_id']);

    if(isset($claim['id']))
    {
        $data['_view'] = 'site/lihat_konfirmasi';
        $data['claim'] = $claim;
        // $data['kepulangan'] = $truee;
        $data['jawaban'] = $jawaban;
        $data['jawaban_post'] = $jawaban_post;
        $data['user'] = $user;
        $data['id'] = $id;

        $this->load->view('layouts/main_user',$data);
    }
    else
        show_error('The claim you are trying to delete
does not exist.');
}
```

### Model Lihat Konfirmasi

```
function get_claim($id)
{
    $this->db->select('claim.* , a.nama as pemilik,
b.nama as penemu, a.no_telp as telp_pemilik, b.no_telp
as telp_penemu, data_tahanan.image,
data_tahanan.berkas as berkas_tahanan,
data_tahanan.nama_tahanan, data_tahanan.jenis AS
jenis, data_tahanan.ciri_khusus,
data_tahanan.created_at AS tanggal_upload,
definisi_pasal.isi, pasal.nama_pasal');

    $this->db->from('claim');
    $this->db->join('user a', 'claim.input_oleh = a.id');
    $this->db->join('user b', 'claim.konfirmasi_oleh =
b.id');
    $this->db->join('data_tahanan',
'claim.data_tahanan_id = data_tahanan.id');
    $this->db->join('pasal', 'data_tahanan.pasal_id =
pasal.id');
    $this->db->join('definisi_pasal',
'data_tahanan.pasal_id = definisi_pasal.id_pasal');

    $this->db->where('claim.id', $id);

    return $this->db->get()->row_array();

}

function get_jawaban_claim($id)
{
    $this->db->select('* , jawaban.jawaban as
jawaban_johar, a.jawaban as jawaban_rita');

    $this->db->from('jawaban');
    $this->db->join('pertanyaan',
'jawaban.pertanyaan_id = pertanyaan.id');
```

```

        $this->db->join('jawaban_post a',
'jawaban.data_tahanan_id = a.data_tahanan_id and
jawaban.pertanyaan_id = a.pertanyaan_id');
        $this->db->join('user', 'jawaban.user_id = user.id');
        $this->db->where('jawaban.data_tahanan_id', $id);
        $this->db->group_by('pertanyaan.id');

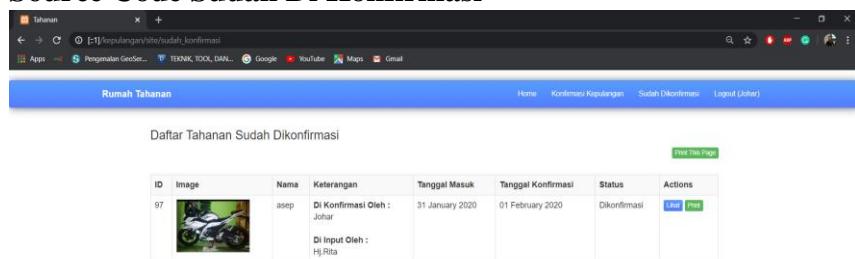
        return $this->db->get()->result();
    }

function get_jawaban_post($id)
{
    $this->db->select('*');
    $this->db->from('jawaban_post');
    $this->db->join('pertanyaan',
'jawaban_post.pertanyaan_id = pertanyaan.id');
    $this->db->where('data_tahanan_id', $id);
    $this->db->group_by('pertanyaan.id');

    return $this->db->get()->result();
}

```

### 5.1.11 Source Code Sudah Di Konfirmasi



The screenshot shows a web application titled "Rumah Tahanan". The main page displays a table with the heading "Daftar Tahanan Sudah Dikonfirmasi". The table has columns: ID, Image, Nama, Keterangan, Tanggal Masuk, Tanggal Konfirmasi, Status, and Actions. One row is visible, showing an image of a motorcycle, the name "asep", and the following details in the "Keterangan" column:

ID	Image	Nama	Keterangan	Tanggal Masuk	Tanggal Konfirmasi	Status	Actions
97		asep	Di Konfirmasi Oleh : Johar  Di Input Oleh : Hj.Rita	31 January 2020	01 February 2020	Dikonfirmasi	<a href="#">Edit</a> <a href="#">Free</a>

Gambar 5. 35 From Sudah Di Konfirmasi

## View Konfirmasi

```
<div class="row">
<div class="col-md-12">
<div class="box">
<div class="box-header">
<h3 class="box-title">Daftar Tahanan Sudah
Dikonfirmasi</h3>
<p align="right">
<a href=" site_url('site/print_sudah_konfirmasi') ?&gt;" class="btn btn-success btn-xs"&gt;Print This Page&lt;/a&gt;
&lt;/p&gt;
&lt;br&gt;
&lt;div class="box-tools"&gt;
&lt;!-- &lt;a href="<?php // echo site_url('claim/add'); ?&gt;" class="btn btn-success btn-sm"&gt;Add&lt;/a&gt; --&gt;
&lt;/div&gt;
&lt;/div&gt;
&lt;div class="box-body"&gt;
&lt;table class="table table-striped table-bordered"&gt;
&lt;tr&gt;
&lt;th&gt;ID&lt;/th&gt;
&lt;th&gt;Image&lt;/th&gt;
&lt;th&gt;Nama&lt;/th&gt;
&lt;th&gt;Keterangan&lt;/th&gt;
&lt;th&gt;Tanggal Masuk&lt;/th&gt;
&lt;th&gt;Tanggal Konfirmasi&lt;/th&gt;
&lt;th&gt;Status&lt;/th&gt;
&lt;th&gt;Actions&lt;/th&gt;
&lt;/tr&gt;
&lt;?php foreach($claim as $c){ ?&gt;
&lt;tr&gt;
&lt;td&gt;&lt;?php echo $c['id']; ?&gt;&lt;/td&gt;
&lt;td&gt;&lt;img src="<? base_url($c['image']); ?&gt;" width="150px"&gt;&lt;/td&gt;
&lt;td&gt;&lt;?php echo $c['nama_tahanan']; ?&gt;&lt;/td&gt;
&lt;td&gt;</pre
```

```

<b><?= $c['jenis'] == 'lost' ? 'Di Konfirmasi
Oleh : ' : 'Di Input Oleh : ?></b>
<br>
<?php echo $c['jenis'] == 'lost' ? $c['pemilik'] :
$c['penemu']; ?>
<br>

<br>
<b><?= $c['jenis'] == 'lost' ? 'Di Input Oleh : '
'Di Konfirmasi Oleh : ' ?></b>
<br>
<?php echo $c['jenis'] == 'lost' ? $c['penemu'] :
$c['pemilik']; ?>
<br>

</td>
<td><?=
date_format(date_create($c['tanggal_upload']), 'd F Y') ?></td>
<?php if ($c['status'] == 1) { ?>
<td><?=
date_format(date_create($c['tanggal_konfirmasi']), 'd F Y')
?></td>
<?php } else{ ?>
    <td>Belum Dikonfirmasi</td>
<?php } ?>
<td><?php echo
$c['status'] == 1 ? 'Dikonfirmasi' : 'Belum Dikonfirmasi'
?></td>
<td><a href=<?=
site_url('site/lihat_konfirmasi/'.$c['id'].'.'.$user) ?>" class="btn
btn-primary btn-xs">Lihat</a>
    <a href=<?=
site_url('site/lihat_konfirmasi2/'.$c['id'].'.'.$user) ?>" class="btn
btn-success btn-xs">Print</a></td>
</tr>
<?php } ?>
</table>
<div class="pull-right">
    <?php echo $this->pagination->create_links(); ?>
</div>

```

```
</div>
</div>
</div>
</div>
```

### Model Sudah Verifikasi

```
function get_claim($id)
{
    $this->db->select('claim.*', a.nama as pemilik,
    b.nama as penemu, a.no_telp as telp_pemilik, b.no_telp
    as telp_penemu, data_tahanan.image,
    data_tahanan.berkas as berkas_tahanan,
    data_tahanan.nama_tahanan, data_tahanan.jenis AS
    jenis, data_tahanan.ciri_khusus,
    data_tahanan.created_at AS tanggal_upload,
    definisi_pasal.isi, pasal.nama_pasal');

    $this->db->from('claim');
    $this->db->join('user a', 'claim.input_oleh = a.id');
    $this->db->join('user b', 'claim.konfirmasi_oleh =
    b.id');
    $this->db->join('data_tahanan',
    'claim.data_tahanan_id = data_tahanan.id');
    $this->db->join('pasal', 'data_tahanan.pasal_id =
    pasal.id');
    $this->db->join('definisi_pasal',
    'data_tahanan.pasal_id = definisi_pasal.id_pasal');
    $this->db->where('claim.id', $id);

    return $this->db->get()->row_array();

}

function get_jawaban_claim($id)
{
```

```

        $this->db->select('*', jawaban.jawaban as
        jawaban_johar, a.jawaban as jawaban_rita');
        $this->db->from('jawaban');
        $this->db->join('pertanyaan',
        'jawaban.pertanyaan_id = pertanyaan.id');
        $this->db->join('jawaban_post a',
        'jawaban.data_tahanan_id = a.data_tahanan_id and
        jawaban.pertanyaan_id = a.pertanyaan_id');
        $this->db->join('user', 'jawaban.user_id = user.id');
        $this->db->where('jawaban.data_tahanan_id', $id);
        $this->db->group_by('pertanyaan.id');

        return $this->db->get()->result();
    }

function get_jawaban_post($id)
{
    $this->db->select('*');
    $this->db->from('jawaban_post');
    $this->db->join('pertanyaan',
    'jawaban_post.pertanyaan_id = pertanyaan.id');
    $this->db->where('data_tahanan_id', $id);
    $this->db->group_by('pertanyaan.id');

    return $this->db->get()->result();
}

```

### Controler Konfirmasi

```

function sudah_konfirmasi() // memberikan konfirmasi data
wbp
{
    $params['limit'] = 9;
    $params['offset'] = ($this->input->get('per_page')) ? $this-
>input->get('per_page') : 0;

    $config = $this->config->item('pagination');
    $config['base_url'] = site_url('site/konfirmasi?');

```

```

$config['total_rows'] = $this->Claim_model-
>get_all_claim_count();
$this->pagination->initialize($config);

$data['claim'] = $this->Claim_model-
>get_all_claim_konfirmasi($params);

$data['_view'] = 'site/konfirmasi';
$data['user'] = $this->session->userdata('id');
$this->load->view('layouts/main_user',$data);
}

```

### View Print Sudah Konfirmasi

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible"
content="IE=edge">
<title>Rumah Tahanan</title>
<!-- Tell the browser to be responsive to screen width -->
<meta content="width=device-width, initial-scale=1,
maximum-scale=1, user-scalable=no" name="viewport">
<!-- Bootstrap 3.3.6 -->
<link rel="stylesheet" href="<?php echo
site_url('resources/css/bootstrap.min.css');?>">
<!-- Font Awesome -->
<link rel="stylesheet" href="<?php echo
site_url('resources/css/font-awesome.min.css');?>">
<!-- jQuery 2.2.3 -->
<script src="<?php echo site_url('resources/js/jquery-
2.2.3.min.js');?>"></script>
</head>

<body onload="javascript:window.print()">
<hr><width="100" height="75"></hr>
<table width="100%">
<td width="30" align="center"></td>

```

```

<td width="70" align="center"><h2><center><font size="4"
face="arial">
    KEMENTERIAN HUKUM DAN HAK ASASI RI <br>
    KANTOR WILAYAH JAWA BARAT <br>
    <b>RUMAH TAHANAN NEGARA KLAS 1
    BANDUNG</b>
</font></center></h2>
<center><b>Jl. Jakarta, Kebonwaru, Kec. Batununggal, Kota
Bandung, Jawa Barat 40272< b></center></td>
</table>
<hr><width="100" height="75"></hr>

<div class="container">

<?php
// Metode Knuth Morris Pratt
function KnuthMorrisPratt($text, $pattern) {
    $text = strtolower($text);
    $pattern = strtolower($pattern);

    $patlen = strlen($pattern);
    $textlen = strlen($text);

    $table = makeCharTable($pattern);

    for ($i=$patlen-1; $i < $textlen;) {
        $t = $i;
        for ($j=$patlen-1; $pattern[$j]==$text[$i]; $j--,$i--) {
            if($j == 0) return $i;
        }
        $i = $t;
        if(array_key_exists($text[$i], $table))
            $i = $i + max($table[$text[$i]], 1);
        else
            $i += $patlen;
    }
    return -1;
}
function makeCharTable($string) {
    $len = strlen($string);

