

# WEB PROGRAMING II

## Studi Kasus Rental Mobil



TIM PENYUSUN MODUL

## **KATA PENGANTAR**

Segala puji syukur kepada Tuhan Yang Maha Esa karena berkat rahmatNya penulisan modul Mata Kuliah Web Programming II dapat terselesaikan dengan baik. Modul ini disusun untuk memenuhi kebutuhan mahasiswa dalam mata kuliah Web Programming II yang disajikan dalam bentuk praktikum dan diharapkan dapat membekali mahasiswa dalam memahami Pembuatan web dengan menggunakan Framework Codeigniter.

Modul web Programming II materi bahasan dibatasi sampai dengan halaman Administrator (Back-End). Diakhir perkuliahan mahasiswa diharapkan mampu mengimplementasikannya dalam bentuk final proect yang harus dipresentasikan sebagai syarat kelulusan mata kuliah Web Programming II. Teknik penyajiannya dilakukan secara terpadu dan sistematis.

Seperti layaknya sebuah modul, maka pembahasan dimulai dengan menjelaskan target pembelajaran yang hendak dicapai. Dengan demikian pengguna modul ini secara mandiri dapat mengukur tingkat ketuntasan yang dicapainya.

Penulis menyadari sepenuhnya bahwa modul ini tentu memiliki banyak kekurangan. Untuk itu penulis dengan lapang dada menerima masukan dan kritik yang konstruktif dari berbagai pihak demi kesempurnaannya di masa yang akan datang. Semoga modul ini dapat bermanfaat bagi para pembaca.

Jakarta, Juni 2020

**Tim Penulis**

## DAFTAR ISI

HALAMAN SAMPUL .....	i
KATA PENGANTAR.....	ii
DAFTAR ISI .....	iii
PERTEMUAN 1 PENJELASAN KONTRAK KULIAH, PENEGASAN TUGAS .....	1
1.1    Penjelasan Kontrak Kuliah dan Penegasan Tugas .....	1
PERTEMUAN 2 PENGENALAN CODE IGNITER .....	1
2.1    Framework Web .....	2
2.2    Codeigniter .....	3
2.3    Instalasi Software .....	3
2.4    Instalasi Codeigniter.....	3
2.5    Struktur Folder Codeigniter .....	5
PERTEMUAN 3 MODEL, VIEW, CONTROLLER.....	9
3.1    Pengertian MVC.....	9
3.2    Menentukan Kontroler Standar (Default) .....	14
3.3    Menghilangkan index.php pada codeigniter .....	15
3.4    Aturan merancang Model, View, dan Controller .....	16
3.5    Parsing Data (mengirimkan nilai) dari controller Ke View .....	17
PERTEMUAN 4 MEMBUAT TEMPLATE (TEMPLATING) WEB SEDERHANA .....	21
4.1    Membuat Template Sederhana yang dinamis .....	22
4.2    Seting base_url () pada codeigniter .....	26
PERTEMUAN 5 MEMBUAT FORM VALIDASI PADA CODEIGNITER .....	31
5.1    Form Validasi .....	31
5.2    Mekanisme Validasi Data di dalam Form .....	31
5.3    Memuat Library Form_validation .....	31
PERTEMUAN 6 STUDI KASUS RENTAL MOBIL .....	33
6.1    Persiapan Membuat Aplikasi Rental Mobil .....	34
6.2    Setting Base URL.....	36
6.3    Merancang & Membuat Database Aplikasi rental_mobil dan Konfigurasi database di Codeigniter .....	37
6.4    Membuat Koneksi Database Codeigniter dengan MySQL .....	38
6.5    Membuat Model untuk CRUD .....	40
PERTEMUAN 7 REVIEW MATERI.....	43
PERTEMUAN 8 UJIAN TENGAH SEMESTER .....	45
PERTEMUAN 9 MEMBUAT FORM LOGIN DAN PESAN NOTIFIKASI.....	46
9.1    Membuat Halaman Login.....	46

9.2	Membuat Halaman Admin .....	56
PERTEMUAN 10 MEMBUAT FORM REGISTRASI USER.....		68
10.1	Membuat helper.....	69
10.2	Membuat Form Registrasi.....	75
10.3	Membuat Halaman My Profile dan Ubah Profile .....	80
PERTEMUAN 11 MEMBUAT KATEGORI MOBIL .....		85
11.1	Membuat Tampil Data Kategori .....	86
11.2	Hapus data kategori .....	89
PERTEMUAN 12 MEMBUAT CRUD DATA MOBIL .....		90
12.1	Menampilkan Data Mobil dan tambah Mobil .....	91
12.2	Update Data Mobil .....	100
12.3	Menghapus Data Mobil .....	103
PERTEMUAN 13-15 PRESENTASI PROJEK .....		104
PERTEMUAN 16 UJIAN AKHIR SEMESTER .....		105
DAFTAR PUSTAKA.....		106

**PERTEMUAN 1**  
**PENJELASAN KONTRAK KULIAH, PENEGASAN TUGAS**

**1.1 Penjelasan Kontrak Kuliah dan Penegasan Tugas**

**A. Tugas Projek (Kelompok)**

1. Tugas project diadakan untuk memperoleh nilai dan pengganti UTS dan UAS. Tugas ini dikerjakan secara kelompok dengan 1 kelompok sebanyak 5 mahasiswa atau disesuaikan dengan jumlah mahasiswa dalam satu kelas.
2. Aplikasi web nativ diperbolehkan didapat dari berbagai sumber.

**B. Tugas Mingguan (Kelompok)**

1. Tugas mingguan mulai dikerjakan di minggu pertama untuk disetorkan dan didemonstrasikan di minggu berikutnya. Begitu seterusnya di setiap minggu.
2. Bentuk tugas mingguan adalah mengimplementasikan materi tiap pertemuan yang ada di modul dimulai dari pertemuan ke 2 sampai materi pertemuan terakhir.

**Tugas 1**

- a. Mahasiswa mengerjakan materi yang ada di pertemuan 2 yaitu installasi Codeigniter dan mengerjakan contoh 1 yang ada pada modul halaman 3-4 dan halaman 7
- b. Point a dan b akan didemonstrasikan dan dikumpulkan di pertemuan selanjutnya

## **PERTEMUAN 2**

### **PENGENALAN CODE IGNITER**

Pada bab ini diharapkan mahasiswa dapat memahami materi-materi fundamental yang diperlukan sebelum mempelajari framework codeigniter, yaitu bagaimana cara melakukan installing perangkat-perangkat lunak yang diperlukan untuk dapat bekerja dengan codeigniter seperti PHP, web server, database server, dan framework codeigniter itu sendiri. Mahasiswa juga diharapkan dapat memahami pola desain atau arsitektur dari Model-View-Controller (MVC), yang merupakan syarat mutlak dalam proses pengembangan aplikasi menggunakan codeigniter. Sebelum memulai materi ada baiknya untuk mempersiapkan tools yang dibutuhkan untuk mendukung pembelajaran materi web programming 2. Tools yang dipergunakan untuk materi web programming 2 ini antara lain :

- Xampp : Dengan versi PHP 5.6 atau terbaru
- Code Igniter Versi 3
- Adobe/ Macromedia Dreamweaver CS atau editor sejenisnya (untuk pembelajaran kita gunakan tools yang memiliki bagian views dan code untuk memudahkan dalam medesain)
- Browser Google chrome , mozilla, Opera, Safari yang update agar mensupport fungsi HTML5

#### **2.1 Framework Web**

Web Application Framework (WAF), atau sering disingkat web framework, adalah Suatu kumpulan kode berupa pustaka (library) dan alat (tool) yang dipadukan sedemikian rupa menjadi kerangka kerja (framework) guna memudahkan dan mempercepat proses pengembangan aplikasi.