```

```


|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>function</b> stringReverse( <b>string</b> \$string) { <b>for</b> (\$i=0; \$i < <b>strlen</b> (\$string); \$i++) {         \$table[\$string[\$i]] = <b>strlen</b> (\$string) - \$i - 1;     } <b>return</b> \$table; } ?> <div class="detail">     <center>         <div style="color: green; font-size: 20px;">Dikonfirmasi</div>     </center>     <div class="gambar">         <h4 align="center">Jawaban Konfirmasi</h4>         <table width="100%" class="table table-striped table-bordered">             <?php \$no=1; <b>foreach</b> (\$jawaban <b>as</b> \$key => \$value) { ?>                 <tr>                     <td rowspan="3"><?= \$no++ ?></td>                 <tr>                     <td><b><?= \$value->pertanyaan ?></b></td>                 <tr>                     <td><b><?= \$value->jawaban ?></b></td>                 <tr>                     <td>Hasil Pencocokan</td>                     <td><?= KnuthMorrisPratt(\$value->jawaban, \$jawaban_post[\$key]->jawaban) ?> 0 ? '<span style="color: green">Ada Kecocokan</span>' : '<span style="color: red">Tidak Ada Kecocokan</span>' ?></td>                 <?php } ?>             </tr>         </table>         <!-- " style="width: 100%;" -->     </div> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|


```

```

<div class="gambar">
    <h4 align="center">Keterangan Data</h4>
    <table width="100%" class="table table-striped table-bordered">
        <tr>
            <th>Nama</th>
            <td><?= $claim['nama_tahanan'] ?></td>
        </tr>
        <tr>
            <th>Ciri Khusus</th>
            <td><?= $claim['ciri_khusus'] ?></td>
        </tr>
        <tr>
            <th>Keterangan</th>
            <td>
                <b><?= $claim['jenis'] == 'lost' ? 'Di Input Oleh :' : 'Di Konfirmasi Oleh :' ?></b>
                <br>
                <?= $claim['jenis'] == 'lost' ? $claim['pemilik'] : $claim['penemu'] ?>
                <?php if ($claim['status'] == 1) { ?>
                <br>

                <?php } ?>
                <br>
                <b><?= $claim['jenis'] == 'lost' ? 'Di Konfirmasi Oleh :' : 'Di Input Oleh :' ?></b>
                <br>
                <?= $claim['jenis'] == 'lost' ? $claim['penemu'] : $claim['pemilik'] ?>
                <?php if ($claim['status'] == 1) { ?>
                <br>

                <?php } ?>
            </td>
        </tr>
        <tr>
            <th>Tanggal Masuk</th>

```

```

<td><?= date_format(date_create($claim['tanggal_upload']), 'd F Y') ?></td>
</tr>
<tr>
    <th>Status</th>
    <td><?= $claim['status'] == 1 ? 'Dikonfirmasi' : 'Belum Dikonfirmasi' ?></td>
</tr>
<?php
if ($claim['status'] == 1) {
?
<tr>
    <th>Tanggal Konfirmasi</th>
    <td><?= date_format(date_create($claim['tanggal_konfirmasi']), 'd F Y') ?></td>
</tr>
<?php }else{ ?>

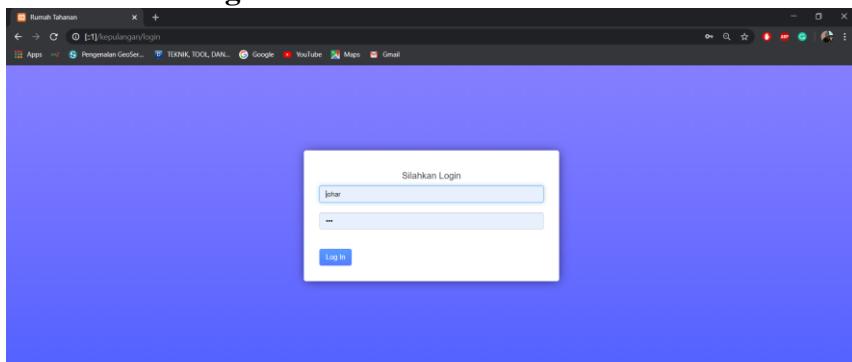
    <?php } ?>
<tr>
    <th>Berkas Tahanan Masuk</th>
    <td><img style="width: 270px; height: 300px;" src=<?=$base_url();?><?=$claim['berkas_tahanan'][0];?>></td>
</tr>
<tr>
    <th>Berkas Tahanan Keluar</th>
    <td><img style="width: 270px; height: 300px;" src=<?=$base_url();?>uploads/berkas/<?=$claim['berkas'][0];?>></td>
</tr>

<tr>
    <th>Keterangan</th>
    <td><?= $claim['keterangan'] ?></td>
</tr>
</table>

```



### 5.1.12 Source Code Login



Gambar 5. 36 Halaman Login

View Login	
<div class="login">	
<div class="login_wrapper">	
<div class="animate form login_form">	
<div class="login_content">	
<h4 align="center">Silahkan Login</h4>	
<form action=" site_url('login/aksi_login')?&gt;" method="POST"&gt;</td <td></td>	
<input type="text" name="username" class="form-control" placeholder="Username" autofocus="true">	
<input type="password" name="password" class="form-control" placeholder="Password">	

```
<br>

    <button type="submit" class="btn btn-primary
submit">Log In</button>

</form>

</div>
</div>
</div>
</div>

<style>
.login {
    height: -webkit-fill-available;
    padding: 10% 35%;
    background: linear-gradient(to bottom, #857fff,
#7a78ff, #6f70ff, #6269ff, #5362ff);
}
.login_content {
    background: #fff;
    border: 1px solid #fff;
    padding: 30px;;
    border-radius: 5px;
    -webkit-box-shadow: 1px 1px 20px -4px
rgba(0,0,0,0.75);
    -moz-box-shadow: 1px 1px 20px -4px
rgba(0,0,0,0.75);
    box-shadow: 1px 1px 20px -4px rgba(0,0,0,0.75);
}
.login_wrapper {
    margin: 0 auto;
}
</style>
```

## Controller Login

```
<?php

class Login extends CI_Controller{

    function __construct() { // memanggil sesi
        parent::__construct();
        $this->load->model('m_login');
        $this->load->model('User_model');
        $this->load->library('session');
        $this->load->library('google');
    }

    function loginClaim($id){ // login untuk claim data
        tahanan jika user belum melakukan login
        $data['login_url'] = $this->google-
        >get_login_url();
        $data['_view'] = 'loginClaim';
        $data['id'] = $id;
        $this->load->view('layouts/main_login',
        $data);
    }

    function cekLogin(){ // memasukan data user
        kedalam table database
        $cek = $this->m_login-
        >cek_login_email($_GET['email']);

        if($cek->jumlah > 0){
            $data_session = array(
                'id' => $cek->id,
                'username' => $cek-
                >username,
                'nama' => $cek->nama,
                'role' => $cek->role,
                'gender' => $cek->gender,
```

```

'email' => $cek->email,
'no_telp' => $cek-
>no_telp,
'auth_key' => $cek-
>auth_key
);

$this->session-
>set_userdata($data_session);

redirect(site_url("dashboard"));

}else{
$params = array(
'gender' => 'L',
'role' => 2,
'password' =>
md5(123123),
'username' =>
explode('@', $_GET['email'])[0],
'nama' => $_GET['name'],
'email' =>
$_GET['email'],
'no_telp' => 0,
'auth_key' =>
md5('lostfound'),
'created_at' => date('Y-m-
d H:i:s'),
'alamat' => '-',
);

$user_id = $this->User_model-
>add_user($params);

```

```

$cek = $this->m_login-
>cek_login_id($user_id);

redirect(site_url("login/insert_no_telp?id=" .
$user_id));
}

}

function aksi_login(){ // aksi login manual pada aplikasi
tanpa google login
$username = $this->input-
>post('username');
$password = $this->input->post('password');

$cek = $this->m_login-
>cek_login($username ,md5($password));

if($cek->jumlah > 0){
$data_session = array(
'id' => $cek->id,
'username' => $cek-
>username,
'nama' => $cek->nama,
'role' => $cek->role,
'gender' => $cek->gender,
'email' => $cek->email,
'no_telp' => $cek-
>no_telp,
'auth_key' => $cek-
>auth_key
);

$this->session-
>set_userdata($data_session);
}

```

```

        redirect(site_url("dashboard"));

    } else {
        redirect(site_url("login"));
    }
}

function logout(){ // logout website
    $this->session->sess_destroy();
    redirect(site_url('dashboard'));
}

```

### Model Login

```

<?php

class M_login extends CI_Model{
    function cek_login($username,$password){
        $query = $this->db->query("SELECT *,
COUNT(*) AS jumlah FROM user WHERE
username=? AND password=?", [
            $username,
            $password
        ])->row();

        return $query;
    }

    function cek_login_email($email){
        $query = $this->db->query("SELECT *,
COUNT(*) AS jumlah FROM user WHERE email=?",
[
            $email
        ])->row();
    }
}

```

```

        return $query;
    }

    function cek_login_id($id){
        $query = $this->db->query("SELECT *,
        COUNT(*) AS jumlah FROM user WHERE id=?",
        [
            $id
        ])->row();

        return $query;
    }
}

?>

```

### 5.1.13 Source Code Kategori Admin

Berfungsi sebagai main content yang akan menampilkan data data berupa data table yang akan di panggil dari database, dan menambahkan data.

Pasal Listing			Add
ID	Pasal	Actions	
9	362 KUHP Tentang Pencurian	<a href="#"></a> <a href="#"></a>	
8	363 KUHP Tentang Pencurian	<a href="#"></a> <a href="#"></a>	
7	378 KUHP Tentang Penipuan	<a href="#"></a> <a href="#"></a>	
6	127 AYAT (1) Tentang Narkotika	<a href="#"></a> <a href="#"></a>	
5	372 KUHP Tentang Penggelapan	<a href="#"></a> <a href="#"></a>	

Gambar 5. 37 Kategori

View Kategori
<div class="row">     <div class="col-md-12">         <div class="box">             <div class="box-header">                 <h3 class="box-title">Pasal Listing</h3>                 <div class="box-tools">                     <a href=" php echo site_url('kategori/add'); ?&gt;"&gt;                         class="btn btn-success btn-sm"&gt;Add&lt;/a&gt;                 &lt;/div&gt;             &lt;/div&gt;         &lt;/div&gt;     &lt;/div&gt; &lt;/div&gt; </td

```

        </div>
    </div>
    <div class="box-body">
        <table class="table table-striped table-bordered">
            <tr>
                <th>ID</th>
                <th>Pasal</th>
                <th>Actions</th>
            </tr>
            <?php foreach($kategori as $k){ ?>
            <tr>
                <td><?php echo
                    $k->id; ?></td>
                <td><?php echo
                    $k->nama_pasal; ?></td>
                <td>
                    <a href=<?php echo
                        site_url('kategori/edit/'.$k->id); ?>" class="btn btn-info btn-xs"><span class="fa fa-pencil"></span> Edit</a>
                    <a href=<?php echo
                        site_url('kategori/remove/'.$k->id); ?>" class="btn btn-danger btn-xs" onclick="return validasiDelete();"><span class="fa fa-trash"></span> Delete</a>
                </td>
            </tr>
            <?php } ?>
        </table>
        <div class="pull-right">
            <?php echo $this->pagination->create_links(); ?>
        </div>
    </div>
</div>

```

Controller Kategori
<pre> class Kategori extends CI_Controller{     var \$API = "";     function __construct()     { </pre>

```

parent::__construct();
$this-
>API="http://localhost/kepulangan_server/index.php";
$this->load->model('Kategori_model');
$this->load->library('curl');
}
function index() // data kategori wbp datahanan (admin)
{
    // $params['limit'] = RECORDS_PER_PAGE;
    // $params['offset'] = ($this->input->get('per_page')) ?
$params['per_page'] : 0;

    $config = $this->config->item('pagination');
    $config['base_url'] = site_url('kategori/index?');
    // $config['total_rows'] = $this->Kategori_model-
>get_all_kategori_count();
    $config['total_rows'] = json_decode($this->curl-
>simple_get($this->API.'/Pasal2'));
    $this->pagination->initialize($config);

    // $data['kategori'] = $this->Kategori_model-
>get_all_kategori($params);
    $data['kategori'] = json_decode($this->curl-
>simple_get($this->API.'/Pasal'));

    $data['_view'] = 'kategori/index';
    $this->load->view('layouts/main',$data);
}

function add() // menambahkan pasal baru (admin)
{
    $this->load->library('form_validation');

    $this->form_validation->set_rules('nama_pasal','Nama
Pasal','required');

    if($this->form_validation->run())
    {
        $params = array(

```

```

'id' => $this->input->post('id'),
'nama_pasal' => $this->input->post('nama_pasal'),
);

// $this->curl->simple_post($this->API.'/Pasal',
$params, array(CURLOPT_BUFFERSIZE => 10));
$kategori_id = $this->Kategori_model-
>add_kategori($params);

$test = array(
'id_pasal' => $this->input->post('id'),
'isi' => $this->input->post('isi'),
);
// $this->curl->simple_post($this->API.'/Pasal', $test,
array(CURLOPT_BUFFERSIZE => 10));
$kategori_id = $this->Kategori_model-
>add_definisi_kategori($test);

redirect('kategori/index');
}
else
{
    $data['_view'] = 'kategori/add';
    $this->load->view('layouts/main',$data);
}
}

function edit($id) // edit pasal (admin)
{
    $data['kategori'] = $this->Kategori_model-
>get_kategori($id);

    if(isset($data['kategori']['id']))
    {
        $this->load->library('form_validation');

```

```

$this->form_validation->set_rules('nama_pasal','Nama
Pasal','required');

if($this->form_validation->run())
{
    $params = array(
        'id' => $id,
        'nama_pasal' => $this->input->post('nama_pasal'),
    );

    $update = $this->curl->simple_put($this-
>API.'/Pasal', $params, array(CURLOPT_BUFFERSIZE =>
10));
    // $this->Kategori_model-
>update_kategori($id,$params);
    redirect('kategori/index');
}
else
{
    $data['_view'] = 'kategori/edit';
    $this->load->view('layouts/main',$data);
}
else
    show_error('The kategori you are trying to edit does not
exist.');
}

function remove($id) // delete pasal (admin)
{
    // $kategori = $this->Kategori_model->get_kategori($id);
    $data = array('id' => $id, );
    // $kategori = json_decode($this->curl->simple_get($this-
>API.'/Pasal', $data));
    // print_r($kategori); die;

    $this->curl->simple_delete($this->API.'/Pasal', $data,
array(CURLOPT_BUFFERSIZE => 10));
    // $this->Kategori_model->delete_kategori($id);
}

```

```
    redirect('kategori/index');

}

}
```

Model Kategori
<?php /* * Generated by CRUDigniter v3.2 * www.crudigniter.com */  class Kategori_model extends CI_Model { function __construct() { parent::__construct(); }  /* * Get kategori by id */ function get_kategori(\$id) { return \$this->db->get_where('pasal',array('id'=>\$id))->row_array(); }  /* * Get all kategori count */ function get_all_kategori_count() { \$this->db->from('pasal'); return \$this->db->count_all_results(); }  /* * Get all kategori */

```

*/
function get_all_kategori($params = array())
{
    $this->db->order_by('id', 'desc');
    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'], $params['offset']);
    }
    return $this->db->get('pasal')->result_array();
}

function getKategori(){
    return $this->db->get('pasal')->result_array();
}

/*
 * function to add new kategori
 */
function add_kategori($params)
{
    $this->db->insert('pasal',$params);
    return $this->db->insert_id();
}

function add_definisi_kategori($test)
{
    $this->db->insert('definisi_pasal',$test);
    // return $this->db->insert_id();
}

function update_kategori($id,$params)
{
    $this->db->where('id',$id);
    return $this->db->update('pasal',$params);
}

/*
 * function to delete kategori
 */
function delete_kategori($id)
{
}

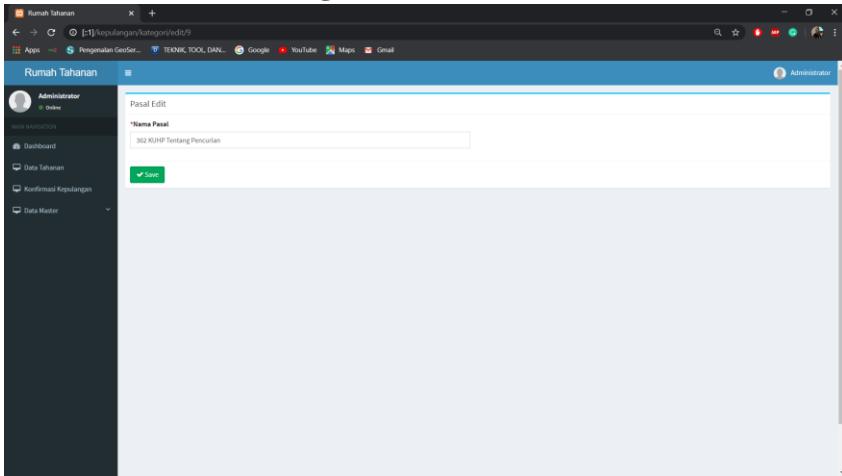
```

```

        return $this->db->delete('pasal',array('id'=>$id));
    }
}

```

### 5.1.14 Source Code Edit Kategori



Gambar 5. 38 From Edit Kategori Pasal

View Edit Kategori
<pre> &lt;div class="row"&gt;     &lt;div class="col-md-12"&gt;         &lt;div class="box box-info"&gt;             &lt;div class="box-header with-border"&gt;                 &lt;h3 class="box-title"&gt;Pasal Edit&lt;/h3&gt;             &lt;/div&gt;             &lt;?php echo form_open('kategori/edit/'.\$kategori['id']); ?&gt;             &lt;div class="box-body"&gt;                 &lt;div class="row clearfix"&gt;                     &lt;div class="col-md-6"&gt;                         &lt;label for="nama_pasal" class="control-label"&gt;&lt;span class="text- danger"&gt;*&lt;/span&gt;Nama Pasal&lt;/label&gt;                         &lt;div class="form- group"&gt;                             &lt;input type="text" name="nama_pasal" value="&lt;?php echo (\$this-&gt;</pre>

```

>input->post('nama_pasal') ? $this->input->post('nama_pasal') :
$kategori['nama_pasal']); ?>" class="form-control"
id="nama_pasal" />
<span
class="text-danger"><?php echo
form_error('nama_pasal');?></span>
</div>
</div>
</div>
</div>
<div class="box-footer">
<button type="submit" class="btn btn-success">
<i class="fa fa-
check"></i> Save
</button>
</div>
<?php echo form_close(); ?>
</div>
</div>
</div>

```

Model Edit Kategori
<pre> function get_kategori(\$id) {     return \$this-&gt;db-&gt;get_where('pasal',array('id'=&gt;\$id))-&gt;row_array(); } }  function getKategori(){     return \$this-&gt;db-&gt;get('pasal')-&gt;result_array(); } function update_kategori(\$id,\$params) {     \$this-&gt;db-&gt;where('id',\$id);     return \$this-&gt;db-&gt;update('pasal',\$params); } </pre>

### Controller Edit Kategori

```
function edit($id) // edit pasal (admin)
{
    $data['kategori'] = $this->Kategori_model->get_kategori($id);

    if(isset($data['kategori']['id']))
    {
        $this->load->library('form_validation');

        $this->form_validation->set_rules('nama_pasal','Nama Pasal','required');

        if($this->form_validation->run())
        {
            $params = array(
                'id' => $id,
                'nama_pasal' => $this->input->post('nama_pasal'),
            );

            $update = $this->curl->simple_put($this->API.'/Pasal', $params, array(CURLOPT_BUFFERSIZE => 10));
            // $this->Kategori_model->update_kategori($id,$params);
            redirect('kategori/index');
        }
        else
        {
            $data['_view'] = 'kategori/edit';
            $this->load->view('layouts/main',$data);
        }
    }
    else
        show_error('The kategori you are trying to edit does not exist.');
}
```

### 5.1.15 Bootstrap Top Menu Petugas Tahanan Masuk

Home      Data Tahanan      Logout (Hj.Rita)

Gambar 5. 39 Top Menu Petugas Tahanan Masuk

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible"
content="IE=edge">
        <title>Rumah Tahanan</title>
        <!-- Tell the browser to be responsive to screen width -->
        <meta content="width=device-width, initial-scale=1,
maximum-scale=1, user-scalable=no" name="viewport">
        <!-- Bootstrap 3.3.6 -->
        <link rel="stylesheet" href="<?php echo
site_url('resources/css/bootstrap.min.css');?>">
        <!-- Font Awesome -->
        <link rel="stylesheet" href="<?php echo
site_url('resources/css/font-awesome.min.css');?>">
        <!-- jQuery 2.2.3 -->
        <script src="<?php echo site_url('resources/js/jquery-
2.2.3.min.js');?>"></script>
    </head>

    <body>
        <nav class="navbar navbar-default">
            <div class="container-fluid">
                <!-- Brand and toggle get grouped for better mobile
display -->
                <div class="navbar-header">
                    <button type="button" class="navbar-toggle
collapsed" data-toggle="collapse" data-target="#bs-example-
navbar-collapse-1" aria-expanded="false">
                        <span class="sr-only">Toggle
navigation</span>
                        <span class="icon-bar"></span>
                </div>
            <div id="bs-example-navbar-collapse-1" class="collapse
navbar-collapse">
                <ul class="nav navbar-nav">
                    <li class="active"><a href="#">Home</a></li>
                    <li><a href="#">Data Tahanan</a></li>
                    <li><a href="#">Logout (Hj.Rita)</a></li>
                </ul>
            </div>
        </div>
    </nav>
</body>
```

```

        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
    </button>
    <a class="navbar-brand" href="/">Rumah
Tahanan</a>
    </div>

        <!-- Collect the nav links, forms, and other content
for toggling -->
        <div class="collapse navbar-collapse" id="bs-
example-navbar-collapse-1">
            <ul class="nav navbar-nav navbar-right">
                <li><a href=<?= base_url()
?>">Home</a></li>
                    <?php if (!empty($this->session-
>userdata('nama'))){ ?>
                        <?php if ($this->session->userdata('email') ==
'johar@gmail.com'){ ?>
                            <?php }else{ ?>
                            <li><a href=<?php echo
site_url('site/wbp/lost');?>">Data Tahanan</a></li>
                            <?php } ?>

                            <?php if ($this->session->userdata('email') ==
'rita5@gmail.com'){ ?>
                                <?php }else{ ?>
                                <li><a href=<?php echo
site_url('site/konfirmasi');?>">Konfirmasi
Kepulangan</a></li>
                                <li><a href=<?php echo
site_url('site/sudah_konfirmasi');?>">Sudah
Dikonfirmasi</a></li>
                                <?php } ?>
                                <li><a href=<?php echo
site_url('login/logout');?>">Logout (<?= $this->session-
>userdata('nama') ?>)</a></li>
                                <?php } else { ?>
                                    <li><a href=<?php echo
site_url('login');?>">Login</a></li>
                                    <?php } ?>

```

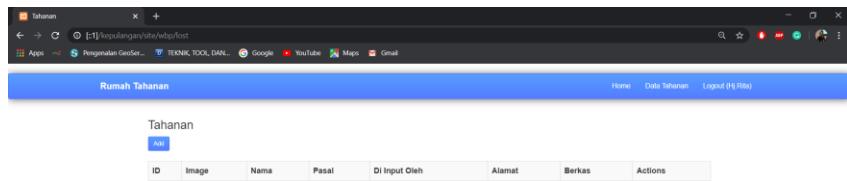
```

        </ul>
    </div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>
<!-- ./wrapper -->

<div class="container">
<?php
if(isset($_view) && $_view) {
    $this->load->view($_view);
}
?>
</div>

```

### 5.1.16 Source Code Tampilan Data Tahanan



Gambar 5. 40 Tampilan Data Tahanan

View Data Tahanan						
<div class="row">						
<div class="col-md-12">						
<div class="box">						
<div class="box-header">						
<h3 class="box-title">Tahanan</h3>						
<div class="box-tools" style="margin-bottom: 20px">						