Framework web untuk PHP:

- ✓ Codeigniter
- ✓ Yii
- ✓ Slim framework
- ✓ Zend framework
- ✓ Laravel

- ✓ Symfony
- ✓ CakePHP
- ✓ Phalcon
- ✓ Kohana
- ✓ FuelPHP, dll

Sebagian besar dari framework di atas mengimplementasikan pola desain Model-View-Controller (MVC), yang memisahkan bagian kode untuk penanganan proses bisnis dengan bagian kode untuk keperluan presentasi (tampilan).

## 2.2 Codeigniter

Codeigniter adalah framework web untuk bahasa pemrograman PHP, yang dibuat oleh Rick Ellis pada tahun 2006, penemu dan pendiri EllisLab ([www.ellislab.com](http://www.ellislab.com)). EllisLab merupakan suatu tim kerja yang berdiri pada tahun 2002 yang bergerak di bidang pembuatan software dan tool untuk para pengembang web. Sejak tahun 2014 sampai sekarang, EllisLab telah menyerahkan hak kepemilikan codeigniter ke British Columbia Institute of Technology (BCIT) saat ini situs resmi codeigniter adalah [www.codeigniter.com](http://www.codeigniter.com).

## 2.3 Instalasi Software

Software yang dibutuhkan pada pembelajaran kali ini yaitu:

- a. Web server (Apache2Triad, WAMPP server, Xampp Server, dll)

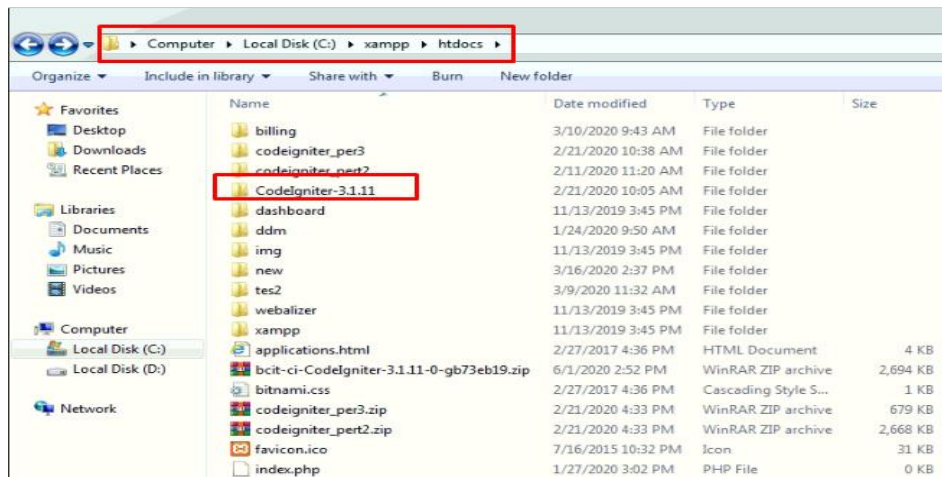
Untuk Xampp server bisa didownload di <https://www.apachefriends.org/>

- b. Text Editor (dream weaver, notepad, notepad++, sublime text, atom, visual studio code, dll)
- c. Codeigniter <https://codeigniter.com/>
- d. Web Browser (Mozilla Firefox, google chrome, IE, Safari, UCBrowser, Opera, dll).

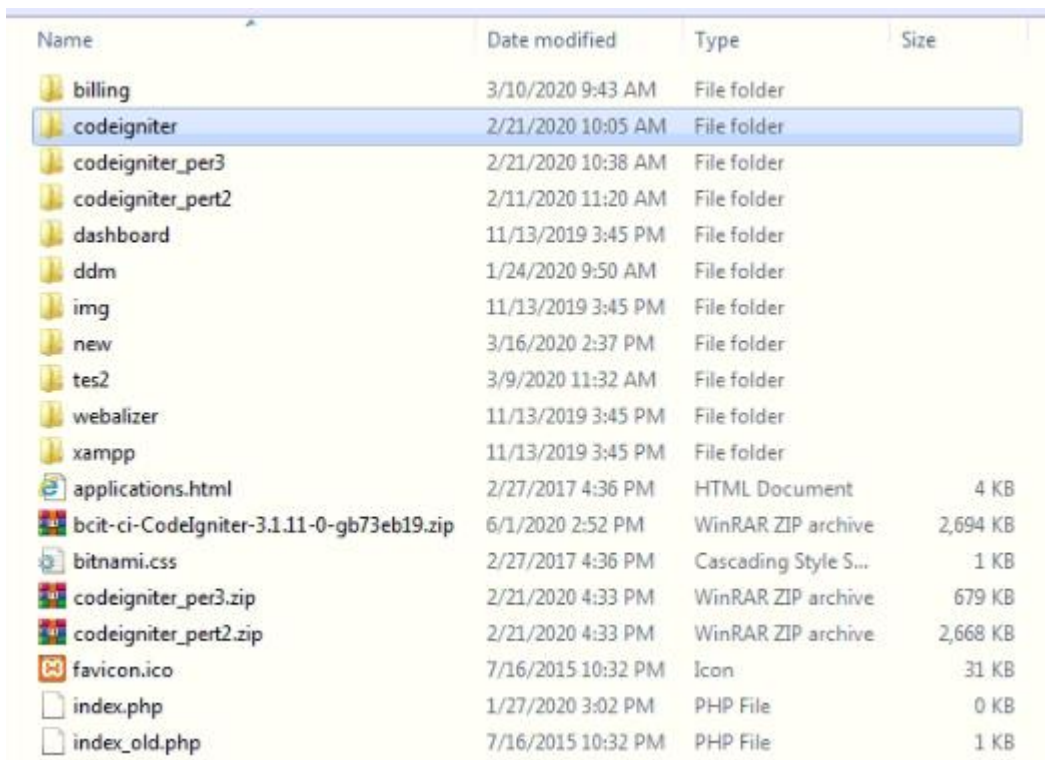
## 2.4 Instalasi Codeigniter

- a. Download Package codeigniter pada situs resminya yang sudah disebutkan di atas.
- b. Saat ini versi dari codeigniter adalah 3.1.11, sedangkan untuk versi 4 untuk menggunakan php versi 7.2
- c. Setelah di download ekstrak file codeigniter pada direktori **C:\xampp\htdocs**