```

        <a href="#"><?php echo
```

```
site_url('site/insert_tahanan/lost'); ?>" class="btn btn-primary
```

```
btn-sm">Add</a>
```

```
</div>
```

```
</div>
```

```
<div class="box-body">
```

```
<table class="table table-striped table-bordered">
```

```
<tr>
```

```
<th>ID</th>
```

```
<th>Image</th>
```

```
<th>Nama</th>
```

```
<th>Pasal</th>
```

```
<th>Di Input Oleh</th>
```

```
<th>Alamat</th>
```

```
<th>Berkas</th>
```

```
<th>Actions</th>
```

```
</tr>
```

```
<?php foreach($data_wbp as $b){ ?>
```

```
<tr>
```

```
<td><?= $b['id_wbp']; ?></td>
```

```
<td></td>
```

```
<td><?= $b['nama_tahanan']; ?></td>
```

```
<td><?= $b['nama_pasal']; ?></td>
```

```
<td><?= $b['nama']; ?></td>
```

```
<td><?= $b['alamat_tahanan']; ?></td>
```

```
<td></td>
```

```
<td>
```

```
<a href="#"><?php echo
```

```
site_url('site/hapus_data/'.$b['id_wbp'].'lost'); ?>" class="btn
```

```
btn-danger btn-xs"><span class="fa fa-trash"></span>
```

```
Hapus</a>
```

```
<a href="#"><?php echo
```

```
site_url('site/edit_tahanan/'.$b['id_wbp'].'lost'); ?>" class="btn
```

```
btn-primary btn-xs"><span class="fa fa-pencil"></span>
```

```
Edit</a>
```

```
</td>
```

```
</tr>
```

```
<?php } ?>
```

```

</table>
<div class="pull-right">
    <?php echo $this->pagination->create_links(); ?>
</div>
</div>
</div>
</div>
```

### Model Data Tahanan

```

<?php
/*
 * Generated by CRUDigniter v3.2
 * www.crudigniter.com
 */

class data_tahanan_model extends CI_Model
{
    function __construct()
    {
        parent::__construct();
        $this->load->library('session');
    }

    /*
     * Get data_tahanan by id
     */
    function get_data_wbp($id)
    {
        $this->db->order_by('data_tahanan.id', 'desc');
        if(isset($params) && !empty($params))
        {
            $this->db->limit($params['limit'], $params['offset']);
        }
        $this->db->select('*', 'data_tahanan.id AS id_wbp');
        $this->db->from('data_tahanan');
        $this->db->join('pasal',      'data_tahanan.pasal_id      =      pasal.id');
        $this->db->join('user', 'data_tahanan.post_by = user.id');
        $this->db->where('data_tahanan.id', $id);
```

```

        return $this->db->get()->row_array();
    }

function getKategori(){
    $tes = $this->db->get('pasal');
    return $tes;
}

/*
 * Get all data_tahanan count
 */
function get_all_data_wbp_count($tipe = null)
{
    $this->db->select('*', data_tahanan.id AS id_wbp');
    $this->db->from('data_tahanan');
    $this->db->join('pasal',      'data_tahanan.pasal_id      =
pasal.id');
    $this->db->join('user', 'data_tahanan.post_by = user.id');
    $this->db->where('status_claim', 0);
    if ($tipe == 'lost') {
        $this->db->where('jenis', 'lost');
    } else {
        $this->db->where('jenis', 'found');
    }
    if ($this->session->userdata('role') == 2) {
        $this->db->where('post_by',           $this->session-
>userdata('id'));
    }
    return $this->db->count_all_results();
}

/*
 * Get all data_tahanan
 */
function get_all_data_wbp($params = array(), $tipe = null)
{
    $this->db->order_by('data_tahanan.id', 'desc');
    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'], $params['offset']);
    }
}

```

```

}

$this->db->select('*', data_tahanan.id AS id_wbp);
$this->db->from('data_tahanan');
$this->db->join('pasal',      'data_tahanan.pasal_id      =
pasal.id');
    $this->db->join('user', 'data_tahanan.post_by = user.id');
    //  $this->db->join('berkas',   'data_tahanan.berkas      =
berkas.id_berkas');
        $this->db->where('status_claim', 0);
        if ($tipe == 'lost') {
            $this->db->where('jenis', 'lost');
        } else {
            $this->db->where('jenis', 'found');
        }
        if ($this->session->userdata('role') == 2) {
            $this->db->where('post_by',           $this->session-
>userdata('id'));
        }
        return $this->db->get()->result_array();
    }

function get_all_wbp_home_count($tipe = null)
{
    $this->db->select('*', data_tahanan.id AS id_wbp);
    $this->db->from('data_tahanan');
    $this->db->join('pasal',      'data_tahanan.pasal_id      =
pasal.id');
        $this->db->join('user', 'data_tahanan.post_by = user.id');
        $this->db->where('status_claim', 0);
        return $this->db->count_all_results();
}

function get_all_wbp_home($params = array(), $tipe = null)
{
    $this->db->order_by('data_tahanan.id', 'desc');
    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'], $params['offset']);
    }
    $this->db->select('*', data_tahanan.id AS id_wbp);
}

```

```

        $this->db->from('data_tahanan');
        $this->db->join('pasal',      'data_tahanan.pasal_id      =
pasal.id');
        $this->db->join('user', 'data_tahanan.post_by = user.id');
        $this->db->where('status_claim', 0);

        return $this->db->get()->result_array();
    }

    function get_all_wbp_search_count($q)
    {
        $this->db->select('*', data_tahanan.id AS id_wbp');
        $this->db->from('data_tahanan');
        $this->db->join('pasal',      'data_tahanan.pasal_id      =
pasal.id');
        $this->db->join('user', 'data_tahanan.post_by = user.id');
        $this->db->where('status_claim', 0);
        $this->db->like('nama_tahanan', $q);
        return $this->db->count_all_results();
    }

    function get_all_wbp_search($params = array(), $q)
    {
        // print_r($params); die;
        $this->db->order_by('data_tahanan.id', 'desc');
        if(isset($params) && !empty($params))
        {
            $this->db->limit($params['limit'], $params['offset']);
        }
        $this->db->select('*', data_tahanan.id AS id_wbp');
        $this->db->from('data_tahanan');
        $this->db->join('pasal',      'data_tahanan.pasal_id      =
pasal.id');
        $this->db->join('user', 'data_tahanan.post_by = user.id');
        $this->db->where('status_claim', 0);
        $this->db->like('nama_tahanan', $q);

        return $this->db->get()->result_array();
    }

```

```

/*
 * function to add new data_tahanan
 */
function add_data_wbp($params)
{
    $this->db->insert('data_tahanan',$params);
    return $this->db->insert_id();
}

function add_data_berkas($data)
{
    return $this->db->insert('berkas',$data);
// return $this->db->insert_id();
}

/*
 * function to update data_tahanan
 */
function update_data_wbp($id,$params)
{
    $this->db->where('id',$id);
    return $this->db->update('data_tahanan',$params);
}

/*
 * function to delete data_tahanan
 */
function delete_data_wbp($id)
{
    return $this->db->delete('data_tahanan',array('id'=>$id));
}
}

```

### Controller Data Tahanan

```

function hapus_data($params) // menghapus data
tahanan sesuai hak user
{
    $get_params = explode('%7C', $params);
    $id = $get_params[0];
}

```

```

$tipe = $get_params[1];
$data_wbp = $this->Data_tahanan_model->get_data_wbp($id);

if(isset($data_wbp['id']))
{
    $this->Data_tahanan_model->delete_data_wbp($id);
    redirect('site/wbp/'.$tipe);
}
else
    show_error('The data_wbp you are trying to delete does
not exist.');
}

function edit_tahanan($params) // edit data wbp user
{
    $get_params = explode('%7C', $params);
    $id = $get_params[0];
    $tipe = $get_params[1];

    $data['data_wbp'] = $this->Data_tahanan_model-
>get_data_wbp($id);

    if(isset($data['data_wbp']['id']))
    {
        $this->load->library('form_validation');

        $this->form_validation-
>set_rules('pasal_id','Kategori','required');
        $this->form_validation-
>set_rules('nama_tahanan','Nama Wbp','required');
        $this->form_validation-
>set_rules('ciri_khusus','Deskripsi Wbp','required');
        $this->form_validation-
>set_rules('alamat_tahanan','Lokasi Ditemukan','required');

        if($this->form_validation->run())
        {
            $config['upload_path'] = './uploads/';
            $config['allowed_types'] = 'gif|jpg|png';

```

```

// $config['file_name']          = date('YmdHi'). '-' .
$this->session->userdata('id');
$config['max_size']           = 100;
$config['max_width']          = 1024;
$config['max_height']         = 768;

$this->load->library('upload');

$this->upload->initialize($config);

if (!$this->upload->do_upload('image') and !$this-
>upload->do_upload('berkas')){
    $params = array(
        'pasal_id' => $this->input->post('pasal_id'),
        'jenis' => $tipe,
        'nama_tahanan'      =>      $this->input-
>post('nama_tahanan'),
        'ciri_khusus'        =>      $this->input-
>post('ciri_khusus'),
        'alamat_tahanan'      =>      $this->input-
>post('alamat_tahanan'),
    );
}

$this->Data_tahanan_model-
>update_data_wbp($id,$params);
redirect('site/wbp/'.$tipe);
} elseif ($this->upload->do_upload('image') and
!$this->upload->do_upload('berkas')) {
    $result1 = $this->upload->data();
    $nama_image = $result1['file_name'];
    $params = array(
        'pasal_id' => $this->input->post('pasal_id'),
        'jenis' => $tipe,
        'nama_tahanan'      =>      $this->input-
>post('nama_tahanan'),
        'image' => '/uploads/' . $nama_image,
        'ciri_khusus'        =>      $this->input-
>post('ciri_khusus'),
        'alamat_tahanan'      =>      $this->input-
>post('alamat_tahanan'),
    );
}

```

```

);

        $this->Data_tahanan_model-
>update_data_wbp($id,$params);
        redirect('site/wbp/'.$tipe);
    }elseif ($this->upload->do_upload('berkas') and
!$this->upload->do_upload('image')) {
        $result2 = $this->upload->data();
        $nama_berkas = $result2['file_name'];
        $params = array(
            'pasal_id' => $this->input->post('pasal_id'),
            'jenis' => $tipe,
            'nama_tahanan' => $this->input-
>post('nama_tahanan'),
            'berkas' => '/uploads/' . $nama_berkas,
            'ciri_khusus' => $this->input-
>post('ciri_khusus'),
            'alamat_tahanan' => $this->input-
>post('alamat_tahanan'),
        );
}

        $this->Data_tahanan_model-
>update_data_wbp($id,$params);
        redirect('site/wbp/'.$tipe);
} else {
    if($this->upload->do_upload('image')){
        $result1 = $this->upload->data();
        $nama_image = $result1['file_name'];
    }
    if($this->upload->do_upload('berkas')){
        $result2 = $this->upload->data();
        $nama_berkas = $result2['file_name'];
    }
    $params = array(
        'pasal_id' => $this->input->post('pasal_id'),
        'jenis' => $tipe,
        'nama_tahanan' => $this->input-
>post('nama_tahanan'),
        'image' => '/uploads/' . $nama_image,
        'berkas' => '/uploads/' . $nama_berkas,
    )
}

```

```

        'ciri_khusus'      =>      $this->input-
>post('ciri_khusus'),
        'alamat_tahanan'    =>      $this->input-
>post('alamat_tahanan'),
);
// print_r($params);die;

        $this->Data_tahanan_model-
>update_data_wbp($id,$params);
        redirect('site/wbp/'.$tipe);
    }
}
else
{
    $this->load->model('Kategori_model');
    $data['all_kategori'] = $this->Kategori_model-
>get_all_kategori();

    $data['_view'] = 'site/edit_tahanan';
    $data['title'] = 'Edit Tahanan ' . ($tipe == 'lost' ?
'Hilang' : 'Temuan');
    $data['tipe'] = $tipe;
    $data['id'] = $id;
    $this->load->view('layouts/main_user',$data);
}
}
else
    show_error('The data_wbp you are trying to edit does
not exist.');
}

function insert_tahanan($tipe) // insert data wbp kedalam data
wbp hilang
{
    $this->load->library('form_validation');