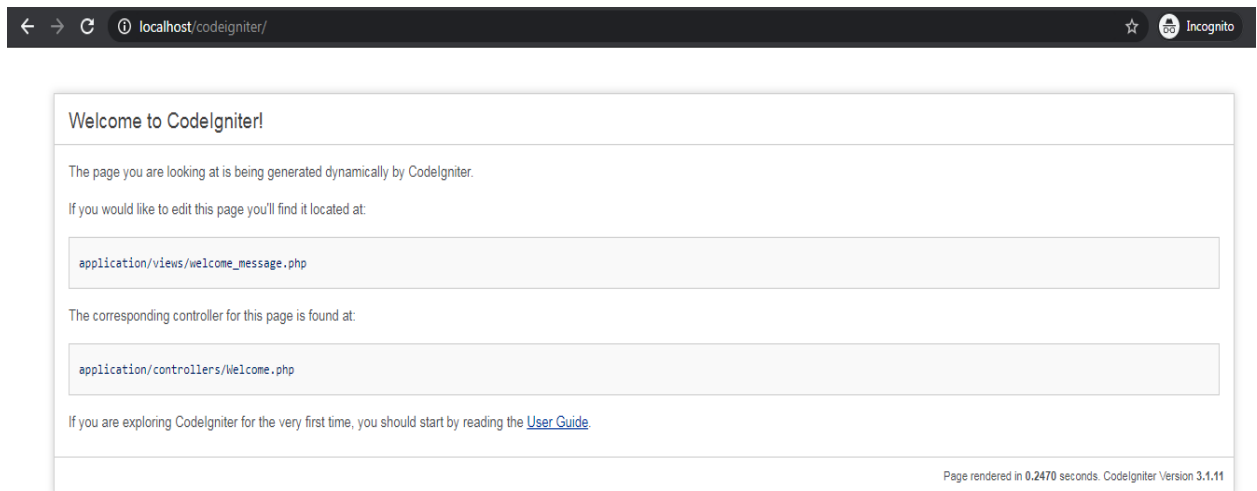


- d. lalu ubah folder **Codeigniter** menjadi sesuai keinginan kita. disini saya akan ubah folder codeigniter tersebut menjadi **codeigniter**.



- e. Setelah itu, sekarang coba akses folder tersebut melalui browser.
- f. jangan lupa untuk menyalakan **XAMPP** terlebih dahulu.
- g. lalu akses dengan url **http://localhost/nama\_folder\_project** pada tutorial kali ini saya beri nama dengan **codeigniter**

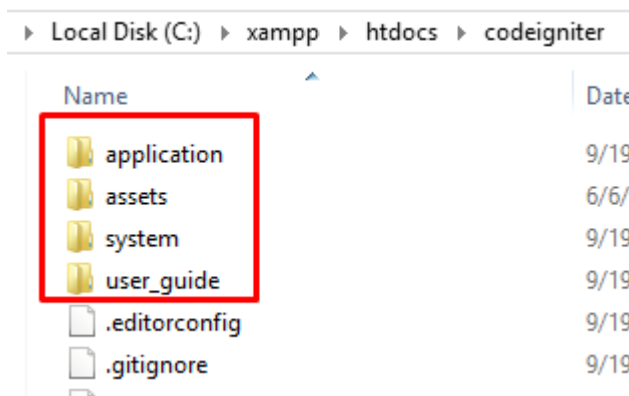




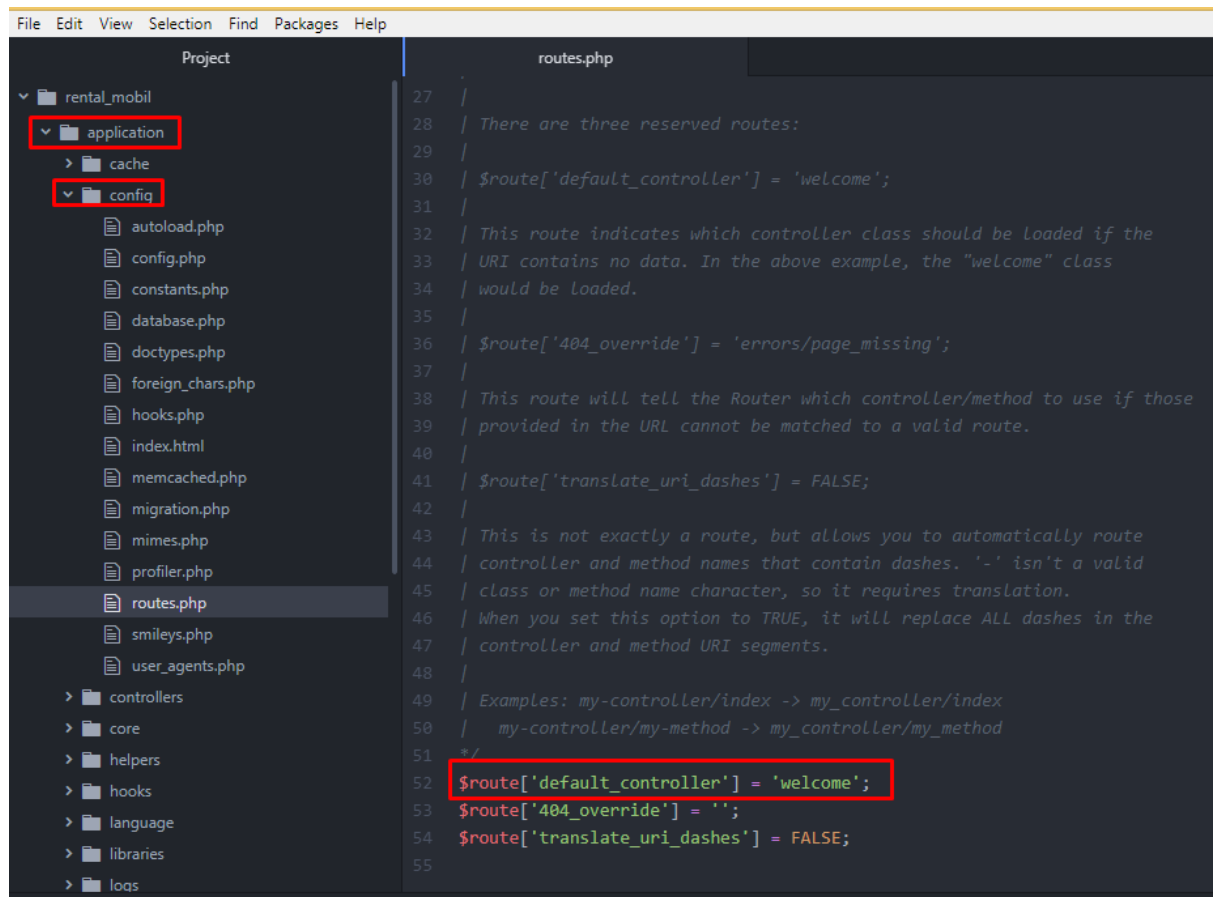
- h. Jika sudah tampil halaman **Welcome Codeigniter**, itu artinya anda telah berhasil melakukan instalasi codeigniter.

## 2.5 Struktur Folder Codeigniter

Di dalam folder codeigniter terdapat 3 folder utama yaitu **Application**, **System**, **User Guide**. Folder **application** berguna sebagai tempat untuk mendvelop aplikasi nanti, Folder **system** berisi library atau kumpulan fungsi-fungsi dasar Codeigniter (CI), sedangkan **User Guide** berisi sebagai buku panduan atau dokumentasi dari codeigniter.



Lalu kita fokus pada folder **application/config** dan semua berawal dari file **routes.php**. Apabila kita buka file tersebut, maka kita akan menemukan **default\_controller** yang telah



terisi dengan nama **“welcome”**, yang mana **default\_controller** ini berfungsi ketika user tidak melakukan atau memasukkan apapun pada URL browser, maka secara otomatis mencari controller dengan nama **“welcome”**. (lihat gambar).



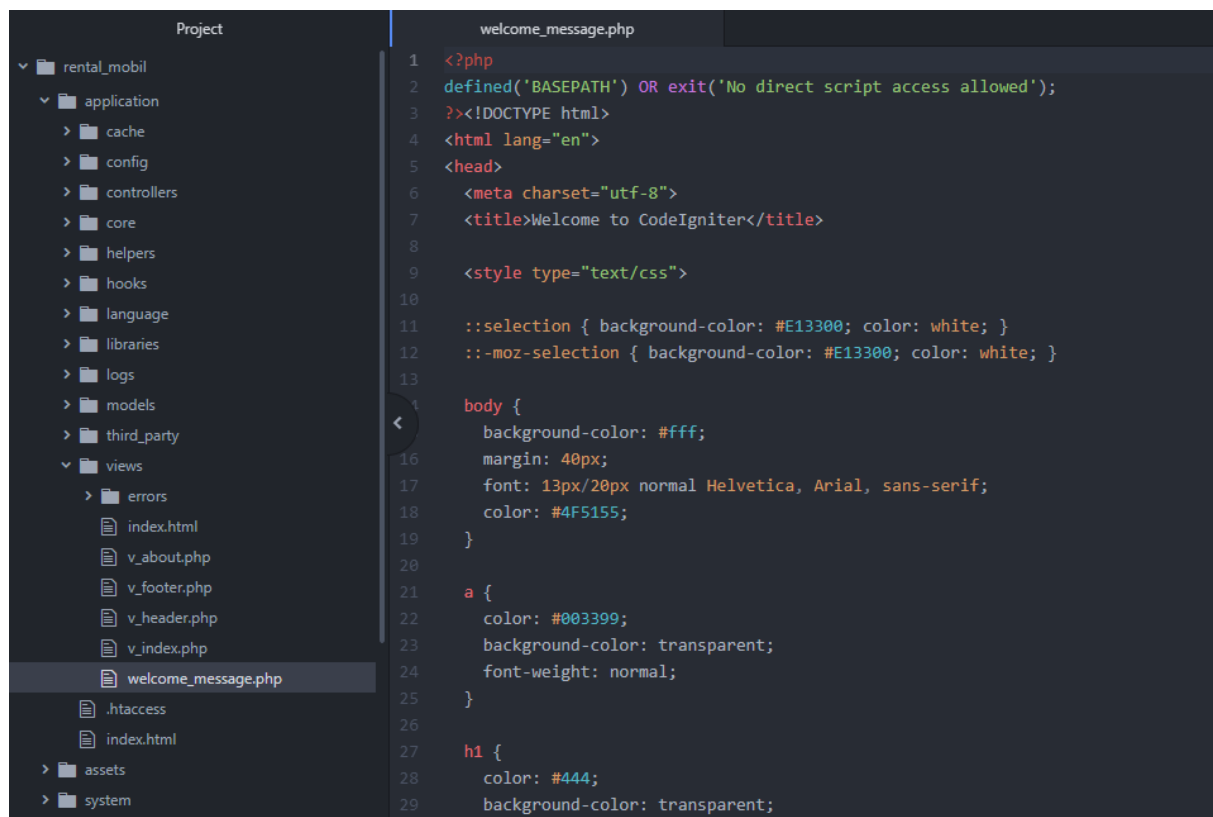
**Controller Welcome** itu sendiri bisa anda temukan pada folder **application/controllers**. Pada file **welcome\_controller** tersebut terdapat **fungsi index** yang gunanya **meload bagian view** atau memanggil halaman views **dengan nama welcome\_messages** sebagai berikut.

```

20     */
21     public function index()
22     {
23         $this->load->view('welcome_message');
24     }
25 }
26

```

Sekarang lihat pada bagian views yang terletak pada **application/views**. Pada folder view tersebut terdapat file dengan nama **welcome\_message**. Pada bagian inilah semua html dan css akan diletakkan yang nantinya akan ditampilkan ke pengunjung.



Contoh menampilkan Biodata

Buatlah sebuah file diberi nama **Contoh1.php** disimpan dalam folder **Application/Controllers**

```

<?php
class Contoh1 extends CI_Controller
{
    public function index()
    {

```

```
    echo "<h1>Perkenalkan</h1>";  
    echo "Nama saya Imam Nawawi  
        Saya tinggal di daerah Ciputat  
        olah raga yang saya sukai adalah  
        Bulutangkis";  
}  
}
```

Untuk melihat hasilnya silahkan ketik URL berikut  
**localhost/rental\_mobil/index.php/contoh1**

## **Tugas 2**

- a. Mahasiswa mengerjakan materi yang ada di pertemuan 3 yaitu mengerjakan contoh1, contoh2, contoh3, dan menghilangkan index.php
- b. Point a dan b akan didemonstrasikan dan dikumpulkan di pertemuan selanjutnya

## PERTEMUAN 3

### MODEL, VIEW, CONTROLLER

Pada bab ini diharapkan mahasiswa dapat memahami secara detail tentang pembuatan dan penggunaan komponen model, view, controller yang merupakan kunci utama dalam menggunakan framework codeigniter.

#### 3.1 Pengertian MVC

Pada pengertian codeigniter di atas tadi di jelaskan bahwa codeigniter menggunakan metode MVC. Apa itu MVC? kita juga harus mengetahui apa itu MVC sebelum masuk dan lebih jauh dalam belajar codeigniter.

MVC adalah teknik atau konsep yang memisahkan komponen utama menjadi tiga komponen yaitu model, view dan controller.