    $this->form_validation-
>set_rules('pasal_id','Kategori','required');
    $this->form_validation->set_rules('nama_tahanan','Nama
Wbp','required');

```

```

$this->form_validation-
>set_rules('ciri_khusus','Deskripsi Wbp','required');
    $this->form_validation-
>set_rules('alamat_tahanan','Lokasi Ditemukan','required');

if($this->form_validation->run())
{
    $config['upload_path']      = './uploads/';
    $config['allowed_types']    = 'gif|jpg|png';
    // $config['file_name']       = date('YmdHis'). '-' . $this-
>session->userdata('id');
    $config['max_size']         = 100;
    $config['max_width']        = 1024;
    $config['max_height']       = 768;

    $this->load->library('upload');

    $this->upload->initialize($config);

    if($this->upload->do_upload('image')){
        $result1 = $this->upload->data();
        $nama_image = $result1['file_name'];
    }
    if($this->upload->do_upload('berkas')){
        $result2 = $this->upload->data();
        $nama_berkas = $result2['file_name'];
    }
    $params = array(
        'pasal_id' => $this->input->post('pasal_id'),
        'post_by' => $this->session->userdata('id'),
        'jenis' => $tipe,
        'nama_tahanan'      =>           $this->input-
>post('nama_tahanan'),
        'image' => '/uploads/' . $nama_image,
        'berkas' => '/uploads/' . $nama_berkas,
        'status_claim' => 0,
        'created_at' => date('Y-m-d H:i:s'),
        'ciri_khusus' => $this->input->post('ciri_khusus'),
        'alamat_tahanan'     =>           $this->input-
>post('alamat_tahanan'),

```

```

);
    $data_wbp_id      =      $this->Data_tahanan_model-
>add_data_wbp($params);
    redirect('site/add_jawaban/' . $data_wbp_id . '%7C' .
$tipe);

} else {
    $this->load->model('Kategori_model');
    $data['all_kategori']      =      $this->Kategori_model-
>get_all_kategori();

    $data['_view'] = 'site/tambah_tahanan';
    $data['title'] = 'Tambah Tahanan ' . ($tipe == 'lost' ?
'Hilang' : 'Temuan');
    $data['tipe'] = $tipe;

    $this->load->view('layouts/main_user',$data);
}
}

```

### 5.1.17 Source Code Input Data Tahanan

The screenshot shows a web application interface for adding a detainee. The title bar says 'Tahanan'. The main content area is titled 'Rumah Tahanan' and contains a form titled 'Tambah Tahanan'. The form includes fields for 'Pasal' (with a dropdown menu showing 'Pith Pasal'), 'Nama' (text input), 'Alamat' (text input), 'Image' (file input with placeholder 'Choose File'), and 'berkas' (file input with placeholder 'Choose File'). A blue 'Simpan' button is at the bottom.

Gambar 5. 41 From Input Tahanan

View Tambah Tahanan
<div class="row">

```

<div class="col-md-12">
    <div class="box box-info">
        <div class="box-header with-border">
            <h3 class="box-title">Tambah Tahanan</h3>
        </div>
        <form action=<?=site_url('site/insert_tahanan/'.$tipe)?>" method="POST" enctype="multipart/form-data">
            <div class="box-body">
                <div class="row clearfix">
                    <div class="col-md-6">
                        <label for="pasal_id" class="control-label"><span class="text-danger">*</span>Pasal</label>
                        <div class="form-group">
                            <select name="pasal_id" class="form-control">
                                <option value="">Pilih Pasal</option>
                                <?php
                                    foreach($all_kategori as $kategori)
                                    {
                                        $selected = ($kategori['id'] == $this->input->post('pasal_id')) ? ' selected="selected"' : "";
                                        echo '<option value="'.$kategori['id'].'.'.$selected.'>'.$kategori['nama_pasal'].'</option>';
                                    }
                                ?>
                        </div>
                    </div>
                </div>
            </div>
        </form>
    </div>
</div>

```

```

        </select>

        <span class="text-danger"><?php echo
form_error('pasal_id');?></span>
                                </div>
                            </div>
                            <div class="col-
md-6">
                                <label
for="nama_tahanan" class="control-label"><span class="text-
danger">*</span>Nama</label>
                                <div
class="form-group">

        <input type="text" name="nama_tahanan"
value=<?php echo $this->input->post('nama_tahanan'); ?>">
        <input type="hidden" name="berkas" value=<?php
echo rand(1111,99999); ?>" class="form-control" id="berkas"
/>

        <span class="text-danger"><?php echo
form_error('berkas');?></span>
                                </div>
                            </div>
                            <div class="col-
md-6">
                                <label
for="ciri_khusus" class="control-label"><span class="text-
danger">*</span>Ciri Khusus</label>
                                <div
class="form-group">

        <textarea name="ciri_khusus" class="form-control"
id="ciri_khusus"><?php echo $this->input->post('ciri_khusus');
?></textarea>

```

```

<span class="text-danger"><?php echo
form_error('ciri_khusus');?></span>
                                </div>
                            </div>
                            <div class="col-
md-6">
                                <label
for="alamat_tahanan" class="control-label"><span class="text-
danger">*</span>Alamat </label>
                                <div
class="form-group">

                                <textarea name="alamat_tahanan" class="form-control"
id="alamat_tahanan"><?php echo $this->input-
>post('alamat_tahanan'); ?></textarea>

                                <span class="text-danger"><?php echo
form_error('alamat_tahanan');?></span>
                                </div>
                            </div>
                            <div class="col-
md-6">
                                <label
for="image" class="control-label"><span class="text-
danger">*</span>Image</label>
                                <div
class="form-group">

                                <input type="file" name="image" class="form-control"
id="image" required />

                                <span class="text-danger"><?php echo
form_error('image');?></span>
                                </div>
                            </div>
                            <div class="col-
md-6">

```

```

        <label
for="berkas" class="control-label"><span class="text-
danger">*</span>berkas</label>
                <div
class="form-group">

        <input type="file" name="berkas" class="form-control"
id="berkas" required />

        <span class="text-danger"><?php echo
form_error('berkas');?></span>
                </div>
            </div>
            </div>
            </div>
        </div>
    </div>
    <div class="box-footer">
        <button type="submit"
class="btn btn-primary">
            <i class="fa fa-
check"></i> Save
        </button>
    </div>
</form>
</div>
</div>
</div>

```

### Model Tambah Tahanan

```

<?php
/*
 * Generated by CRUDigniter v3.2
 * www.crudigniter.com
 */

class data_tahanan_model extends CI_Model
{
    function __construct()
    {

```

```

parent::__construct();
$this->load->library('session');
}

/*
 * Get data_tahanan by id
 */
function get_data_wbp($id)
{
    $this->db->order_by('data_tahanan.id', 'desc');
    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'], $params['offset']);
    }
    $this->db->select('*', data_tahanan.id AS id_wbp');
    $this->db->from('data_tahanan');
    $this->db->join('pasal', 'data_tahanan.pasal_id = pasal.id');
    $this->db->join('user', 'data_tahanan.post_by = user.id');
    $this->db->where('data_tahanan.id', $id);
    return $this->db->get()->row_array();
}

function getKategori(){
    $tes = $this->db->get('pasal');
    return $tes;
}

/*
 * Get all data_tahanan count
 */
function get_all_data_wbp_count($tipe = null)
{
    $this->db->select('*', data_tahanan.id AS id_wbp');
    $this->db->from('data_tahanan');
    $this->db->join('pasal', 'data_tahanan.pasal_id = pasal.id');
    $this->db->join('user', 'data_tahanan.post_by = user.id');
    $this->db->where('status_claim', 0);
    if ($tipe == 'lost') {
        $this->db->where('jenis', 'lost');
    } else {
}

```

```

        $this->db->where('jenis', 'found');
    }
    if ($this->session->userdata('role') == 2) {
        $this->db->where('post_by', $this->session-
>userdata('id'));
    }
    return $this->db->count_all_results();
}

/*
 * Get all data_tahanan
 */
function get_all_data_wbp($params = array(), $tipe = null)
{
    $this->db->order_by('data_tahanan.id', 'desc');
    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'], $params['offset']);
    }
    $this->db->select('*', 'data_tahanan.id AS id_wbp');
    $this->db->from('data_tahanan');
    $this->db->join('pasal', 'data_tahanan.pasal_id = pasal.id');
    $this->db->join('user', 'data_tahanan.post_by = user.id');
    // $this->db->join('berkas', 'data_tahanan.berkas =
berkas.id_berkas');
    $this->db->where('status_claim', 0);
    if ($tipe == 'lost') {
        $this->db->where('jenis', 'lost');
    } else {
        $this->db->where('jenis', 'found');
    }
    if ($this->session->userdata('role') == 2) {
        $this->db->where('post_by', $this->session-
>userdata('id'));
    }
    return $this->db->get()->result_array();
}

function get_all_wbp_home_count($tipe = null)
{

```

```

$this->db->select('*', data_tahanan.id AS id_wbp);
$this->db->from('data_tahanan');
$this->db->join('pasal', 'data_tahanan.pasal_id = pasal.id');
$this->db->join('user', 'data_tahanan.post_by = user.id');
$this->db->where('status_claim', 0);
return $this->db->count_all_results();
}

function get_all_wbp_home($params = array(), $tipe = null)
{
    $this->db->order_by('data_tahanan.id', 'desc');
    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'], $params['offset']);
    }
    $this->db->select('*', data_tahanan.id AS id_wbp);
    $this->db->from('data_tahanan');
    $this->db->join('pasal', 'data_tahanan.pasal_id = pasal.id');
    $this->db->join('user', 'data_tahanan.post_by = user.id');
    $this->db->where('status_claim', 0);

    return $this->db->get()->result_array();
}

function get_all_wbp_search_count($q)
{
    $this->db->select('*', data_tahanan.id AS id_wbp);
    $this->db->from('data_tahanan');
    $this->db->join('pasal', 'data_tahanan.pasal_id = pasal.id');
    $this->db->join('user', 'data_tahanan.post_by = user.id');
    $this->db->where('status_claim', 0);
    $this->db->like('nama_tahanan', $q);
    return $this->db->count_all_results();
}

function get_all_wbp_search($params = array(), $q)
{
    // print_r($params); die;
    $this->db->order_by('data_tahanan.id', 'desc');
    if(isset($params) && !empty($params))

```

```

{
    $this->db->limit($params['limit'], $params['offset']);
}
$this->db->select('*', data_tahanan.id AS id_wbp');
$this->db->from('data_tahanan');
$this->db->join('pasal', 'data_tahanan.pasal_id = pasal.id');
$this->db->join('user', 'data_tahanan.post_by = user.id');
$this->db->where('status_claim', 0);
$this->db->like('nama_tahanan', $q);

return $this->db->get()->result_array();
}

/*
 * function to add new data_tahanan
 */
function add_data_wbp($params)
{
    $this->db->insert('data_tahanan',$params);
    return $this->db->insert_id();
}

function add_data_berkas($data)
{
    return $this->db->insert('berkas',$data);
    // return $this->db->insert_id();
}

/*
 * function to update data_tahanan
 */
function update_data_wbp($id,$params)
{
    $this->db->where('id',$id);
    return $this->db->update('data_tahanan',$params);
}

/*
 * function to delete data_tahanan
 */

```

```

function delete_data_wbp($id)
{
    return $this->db->delete('data_tahanan',array('id'=>$id));
}

```

### Controller Tambah Data Tahanan

```

function insert_tahanan($tipe) // insert data wbp kedalam data
wbp hilang
{
    $this->load->library('form_validation');

    $this->form_validation-
>set_rules('pasal_id','Kategori','required');
    $this->form_validation->set_rules('nama_tahanan','Nama
Wbp','required');
    $this->form_validation->set_rules('ciri_khusus','Deskripsi
Wbp','required');
    $this->form_validation-
>set_rules('alamat_tahanan','Lokasi Ditemukan','required');

    if($this->form_validation->run())
    {
        $config['upload_path']      = './uploads/';
        $config['allowed_types']   = 'gif|jpg|png';
        // $config['file_name']       = date('YmdHis'). '-' .
$this->session->userdata('id');
        $config['max_size']         = 100;
        $config['max_width']        = 1024;
        $config['max_height']       = 768;