##### a. Model

Model adalah kelas yang merepresentasikan atau memodelkan tipe data yang akan digunakan oleh aplikasi. Model juga dapat didefinisikan sebagai bagian penanganan yang berhubungan dengan pengolahan atau manipulasi database. seperti misalnya mengambil data dari database, menginput dan pengolahan database lainnya. semua intruksi atau fungsi yang berhubungan dengan pengolahan database di letakkan di dalam model. Sebagai contoh, jika ingin membuat aplikasi untuk menghitung luas dan keliling lingkaran, maka dapat memodelkan objek lingkaran sebagai kelas model.

Sebagai catatan, Semua model harus disimpan di dalam folder **application\models**

##### b. View

View merupakan bagian yang menangani halaman user interface atau halaman yang muncul pada user(pada browser). tampilan dari user interface di kumpulan pada view untuk memisahkannya dengan controller dan model sehingga memudahkan web designer dalam melakukan pengembangan tampilan halaman website.

### c. Controller

Controller merupakan kumpulan intruksi aksi yang menghubungkan model dan view, jadi user tidak akan berhubungan dengan model secara langsung, intinya data yang tersimpan di database (model) di ambil oleh controller dan kemudian controller pula yang menampilkan nya ke view. Jadi controller lah yang mengolah intruksi.

Dari penjelasan tentang model view dan controller di atas dapat di simpulkan bahwa controller sebagai penghubung view dan model. misalnya pada aplikasi yang menampilkan data dengan menggunakan metode konsep mvc, controller memanggil intruksi pada model yang mengambil data pada database, kemudian controller yang meneruskannya pada view untuk di tampilkan. jadi jelas sudah dan sangat mudah dalam pengembangan aplikasi dengan cara mvc ini karena web designer atau front-end developer tidak perlu lagi berhubungan dengan controller, dia hanya perlu berhubungan dengan view untuk mendesign tampilann aplikasi, karena back-end developer yang menangani bagian controller dan modelnya. Jadi pembagian tugas pun menjadi mudah dan pengembangan aplikasi dapat di lakukan dengan cepat dan terstruktur.

Bentuk umum model seperti berikut:

```
Class Nama_model extends CI_model {  
    // badan kelas  
}
```

jika ingin memanggil konstruktor kelas CI\_model dari konstruktor kelas model yang didefinisikan, maka bentuk umumnya seperti berikut:

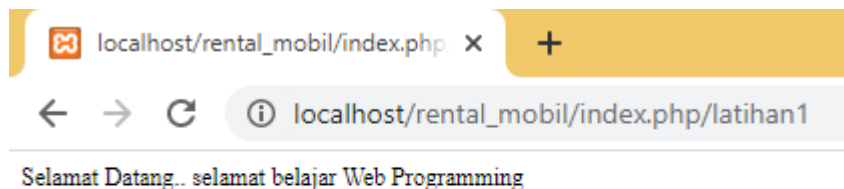
```
class Nama_model extends CI_model {  
    //konstruktor kelas model  
    Function __construct( ) {  
        //memanggil konstruktor kelas CI_model  
        $parent: : __construct();  
        //...  
    }  
    // ...  
}
```

## Contoh1 penggunaan hanya controller

**Controller Latihan1.php** simpan dalam folder **Application/controllers/**

```
<?php
class Latihan1 extends CI_Controller
{
    public function index()
    {
        echo "Selamat Datang.. selamat belajar Web Programming";
    }
}
```

Untuk melihat hasilnya ketikkan url **localhost/rental\_mobil/latihan1**



## Contoh2 menggunakan Controller dan Model

Model **Model\_latihan1.php** simpan di **application\models\**

```
<?php
class Model_latihan1 extends CI_Model
{
    //membuat variable untuk menampung nilai
    public $nilai1, $nilai2, $hasil;

    //method penjumlahan
    public function jumlah($nil1 = null, $nil2 = null)
    {
        $this->nilai1 = $nil1;
        $this->nilai2 = $nil2;
        $this->hasil = $this->nilai1 + $this->nilai2;
    }
}
```



```

        return $this->hasil;
    }
}

```

Ubah controller Latihan1 yang sebelumnya sudah dibuat sehingga menjadi seperti berikut:

```

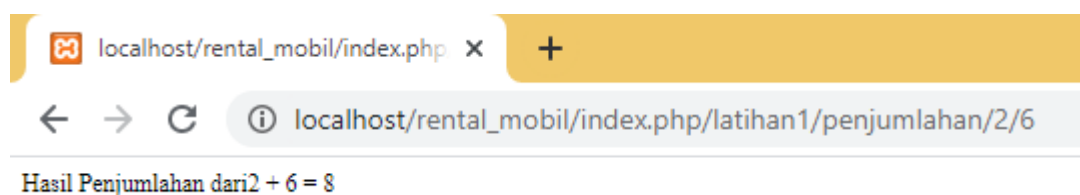
<?php
class Latihan1 extends CI_Controller
{
    public function index()
    {
        echo "Selamat Datang.. selamat belajar Web Programming";
        //$this->load->view('view-latihan1');
    }

    public function penjumlahan($n1, $n2)
    {
        $this->load->model('Model_latihan1');
        $hasil = $this->Model_latihan1->jumlah($n1, $n2);
        echo "Hasil Penjumlahan dari". $n1 ." + ". $n2 ." = "
        . $hasil;
    }
}

```

Untuk melihat hasilnya ketikkan url

**localhost/rental\_mobil/index.php/latihan1/penjumlahan/2/6**



### Contoh3 menggunakan Controller dan View dan Model

View **view-latihan.php** Simpan di **application\views\**

```

<html>

<head>
    <title>Latihan 1</title>
</head>

```

```

<body>
    Halo Kawan.. Yuk kita belajar web programming..!!!<br>
    Nilai 1 = <?= $nilai1; ?>
    Nilai 2 = <?= $nilai2; ?>
    ini hasil dari pemodelan dengan methode penjumlahan yaitu <?=
    $nilai1 . " + " . $nilai2 . " = " . $hasil; ?>

</body>

</html>

```

Ubah controller Latihan1 yang sebelumnya sudah dibuat sehingga menjadi seperti berikut:

```

<?php
class Latihan1 extends CI_Controller
{
    public function index()
    {
        echo "Selamat Datang.. selamat belajar Web Programming";
        //$this->load->view('view-latihan1');
    }

    public function penjumlahan($n1, $n2)
    {
        $this->load->model('Model_latihan1');

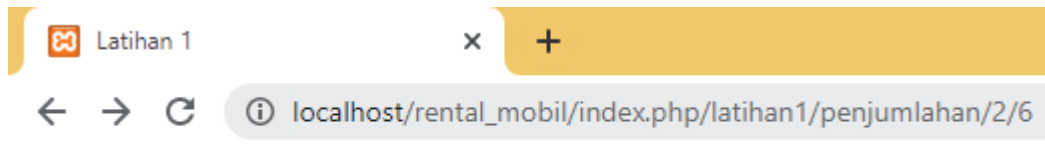
        $data['nilai1'] = $n1;
        $data['nilai2'] = $n2;
        $data['hasil'] = $this->Model_latihan1->jumlah($n1, $n2);

        $this->load->view('view-latihan1', $data);
    }
}