        $this->load->library('upload');

        $this->upload->initialize($config);

        if($this->upload->do_upload('image')){
            $result1 = $this->upload->data();
            $nama_image = $result1['file_name'];
        }
        if($this->upload->do_upload('berkas')){

    
```

```

        $result2 = $this->upload->data();
        $nama_berkas = $result2['file_name'];
    }
    $params = array(
        'pasal_id' => $this->input->post('pasal_id'),
        'post_by' => $this->session->userdata('id'),
        'jenis' => $tipe,
        'nama_tahanan' => $this->input-
>post('nama_tahanan'),
        'image' => '/uploads/' . $nama_image,
        'berkas' => '/uploads/' . $nama_berkas,
        'status_claim' => 0,
        'created_at' => date('Y-m-d H:i:s'),
        'ciri_khusus' => $this->input->post('ciri_khusus'),
        'alamat_tahanan' => $this->input-
>post('alamat_tahanan'),
    );
}

$data_wbp_id = $this->Data_tahanan_model-
>add_data_wbp($params);
redirect('site/add_jawaban/' . $data_wbp_id . '%7C' .
$tipe);

} else {
    $this->load->model('Kategori_model');
    $data['all_kategori'] = $this->Kategori_model-
>get_all_kategori();

    $data['_view'] = 'site/tambah_tahanan';
    $data['title'] = 'Tambah Tahanan ' . ($tipe == 'lost' ?
'Hilang' : 'Temuan');
    $data['tipe'] = $tipe;

    $this->load->view('layouts/main_user',$data);
}
}

```

### 5.1.18 From Input Jawaban

JAWABAN DARI PERTANYAAN

Barang yang di curi?  
Tuliskan Jawaban

Pendidikan?  
Tuliskan Jawaban

Pekerjaan?  
Tuliskan Jawaban

Agama?  
Tuliskan Jawaban

Umur?  
Tuliskan Jawaban

Status Pemilikan?  
Tuliskan Jawaban

Gambar 5. 42 From Input Jawaban

View Input Jawaban

```
<div class="row">
    <div class="col-md-12">
        <div class="box box-info">
            <div class="box-header with-border">
                <h3 class="box-title" align="center">JAWABAN DARI PERTANYAAN</h3>
            </div>
            <?php echo form_open('site/add_jawaban/' . $id . '%7C' . $tipe); ?>
            <div class="box-body">
                <div class="row clearfix">
                    <?php foreach ($pertanyaan as $key) { ?>
                        <label><?= $key->pertanyaan ?></label>
                        <br>
                        <input type="hidden" name="pertanyaan[]" class="form-control" value="<?= $key->id ?>">
                        <br>
                    <?php } ?>
                </div>
            </div>
        </div>
    </div>
</div>
```

```

<textarea
name="jawaban[]" rows="3" class="form-control"
placeholder="Tuliskan Jawaban"></textarea>

<br>
<?php } ?>
</div>
</div>
<div class="box-footer">
    <button type="submit" class="btn btn-success">
        <i class="fa fa-check"></i> Save
    </button>
</div>
<?php echo form_close(); ?>
</div>
</div>
</div>

```

Model Jawaban
<pre> function insert_jawaban_post(\$wbp, \$pertanyaan, \$jawaban) {     return \$this-&gt;db-&gt;query("INSERT INTO jawaban_post (data_tahanan_id, pertanyaan_id, jawaban) VALUES ('" . \$wbp . "', '" . \$pertanyaan . "', '" . \$jawaban . "')"); } function get_all_pertanyaan(\$params = array()) {     \$this-&gt;db-&gt;order_by('id', 'desc');     if(isset(\$params) &amp;&amp; !empty(\$params))     {         \$this-&gt;db-&gt;limit(\$params['limit'], \$params['offset']);     }     \$this-&gt;db-&gt;select('pertanyaan.*', 'pasal.nama_pasal');     \$this-&gt;db-&gt;from('pertanyaan');     \$this-&gt;db-&gt;join('pasal', 'pertanyaan.pasal_id = pasal.id');      return \$this-&gt;db-&gt;get()-&gt;result_array(); } function get_all_pertanyaan_wbp(\$kategori) </pre>

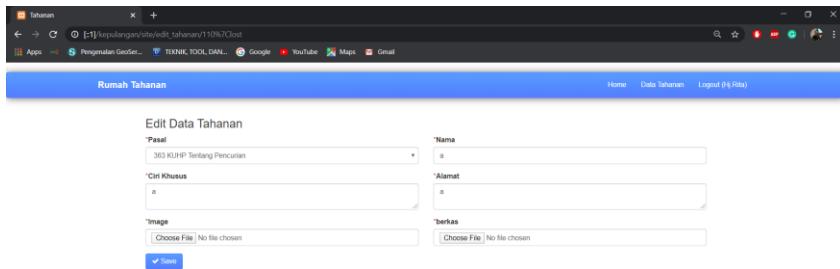
```

{
    $query = $this->db->query("SELECT * FROM
pertanyaan WHERE pasal_id=". $kategori)->result();
    // var_dump('<pre>', $query); exit;
    return $query;
}

```

Controller Jawaban
<pre> function add_jawaban(\$params) // menambahkan jawaban dari data yang di temukan {     \$get_params = explode('%7C', \$params);     \$id = \$get_params[0];     \$tipe = \$get_params[1];     \$this-&gt;load-&gt;library('form_validation');      \$this-&gt;form_validation- &gt;set_rules('jawaban[]','Jawaban','required');      if(\$this-&gt;form_validation-&gt;run())     {         foreach (\$this-&gt;input-&gt;post('jawaban') as \$key =&gt; \$value) {             \$this-&gt;Pertanyaan_model-&gt;insert_jawaban_post(\$id, \$this-&gt;input-&gt;post('pertanyaan')[[\$key]], \$value);         }          redirect('site/wbp/' . \$tipe);     }     else     {         \$data['wbp'] = \$this-&gt;Data_tahanan_model- &gt;get_data_wbp(\$id);         \$data['pertanyaan'] = \$this-&gt;Pertanyaan_model- &gt;get_all_pertanyaan_wbp(\$data['wbp'][['pasal_id']]);         \$data['id'] = \$id;         \$data['tipe'] = \$tipe;         \$data['_view'] = 'site/jawaban';         \$this-&gt;load-&gt;view('layouts/main_user',\$data); }}</pre>

### 5.1.19 Source Code Edit Data Tahanan



Rumah Tahanan

Edit Data Tahanan

\*Pasal  
363 KUHP Tentang Percurian

\*Ciri Khasus  
a

\*Nama  
a

\*Alamat  
a

\*Image  
Choose File | No file chosen

\*berkas  
Choose File | No file chosen

**Save**

Gambar 5. 43 Edit Data Tahanan

```
View Edit Data Tahanan
<div class="row">
    <div class="col-md-12">
        <div class="box box-info">
            <div class="box-header with-border">
                <h3 class="box-title">Edit Data Tahanan</h3>
            </div>
            <?php echo
form_open_multipart('site/edit_tahanan/'.$id.'/'.$tipe); ?>
            <div class="box-body">
                <div class="row clearfix">
                    <div class="col-md-6">
                        <label for="pasal_id" class="control-label"><span class="text-danger">*</span>Pasal</label>
                        <div class="form-group">
                            <select name="pasal_id" class="form-control">
                                <option value="">Pilih Pasal</option>
                                <?php
                                foreach($all_kategori as $kategori)
                                {
                                    $selected = ($kategori['id'] ==
$data_wbp['pasal_id']) ? ' selected="selected"' : "";
                                    <option value="$kategori[id]" $selected>$kategori[nama]
                                }
                            </select>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

```

        echo '<option value="'.$kategori['id'].'"'
'.selected.''.$kategori['nama_pasal'].'</option>';
    }
?>
</select>
<span class="text-danger"><?php echo
form_error('pasal_id');?></span>
</div>
</div>
<div class="col-md-6">
    <label for="nama_tahanan" class="control-
label"><span class="text-danger">*</span>Nama</label>
    <div class="form-group">
        <input type="text" name="nama_tahanan"
value=<?php echo ($this->input->post('nama_tahanan') ?
$this->input->post('nama_tahanan') :
$data_wbp['nama_tahanan']); ?>" class="form-control"
id="nama_tahanan" />
        <span class="text-danger"><?php echo
form_error('nama_tahanan');?></span>
    </div>
    </div>
    <div class="col-md-6">
        <label for="ciri_khusus" class="control-
label"><span class="text-danger">*</span>Ciri
Khusus</label>
        <div class="form-group">
            <textarea name="ciri_khusus" class="form-
control" id="ciri_khusus"><?php echo ($this->input-
>post('ciri_khusus') ? $this->input->post('ciri_khusus') :
$data_wbp['ciri_khusus']); ?></textarea>
            <span class="text-danger"><?php echo
form_error('ciri_khusus');?></span>
        </div>
    </div>
    <div class="col-md-6">
        <label for="alamat_tahanan" class="control-
label"><span class="text-danger">*</span>Alamat</label>
        <div class="form-group">

```

```

        <textarea name="alamat_tahanan"
class="form-control" id="alamat_tahanan"><?php echo ($this-
>input->post('alamat_tahanan') ? $this->input-
>post('alamat_tahanan') : $data_wbpl['alamat_tahanan']);?
></textarea>
        <span class="text-danger"><?php echo
form_error('alamat_tahanan');?></span>
        </div>
    </div>
    <div class="col-md-6">
        <label for="image" class="control-label"><span
class="text-danger">*</span>Image</label>
        <div class="form-group">
            <input type="file" name="image"
class="form-control" id="image" />
            <span class="text-danger"><?php echo
form_error('image');?></span>
        </div>
    </div>
    <div class="col-md-6">
        <label for="berkas" class="control-
label"><span class="text-danger">*</span>berkas</label>
        <div class="form-group">
            <input type="file" name="berkas"
class="form-control" id="berkas"/>
            <span class="text-danger"><?php echo
form_error('berkas');?></span>
        </div>
    </div>
    <div class="box-footer">
        <button type="submit" class="btn btn-primary" ">
            <i class="fa fa-check"></i> Save
        </button>
    </div>
    <?php echo form_close(); ?>
</div>
</div>
</div>

```

### Model Edit Data Tahanan

```
function get_data_wbp($id)
{
    $this->db->order_by('data_tahanan.id', 'desc');
    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'], $params['offset']);
    }
    $this->db->select('*', data_tahanan.id AS id_wbp);
    $this->db->from('data_tahanan');
    $this->db->join('pasal', 'data_tahanan.pasal_id = pasal.id');
    $this->db->join('user', 'data_tahanan.post_by = user.id');
    $this->db->where('data_tahanan.id', $id);
    return $this->db->get()->row_array();
}
function update_data_wbp($id,$params)
{
    $this->db->where('id',$id);
    return $this->db->update('data_tahanan',$params);
}
```

### Controller Edit Data Tahanan

```
function edit_tahanan($params) // edit data wbp user
{
    $get_params = explode('%7C', $params);
    $id = $get_params[0];
    $tipe = $get_params[1];

    $data['data_wbp']      =      $this->Data_tahanan_model-
>get_data_wbp($id);

    if(isset($data['data_wbp']['id']))
    {
        $this->load->library('form_validation');

        $this->form_validation-
>set_rules('pasal_id','Kategori','required');
```

```

        $this->form_validation-
>set_rules('nama_tahanan','Nama Wbp','required');
        $this->form_validation-
>set_rules('ciri_khusus','Deskripsi Wbp','required');
        $this->form_validation-
>set_rules('alamat_tahanan','Lokasi Ditemukan','required');

        if($this->form_validation->run())
{
    $config['upload_path']      = './uploads/';
    $config['allowed_types']    = 'gif|jpg|png';
    // $config['file_name']       = date('YmdHi'). '-' .
$this->session->userdata('id');
    $config['max_size']         = 100;
    $config['max_width']        = 1024;
    $config['max_height']       = 768;

    $this->load->library('upload');

    $this->upload->initialize($config);

    if  (!$this->upload->do_upload('image')  and  !$this-
>upload->do_upload('berkas')){
        $params = array(
            'pasal_id' => $this->input->post('pasal_id'),
            'jenis' => $tipe,
            'nama_tahanan'      =>      $this->input-
>post('nama_tahanan'),
            'ciri_khusus'        =>      $this->input-
>post('ciri_khusus'),
            'alamat_tahanan'     =>      $this->input-
>post('alamat_tahanan'),
        );
    }

    $this->Data_tahanan_model-
>update_data_wbp($id,$params);
    redirect('site/wbp/'.$tipe);
} elseif  ($this->upload->do_upload('image')  and
!$this->upload->do_upload('berkas')) {
    $result1 = $this->upload->data();
}