```

Untuk melihat hasilnya ketikkan url

**localhost/rental\_mobil/index.php/latihan1/penjumlahan/2/6**



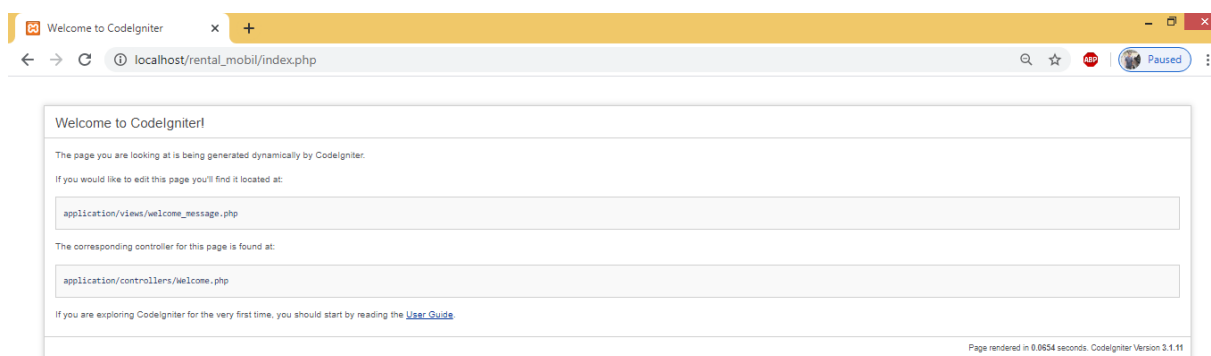
### 3.2 Menentukan Kontroler Standar (Default)

Controller standar (default) adalah controller yang akan dipanggil secara otomatis ketika user tidak menyertakan nama controller di dalam URI.

Contoh penulisan URI:

`http://localhost/rental_mobil/index.php`

pada contoh permintaan di atas, kita tidak menyertakan nama controller yang akan dipanggil (segmen pertama). Permintaan tersebut akan memanggil controller standar yang sudah didefinisikan oleh codeigniter, dan akan memberikan hasil seperti berikut:



Controller standar dapat ditentukan sendiri sesuai dengan keinginan, yaitu dengan cara melakukan konfigurasi pada file **routes.php** yang terletak pada folder **application\config\**. cari baris code berikut:

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

Ubah kata welcome dengan nama controller yang diinginkan untuk dijadikan controller standar

### 3.3 Menghilangkan index.php pada codeigniter

Dalam rangka pengamanan web site, biasanya seorang web developer akan melakukan berbagai trik untuk dapat mengamankan web miliknya. Ada yang dibuatkan sistem sandi yang diubah secara berkala, ada yang menggunakan sistem sandi dengan teknik kriptografi yaitu mengenkrip sandi sedemikian rupa agar tidak mudah dipecahkan oleh pihak yang tidak bertanggung jawab, ada juga yang menggunakan trik kamuplase pengaksesan file, dan lain lain. Berikut ini adalah salah satu trik untuk mengelabui pihak yang tidak bertanggung jawab agar tidak mudah untuk melakukan pencurian atau penggunaan data yang merugikan yaitu dengan cara menghilangkan index.php pada url codeigniter. cara nya buka **config.php** yang terletak pada folder **application/config/**, kemudian cari pengaturan **\$config['index\_page']** seperti di bawah ini.

```
$config['index_page'] = 'index.php';
```

Kemudian hapus index.php pada pengaturan sehingga menjadi

```
$config['index_page'] = '';
```

Setelah menghapus index, selanjutnya adalah membuat file **.htaccess** pada direktori root codeigniter. Caranya buat file baru dengan nama **.htaccess** tanpa diberi extensi, karena .htaccess merupakan file yang tidak berekstensi. Diawali dengan tanda titik ( . ).

File **rental\_mobil/.htaccess**

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php/$1 [L]
```

Simpan dan sekarang coba akses di web. Disini untuk contoh dalam rangka mencoba apakah sudah berhasil atau belum menghilangkan index pada codeigniter, kita akan coba mengakses method penjumlahan yang sudah ada pada controller **Latihan1.php** tanpa menyertakan kata index.php pada URL.

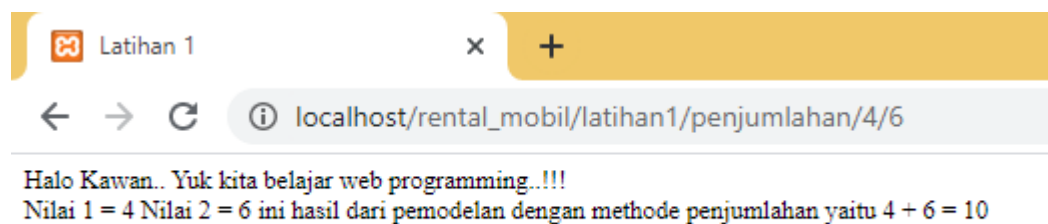
Jika biasanya untuk mengakses method dalam kontroler Latihan1 diatas harus dengan mengetikkan

`http://localhost/rental_mobil/index.php/latihan1/penjumlahan/4/6`

maka sekarang tidak perlu, melainkan bisa langsung mengakses alamat url di atas tanpa menyertakan kata index.php

`http://localhost/rental_mobil/latihan1/penjumlahan/4/6`

berikut hasil dalam browser



### 3.4 Aturan merancang Model, View, dan Controller

#### a. Model

Secara umum perancangan model dapat dibuat menggunakan aturan berikut:

- 1) Model harus memiliki atribut yang dapat mewakili element tertentu.
- 2) Model seharusnya berisi kode kode yang menangani proses bisnis untuk data bersangkutan, misalnya untuk melakukan proses validasi data.
- 3) Model boleh berisi kode kode yang digunakan untuk memanipulasi data, misalnya untuk menambah baris data baru, mengubah baris data, dan menghapus baris data pada sebuah tabel yang terdapat dalam database.
- 4) Model seharusnya tidak berisi kode kode yang berkaitan langsung dengan permintaan yang dilakukan oleh user melalui URL (tidak berisi `$_GET` maupun `$_POST`). Pekerjaan seperti ini seharusnya dilakukan oleh controller.
- 5) Model seharusnya tidak berisi kode kode yang berkaitan dengan presentasi (tampilan) halaman web yang akan disajikan ke hadapan user. Pekerjaan semacam ini seharusnya dilakukan oleh view.

#### b. View

Berikut ini aturan yang dapat digunakan untuk merancang view.

- 1) View harus berisi kode kode yang bersifat presentasional, biasanya berupa kode HTML. View juga dapat berisi perintah perintah PHP sederhana yang masih berkaitan dengan proses pembuatan tampilna, misalnya untuk menampilkan data yang dihasilkan oleh model tertentu.
- 2) View seharusnya tidak berisi kode PHP yang melakukan akses data secara langsung ke database. Pekerjaan semacam ini sebaiknya dilakukan oleh model.
- 3) View seharusnya menghindari kode PHP yang mengakses secara langsung permintaan dari user (tidak berisi \$\_GET maupun \$\_POST). Tugas ini sebaiknya didelegasikan ke controller

c. Controller

Berikut ini aturan yang dapat digunakan untuk merancang controller.

- 1) Controller boleh mengakses \$\_GET, \$\_POST, dan variabel variabel PHP lainnya yang berkaitan dengan permintaan user.
- 2) Controller boleh membuat objek (instance) dari kelas kelas model dan mengarahkan ke view (jika model bersangkutan menghasilkan data yang perlu ditampilkan ke user)
- 3) Controller seharusnya tidak berisi kode kode yang mengakses data secara langsung dari database. Tugas seperti ini sebaiknya dilakukan oleh model. Dan controller tinggal menggunakannya saja.
- 4) Controller seharusnya tidak berisi kode kode HTML yang diperlukan untuk tujuan presentasi. Tugas ini sebaiknya dilakukan oleh view.

### **3.5 Parsing Data (mengirimkan nilai) dari controller Ke View**

Untuk mengirimkan nilai dari controller ke view, anda dapat mengirimkannya dengan menggunakan bantuan array. Jadi data yang akan diparsing kita masukkan ke array.

Perhatikan contoh cara pengiriman nilai ke view codeigniter berikut ini.

**Contoh5** mengirimkan nilai dari controller ke view

Buatlah sebuah view baru beri nama **view-form-matakuliah.php** kemudian simpan dalam folder **Application/views/** dan ketik script berikut:

```
<html>

<head>
    <title>Form Input Matakuliah</title>
</head>

<body>
    <center>
        <form action="<?=php echo base_url('matakuliah/cetak'); ?"
method="post">
            <table>
                <tr>
                    <th colspan="3">
                        Form Input Data Mata Kuliah
                    </th>
                </tr>
                <tr>
                    <td colspan="3">
                        <hr>
                    </td>
                </tr>
                <tr>
                    <th>Kode MTK</th>
                    <th>:</th>
                    <td>
                        <input type="text" name="kode" id="kode">
                    </td>
                </tr>
                <tr>
                    <th>Nama MTK</th>
                    <td>:</td>
                    <td>
                        <input type="text" name="nama" id="nama">
                    </td>
                </tr>
                <tr>
                    <th>SKS</th>
                    <td>:</td>
```



```

        <td>
            <select name="sks" id="sks">
                <option value="">Pilih SKS</option>
                <option value="2">2</option>
                <option value="3">3</option>
                <option value="4">4</option>
            </select>
        </td>
    </tr>
    <tr>
        <td colspan="3" align="center">
            <input type="submit" value="Submit">
        </td>
    </tr>
</table>
</form>
</center>
</body>

</html>

```

Script di atas akan membentuk sebuah tampilan form input data yang akan di submit kemudian di proses ke sebuah controller. Untuk kita perlu membuat controller terlebih dahulu beri nama **Matakuliah.php** kemudian simpan dalam folder **Application/controllers/** lalu ketiklah script berikut ini:

```

<?php
class Matakuliah extends CI_Controller
{

    public function index()

    {

        $this->load->view('view-form-matakuliah');
    }

    public function cetak()
    {
        $data = [
            'kode' => $this->input->post('kode'),
            'nama' => $this->input->post('nama'),

```

```

        'sks' => $this->input->post('sks')
    ];

    $this->load->view('view-data-matakuliah', $data);
}
}

```

Dalam controller di atas, data yang diinput pada form input matakuliah di post kan dan di masukan ke dalam variabel yang bertipe array yang diberi nama **\$data** dan variable array ini memiliki 3 komponen array yaitu **kode**, **nama**, **sks**. Kemudian nilai dari element-element array tadi akan dikirimkan kembali ke view dengan perintah **\$this->load->view('view-data-matakuliah', \$data)**. jadi kita perlu membuat sebuah view baru lagi beri nama **view-data-matakuliah.php** kemudian simpan dalam folder **Application/views/** dan ketiklah script ini:

```

<html>

<head>
    <title>Tampil Data Matakuliah</title>
</head>

<body>
    <center>
        <table>
            <tr>
                <th colspan="3">
                    Tampil Data Mata Kuliah
                </th>
            </tr>
            <tr>
                <td colspan="3">
                    <hr>
                </td>
            </tr>
            <tr>
                <th>Kode MTK</th>
                <th>:</th>
                <td>
                    <?= $kode; ?>
                </td>
            </tr>
            <tr>

```

```

        <td>Nama MTK</td>
        <td>:</td>
        <td>
            <?= $nama; ?>
        </td>
    </tr>
    <tr>
        <td>SKS</td>
        <td>:</td>
        <td>
            <?= $sks; ?>
        </td>
    </tr>
    <tr>
        <td colspan="3" align="center">
            <a href="<?= base_url('matakuliah');
?>">Kembali</a>
        </td>
    </tr>
</table>
</center>
</body>
</html>

```

### Tugas 3

- Mahasiswa mengerjakan materi yang ada di pertemuan 4 yaitu membuat template sederhana dan dinamis dan setting base\_url()
- Point a dan b akan didemonstrasikan dan dikumpulkan di pertemuan selanjutnya

## PERTEMUAN 4

### MEMBUAT TEMPLATE (TEMPLATING) WEB SEDERHANA

Pada pertemuan ini, mahasiswa diharapkan mampu membuat template sederhana. Mahasiswa juga diharapkan mampu membuat template web dinamis.

Pada codeigniter, halaman yang dipecah-pecah dapat dengan mudah untuk me-load halaman-halaman tersebut sesuai dengan keinginan.

#### 4.1 Membuat Template Sederhana yang dinamis

Membuat dan menampilkan template sederhana, dibutuhkan pembuatan controller dan view agar hasilnya bisa dilihat.

Buatlah sebuah kontroler yang akan digunakan untuk menampilkan view. Sebagai contoh, buatlah kontroler Web.php lalu simpan di **application/controllers/Web.php**.