```

```

$nama_image = $result1['file_name'];
$params = array(
    'pasal_id' => $this->input->post('pasal_id'),
    'jenis' => $tipe,
    'nama_tahanan' => $this->input-
>post('nama_tahanan'),
    'image' => '/uploads/' . $nama_image,
    'ciri_khusus' => $this->input-
>post('ciri_khusus'),
    'alamat_tahanan' => $this->input-
>post('alamat_tahanan'),
);

$this->Data_tahanan_model-
>update_data_wbp($id,$params);
redirect('site/wbp/'.$tipe);
}elseif ($this->upload->do_upload('berkas') and
!$this->upload->do_upload('image')) {
$result2 = $this->upload->data();
$nama_berkas = $result2['file_name'];
$params = array(
    'pasal_id' => $this->input->post('pasal_id'),
    'jenis' => $tipe,
    'nama_tahanan' => $this->input-
>post('nama_tahanan'),
    'berkas' => '/uploads/' . $nama_berkas,
    'ciri_khusus' => $this->input-
>post('ciri_khusus'),
    'alamat_tahanan' => $this->input-
>post('alamat_tahanan'),
);

$this->Data_tahanan_model-
>update_data_wbp($id,$params);
redirect('site/wbp/'.$tipe);
}else {
if($this->upload->do_upload('image')){
$result1 = $this->upload->data();
$nama_image = $result1['file_name'];
}
}

```

```

        if($this->upload->do_upload('berkas')){
            $result2 = $this->upload->data();
            $nama_berkas = $result2['file_name'];
        }
        $params = array(
            'pasal_id' => $this->input->post('pasal_id'),
            'jenis' => $tipe,
            'nama_tahanan' => $this->input->post('nama_tahanan'),
            'image' => '/uploads/' . $nama_image,
            'berkas' => '/uploads/' . $nama_berkas,
            'ciri_khusus' => $this->input->post('ciri_khusus'),
            'alamat_tahanan' => $this->input->post('alamat_tahanan'),
        );
        // print_r($params);die;

        $this->Data_tahanan_model->update_data_wbp($id,$params);
        redirect('site/wbp/'.$tipe);
    }
}
else
{
    $this->load->model('Kategori_model');
    $data['all_kategori'] = $this->Kategori_model->get_all_kategori();

    $data['_view'] = 'site/edit_tahanan';
    $data['title'] = 'Edit Tahanan ' . ($tipe == 'lost' ? 'Hilang' : 'Temuan');
    $data['tipe'] = $tipe;
    $data['id'] = $id;
    $this->load->view('layouts/main_user',$data);
}
}
else
    show_error('The data_wbp you are trying to edit does not exist.');
}

```

# BAB VI

## PENERAPAN ALGORITMA KNUTH MORRIS PRATT

### 6.1 Alasan Menggunakan Algoritma Knuth Morris Pratt

Alasan menggunakan algoritma *Knuth Morris Pratt* karena algoritma ini adalah algoritma yang tepat dalam melakukan *string matching* karena algoritma *Knuth Morris Pratt* melakukan pergeseran karakter per karakter hingga mendapatkan kecocokan string dan algoritma ini memiliki otak buatan yang memungkinkan tidak terjadinya pencocokan yang tidak berguna sehingga pencocokan menjadi semakin sedikit dan kompleksitas semakin kecil.

### 6.2 Simulasi Algoritma Knuth Morris Pratt

```
<?php
    // Metode Knuth Morris Pratt
    function KnuthMorrisPratt($text, $pattern) { //fungsi ini butuh 2 parameter
        $text = strtolower($text); //ini membuat parameter yang isinya kecil
        $pattern = strtolower($pattern); //ini membuat parameter yang isinya kecil
        $patlen = strlen($pattern); //fungsi ini untuk mengembalikan panjang string
        $textlen = strlen($text); //fungsi ini untuk mengembalikan panjang string
        $table = makeCharTable($pattern);

        for ($i=$patlen-1; $i < $textlen;) { //fungsi ini untuk menghitung
            $t = $i;
            for ($j=$patlen-1; $pattern[$j]==$text[$i]; $j--, $i--) {
                if($j == 0) return $i;
            }
            $i = $t;
            if(array_key_exists($text[$i], $table))
                $i = $i + max($table[$text[$i]], 1);
            else
                $i += $patlen;
        }
        return -1;
    }

    function makeCharTable($string) { //fungsi ini membutuhkan parameter
        $len = strlen($string); //ini untuk membuat variabel $len dimana $len = panjang string
        $table = array(); //membuat variabel berisi array
        for ($i=0; $i < $len; $i++) { //perhitungan, jadi $i adalah 0, 1, 2, ..., len-1
            $table[$string[$i]] = $len - $i - 1; //setelah perhitungan
        }
        return $table; //mengembalikan hasil array tadi berapa
    }
}
```

Gambar 6. 1 Algoritma Knuth Morris Pratt

Gambar di atas adalah algoritma yang berfungsi sebagai pencocokan jawaban yang di mana di lakukan saat melakukan proses verifikasi kepulangan.

### Metode Knuth Morris Pratt

1. function KnuthMorrisPratt(\$text, \$pattern) {  
fungsi ini butuh 2 parameter yaitu text dan pattern}
2. \$text = strtolower(\$text);  
ini membuat parameter yang isinya dari fungsi fungsi sebelumnya tapi di buat lowercase semua
3. \$pattern = strtolower(\$pattern);  
ini membuat parameter yang isinya dari fungsi fungsi sebelumnya tapi di buat lowercase semua
4. \$patlen = strlen(\$pattern);  
fungsi ini untuk mengembalikan panjang string, string yang di maksud a  
dalah variabel \$pattern
5. \$textlen = strlen(\$text);  
fungsi ini untuk mengembalikan panjang string, string yang di maksud a  
dalah variabel \$text
6. \$table = makeCharTable(\$pattern);  
Membuat fungsi table makechartable
7. For (\$i=\$patlen-1; \$i < \$textlen;) {  
fungsi ini untuk menghitung dimana \$i hasilnya \$patlen dikurang 1, \$i s  
elalu lebih kecil dari variabel \$textlen
8. \$t = \$i;  
Variable \$t = \$i
9. for (\$j=\$patlen-1; \$pattern[\$j]==\$text[\$i]; \$j--, \$i--) {  
fungsi ini untuk menghitung dimana \$j hasilnya \$patlen sama dengan 1\$
10. if(\$j == 0) return \$i;  
jika \$j sama dengan 0, maka kembali ke \$i nilai pergeseran

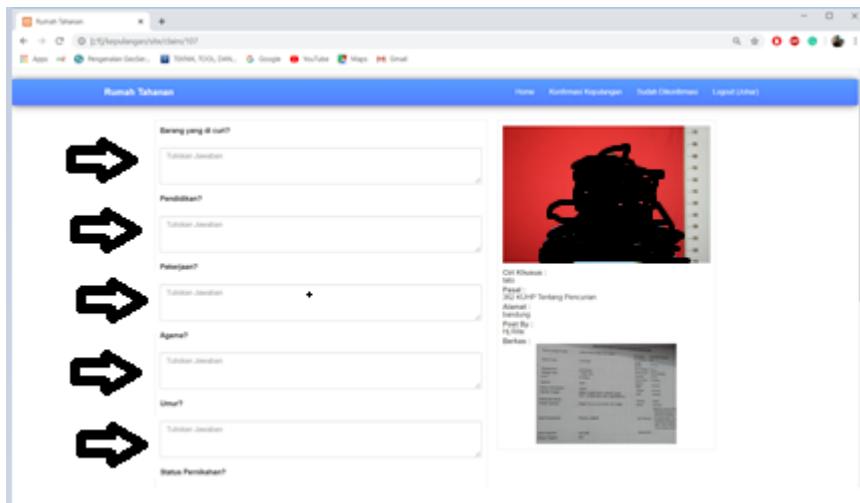
11. `$i = $t;`  
    *\$i sama dengan data table*
12. `if(array_key_exists($text[$i], $table))`  
        *jika array tersedia maka akan melakukan pergeseran table*
13. `$i = $i + max($table[$text[$i]], 1);`  
        *Melakukan pergeseran table array*
14. `$i += $patlen;`  
        *Melakukan pergeseran table array*
15. `return -1;`  
        *kembali ke kondisi awal*
16. `function makeCharTable($string) {`  
        *fungsi ini membutuhkan parameter \$string*
17. `$len = strlen($string);`  
        *ini untuk membuat variabel \$len dimana akan mengembalikan panjang kata dari \$string*
18. `$table = array();`  
        *membuat variabel berisi array*
19. `for ($i=0; $i < $len; $i++) {`  
        *perhitungan, jadi \$i adalah 0, dan \$i selalu lebih kecil dari \$len*
20. `$table[$string[$i]] = $len - $i - 1;`  
        *setelah perhitungan dijalankan, akan membuat lagi variabel \$table yang array tadi di tampung oleh \$string isinya \$len-\$i-1*
21. `return $table;`  
        *mengembalikan hasil array tadi berapa*

Hasil dari kodingan di atas bertujuan untuk:

1. Melakukan pencarian pattern text dari paling kiri ke kanan apakah text tersebut tersedia didalam pattern

2. Jika huruf paling k tersedia pada pattern maka akan dilanjutkan ke proses pergeseran
3. Melakukan pergeseran pattern text
4. Apakah text yang di inputkan memiliki nilai kecocokan dengan pattern jika memiliki kecocokan maka pencarian akan selesai dan dinyatakan string matching, dan jika tidak ada kecocokan akan terjadi miss match maka pencarian akan selesai

### 6.2.1 Hasil Pencocokan Jawaban Identitas



Gambar 6. 2 Input Data Identitas Tahanan

Jika petugas kepulangan melakukan input data dengan data wbp yang sesuai yaitu identitas dan tindak kejahatan maka kepulangan akan di verifikasi.

Pencocokan	Kondisi
1	Tidak Ada Kecocokan Dikonfirmasi oleh : Johar Barang yang dicuri? a
2	Pencocokan Dikonfirmasi oleh : Johar Pendidikan? a
3	Pencocokan Dikonfirmasi oleh : Johar Pekerjaan? a
	Pencocokan Ada Kecocokan

Nama	Asep Saepudin
Ciri Khusus	tato abstrak
Keterangan	Di Input Oleh : Johar Di Konfirmasi Oleh : Hj.Rita
Tanggal Masuk	10 December 2019
Status	Belum Dikonfirmasi

Gambar 6. 3 Hasil Pencocokan Algoritma Knuth Morris Pratt

Gambar di atas adalah algoritma hasil dari penggunaan algoritma *Knuth Morris Pratt*.

ID	Image	Nama	Keterangan	Tanggal Masuk	Tanggal Konfirmasi	Status	Actions
96		asep	Di Konfirmasi Oleh : Johar  Di Input Oleh : Hj.Rita	05 January 2020	05 January 2020	Dikonfirmasi	<a href="#">Edit</a> <a href="#">Print</a>

Gambar 6. 4 Data Tahanan Yang Sudah Di Konfirmasi

Hasil dari verifikasi jawaban akan mendapatkan bukti print out dan status akan berubah menjadi dikonfirmasi.

### 6.2.2 Kelebihan *Algrotima Knuth Morris Pratt*

Salah satu kelebihan algoritma KMP ketimbang algoritma brute force adalah adanya otak buatan yang memungkinkan tidak terjadinya pencocokan yang tidak berguna sehingga pencocokan menjadi semakin sedikit dan kompleksitas semakin kecil.

### **6.2.3 Kekurangan *Algrotima Knuth Morris Pratt***

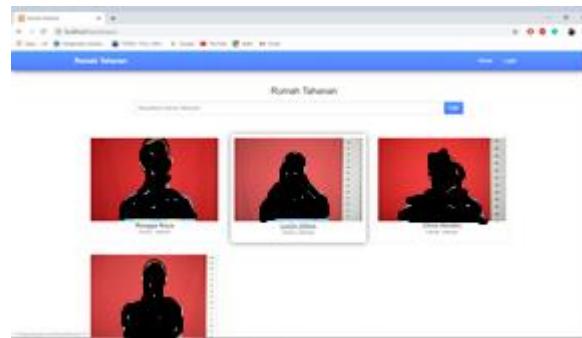
Kekurangan *Algrotima Knuth Morris Pratt* yakni kurang signifikan dalam memberikan hasil yang optimal ketika jumlah apphabet semakin banyak karena ada kemungkinan kesalahan pencocokan.

# BAB VII

## TAMPILAN ANTARMUKA

### 7.1 Halaman Menu Utama

Halaman ini merupakan halaman untuk melihat daftar tahanan, form *Login*, dan juga menampilkan data tahanan yang di temukan.



Gambar 7. 1 Tampilan Antarmuka Halaman Menu Utama

Petunjuk pemakaian:

1. Untuk menu utama terdapat data tahanan.
2. Untuk menu *Login* untuk masuk sebagai member atau *admin*

### 7.2 Halaman Data Tahanan

Halaman ini merupakan tempat *user* yang memasukn data tahanan saat tahan datang.

Image	Name	Post	Di Input Oleh	Aksi
	Rusdiyanto	PTSI KUMP	HQ File	<a href="#">Hapus</a> <a href="#">Edit</a>
	Lucky Muli	PTSI KUMP (1) LULUS 08-08-2009, PUSAT, TPI LULUS 08-08-2009	HQ File	<a href="#">Hapus</a> <a href="#">Edit</a>
	Dina Pratiwi	PTSI KUMP	HQ File	<a href="#">Hapus</a> <a href="#">Edit</a>
	Yunita	PTSI KUMP	HQ File	<a href="#">Hapus</a> <a href="#">Edit</a>

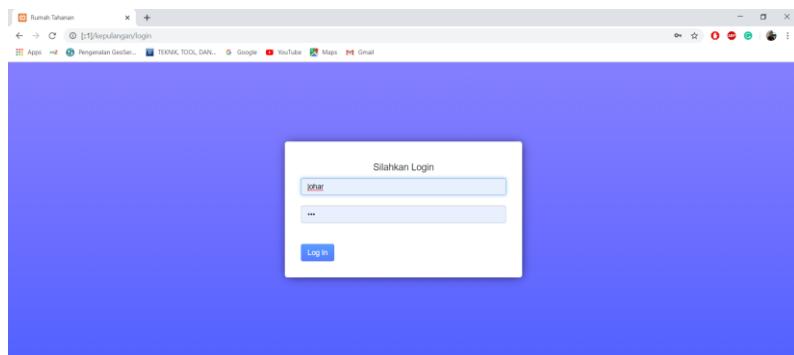
Gambar 7. 2 Tampilan Antarmuka Halaman Data Tahanan

Petunjuk pemakaian:

1. Menekan tombol add untuk menambahkan data bahanan.
2. Menekan tombol edit untuk melakukan proses edit.

### 7.3 Halaman Login

Halaman ini merupakan halaman untuk masuk ke menu utama jika yang masuk *user* dan menu *admin* jika yang masuk *admin*, dengan memasukkan *username* dan *password*.



---

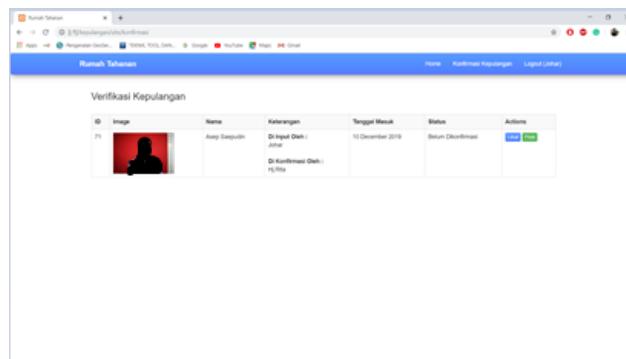
Gambar 7. 3 Tampilan Antarmuka Halaman Login

Petunjuk pemakaian:

1. Memasukkan *username* dan *password* kemudian klik tombol *Sign in*, jika yang masuk *admin* maka akan menampilkan menu *admin* dan jika yang masuk *user* akan menampilkan menu utama.

## 7.4 Halaman Konfirmasi

Halaman ini merupakan halaman untuk melihat konfirmasi yang dilakukan oleh *user*.



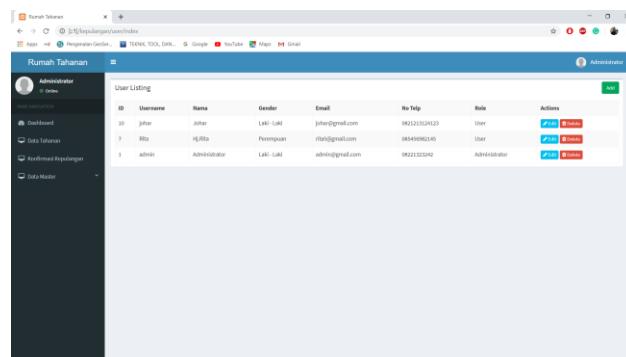
Gambar 7. 4 Tampilan Antarmuka Halaman konfirmasi

Petunjuk pemakaian :

1. Menekan tombol Lihat untuk mengubah status konfirmasi dan akan dikonfirmasi oleh petugas jika datanya *valid* dan berkasnya lengkap.
2. Menekan tombol *print* untuk melakukan print berkas kepulangan.

## 7.5 Halaman Data User

Halaman ini merupakan halaman untuk melihat, menginput, mengedit, dan menghapus data *user* dan hanya bisa diakses oleh *admin*.



Gambar 7. 5 Tampilan Antarmuka Halaman Data User

# BAB VIII

---

## KESIMPULAN

### 8.1 Kesimpulan

Setelah melakukan analisis, perancangan dan implementasi aplikasi verifikasi kepulangan warga binaan pemasyarakatan menggunakan *framework CodeIgniter*, maka dapat disimpulkan sebagai beriku :

1. Algoritma *Knuth Morris Pratt* yang di implementasikan dapat mencocokan data yang telah di inputkan dan hasil pencocokan tersebut akan merubah status tahanan yang akan dipulangkan.
2. Dengan adanya aplikasi verifikasi untuk kepulangan warga binaan pemasyarakatan menggunakan algoritma *Knuth Morris Pratt* akan membantu mempercepat proses verifikasi kepulangan untuk warga binaan pemasyarakatan yang dimana proses ini dibantu oleh algoritma saat pencocokan *string* untuk verifikasi kepulangan.

### 8.2 Kesimpulan Metode

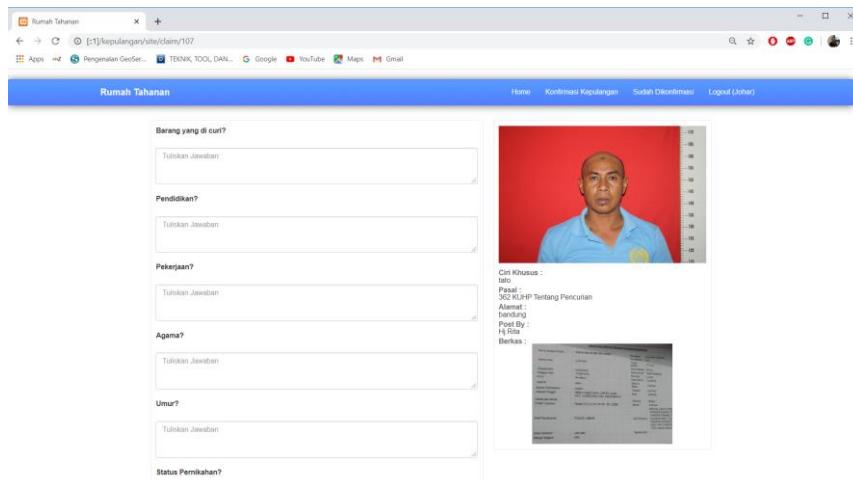
Setelah melakukan analisis sistem, dapat disimpulkan bahwa :

1. Membangun aplikasi verifikasi kepulangan tahanan di Rutan Klas 1 Bandung berbasis website menggunakan Bahasa pemrograman *PHP*, *Database MySQL* dan untuk meningkatkan keamanan rutan klas 1 bandung yaitu dengan cara menempelkan Metode Algoritma *Knuth Morris Pratt* kedalam form inputan identitas tahanan dengan mencocokan data dengan database rutan klas 1 bandung dengan cara melakukan pergeseran setiap karakter yang akan dicocokan sampai dinyatakan *String Matching*, dan jika tidak ada kecocokan maka akan dinyatakan data yang di inputkan tidak memiliki kesamaan *string*.

2. Cara algoritma *Knuth Morris Pratt* dalam melakukan pencocokan, *retrieve* yaitu langkah pertama algoritma *Knuth Morris Pratt* akan mencari huruf paling kiri yang sama dan akan melakukan terus menerus pergeseran jika text tersebut belum di temukan, jika sudah ditemukan algoritma *Knuth Morris Pratt* akan berhenti melakukan pencocokan *string*. Dan jika tidak ada yang sama dari keseluruhan data algoritma *Knuth Morris Pratt* Pun akan berhenti dan menyatakan bahwa kalimat yang di cari tidak ada yang sesuai.

### 8.3 Kesimpulan Pengujian Sistem

Setelah melakukan beberapa tahapan dan melakukan implementasi metode pada system, menyimpulkan bahwa hasil implementasi *Knuth Morris Pratt* pada system sesuai dengan pergeseran manual pada string yang digunakan, Jika pattern pada database dan *text* yang di inputkan sesuai maka pendaftaran kunjungan diverifikasi, dan jika pendaftaran kunjungan tidak sesuai maka hasilnya akan ditolak.



Gambar 8. 1 From Input Identitas Untuk Pencocokan

Rumah Tahanan

ASEP

Dikonfirmasi

Verifikasi Jawaban		Keterangan Data	
1 (Jawaban Masuk)	Jenis Barang yang di gunakan? sabu	Nama	asep
(Jawaban Keluar)	sabu	Ciri Khusus	tato naga
Hasil Pencocokan	Ada Kecocokan	Pasal	127 AYAT (*) Teritang Narkotika
2 (Jawaban Masuk)	Pendidikan? smp	Definisi	Setiap Penyalah Gunaan Narkotika Golongan I bagi diri sendiri dipidana dengan pidana penjara paling lama 4 (empat) tahun, Narkotika Golongan II bagi diri sendiri dipidana dengan pidana penjara paling lama 2 (dua) tahun; dan Narkotika Golongan III bagi diri sendiri dipidana dengan pidana penjara paling lama 1 (satu) tahun.
(Jawaban Keluar)	smp	Keterangan	Di Konfirmasi Oleh : Johar
Hasil Pencocokan	Ada Kecocokan		Di Input Oleh : Hj.Rita
3 (Jawaban Masuk)	Pekerjaan? swasta	Tanggal Masuk	05 January 2020
(Jawaban Keluar)	swasta	Status	Dikonfirmasi
Hasil Pencocokan	Ada Kecocokan	Tanggal Konfirmasi	05 January 2020
4 (Jawaban Masuk)	Agama? Islam	Berkas	

Gambar 8. 2 From Hasil Pencocokan

Rumah Tahanan

Home Konfirmasi Kepulangan Sudah Dikonfirmasi Logout (Johar)

Daftar Tahanan Sudah Dikonfirmasi

[Print This Page](#)

ID	Image	Nama	Keterangan	Tanggal Masuk	Tanggal Konfirmasi	Status	Actions
96		asep	Di Konfirmasi Oleh : Johar  Di Input Oleh : Hj.Rita	05 January 2020	05 January 2020	Dikonfirmasi	<a href="#">Edit</a> <a href="#">Print</a>

Gambar 8. 3 From Yang Di Konfirmasi

id	data_tahanan_id	input_oleh	konfirmasi_oleh	status	berkas	keterangan
96	108	10	7	1	asep_saepudin6.jpg	Berkas Tahanan,Berkas Penahanan,Berkas A

Gambar 8. 4 Data Base Data Tahanan