```
<?php
defined('BASEPATH') or exit ('no direct script access allowed');

class Web extends CI_Controller{

    function __construct(){
        parent::__construct();
    }

    public function index(){
        $data['judul'] = "Halaman Depan";
        $this->load->view('v_header',$data);
        $this->load->view('v_index',$data);
        $this->load->view('v_footer',$data);
    }

}
```

Kemudian buatlah 3 buah file view beri nama **v\_header.php**, **v\_index.php**, dan **v\_footer.php** lalu simpan dalam folder **application/views/** kemudian ketik script berikut:

## v\_header.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Web Prog II | Merancang Template sederhana dengan
codeigniter</title>
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url() ?>assets/css/stylemobil.css">
</head>
<body>
    <div id="wrapper">
        <header>
            <hgroup>
                <h1>Rental Mobil</h1>
                <h3>Membuat Template Sederhana dengan
CodeIgniter</h3>
            </hgroup>
            <nav>
                <ul>
                    <li><a href="<?php echo
base_url().'index.php/web' ?>">Home</a></li>
                    <li><a href="<?php echo
base_url().'index.php/web/about' ?>">About</a></li>
                </ul>
            </nav>
            <div class="clear"></div>
        </header>
```

## v\_index.php

```
<section>
    <h1><?php echo $judul ?></h1>
    <p align='justify'>Pada pengertian codeigniter di atas
tadi di jelaskan bahwa codeigniter menggunakan metode MVC. Apa itu
MVC? Kita juga harus mengetahui apa itu MVC sebelum masuk dan lebih
jauh dalam belajar codeigniter.</p>
    <p>MVC adalah teknik atau konsep yang memisahkan
komponen utama menjadi tiga komponen yaitu model, view dan
controller.</p>

    <ol type="a">
```

```
<li>Model</li>
<p align='justify'>Model adalah kelas yang merepresentasikan atau memodelkan tipe data yang akan digunakan oleh aplikasi. Model juga dapat didefinisikan sebagai bagian penanganan yang berhubungan dengan pengolahan atau manipulasi database. Seperti misalnya mengambil data dari database, menginput dan pengolahan database lainnya. Semua intruksi atau fungsi yang berhubungan dengan pengolahan database diletakkan di dalam model. Sebagai contoh, jika ingin membuat aplikasi untuk menghitung luas dan keliling lingkaran, maka dapat memodelkan objek lingkaran sebagai kelas model.</p>
<p align='justify'>Sebagai catatan, Semua model harus disimpan di dalam folder application\models</p>
<li>View</li>
<p align='justify'>View merupakan bagian yang menangani halaman user interface atau halaman yang muncul pada user(pada browser). Tampilan dari user interface di kumpulkan pada view untuk memisahkannya dengan controller dan model sehingga memudahkan web designer dalam melakukan pengembangan tampilan halaman website.</p>
<li>Controller</li>
<p align='justify'>Controller merupakan kumpulan intruksi aksi yang menghubungkan model dan view, jadi user tidak akan berhubungan dengan model secara langsung, intinya data yang tersimpan di database (model) di ambil oleh controller dan kemudian controller pula yang menampilkan nya ke view. Jadi controller lah yang mengolah intruksi.</p>
<p align='justify'>Dari penjelasan tentang model view dan controller di atas dapat di simpulkan bahwa controller sebagai penghubung view dan model. Misalnya pada aplikasi yang menampilkan data dengan menggunakan metode konsep mvc, controller memanggil intruksi pada model yang mengambil data pada database, kemudian controller yang meneruskannya pada view untuk di tampilkan. Jadi jelas sudah dan sangat mudah dalam pengembangan aplikasi dengan cara mvc ini karena web designer atau front-end developer tidak perlu lagi berhubungan dengan controller, dia hanya perlu berhubungan dengan view untuk mendesign tampilann aplikasi, karena back-end developer yang menangani bagian controller dan modelnya. Jadi pembagian tugas pun menjadi mudah dan pengembangan aplikasi dapat di lakukan dengan cepat dan terstruktur.</p>
</section>
```

## V\_footer.php

```
<footer>
    <a href="http://www.RentalMobil.com">RentalMobil</a>
</footer>
</div>
</body>
</html>
```

Kemudian buatlah **stylemobil.css** untuk membuat tampilan lebih bagus dan simpan di dalam folder root seperti berikut: **rental\_mobil/assets/css/**

```
body{
    background: #eee;
    color: #333;
    font-family: sans-serif;
    font-size:15px;
}

#wrapper{
    background: #fff;
    width: 1100px;
    margin: 20px auto;
}

#wrapper header{
    background: #232323;
    padding: 20px;
}

#wrapper header hgroup{
    float: left;
    color: #fff;
}

#wrapper header nav{
    float: right;
    margin-top: 50px;
}

#wrapper header nav ul{
    padding: 0;
```



```

    margin: 0;
}

#wrapper header nav ul li{
    float: left;
    list-style: none;
}

#wrapper header nav ul li a{
    padding: 15px;
    color: #fff;
    text-decoration: none;
}

.clear{
    clear: both;
}

footer{
    background: #232323;
    padding: 20px;
}

footer a{
    color: #fff;
    text-decoration: none;
}

section{
    padding: 20px;
}

```

Setelah membuat file-file di atas. sebelum di jalankan melalui browser, terlebih dahulu perlu dilakukan seting `base_url()` untuk memudahkan dalam menghubungkan file view dengan file css nya.

## 4.2 Seting `base_url()` pada codeigniter

Untuk melakukan seting `base_url()`, dapat dilakukan melalui file **config.php** yang ada di dalam **application/config/config.php**. buka file tersebut lalu cari baris sintak seperti berikut:

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

Kemudian diubah menjadi

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

pengaturan di atas mengikuti alamat path proyek masing-masing. Kemudian dapat diperhatikan pada contoh file **v\_index.php** di bagian <head> bahwa untuk menghubungkan dengan file css, dibutuhkan bantuan `base_url()`. seperti berikut:

```
<link rel="stylesheet" type="text/css" href="<?php echo base_url()  
?>assets/css/stylemobil.css">
```

Hasil perintah di atas akan sama seperti ini

```
<link rel="stylesheet" type="text/css"  
href="http://localhost/rental_mobil/assets/css/stylemobil.css">
```

Hal tersebut di atas dikarenakan `base_url()` sudah terlebih dahulu di seting menjadi “localhost/rental\_mobil”. Ketika `base_url()` sudah diseting atau diubah dari bentuk default menjadi seperti di atas, maka perlu juga dilakukan pengaktifan helper url. Hal ini dapat dilakukan dengan cara menambahkan script load helper pada controller **Web.php** yang sudah dibuat pada bagian **construct**.

```
<?php  
defined('BASEPATH') or exit ('no direct script access allowed');  
  
class Web extends CI_Controller{  
  
    function __construct(){  
        parent::__construct();  
        $this->load->helper('url');  
    }  
  
    public function index(){  
        $data['judul'] = "Halaman Depan";  
        $this->load->view('v_header',$data);  
        $this->load->view('v_index',$data);  
        $this->load->view('v_footer',$data);  
    }  
}
```

Atau bisa juga melalui file **autoload.php** yang ada di dalam folder **application/config/** kemudian cari baris sintak seperti di bawah

```
$autoload['helper'] = array();
```

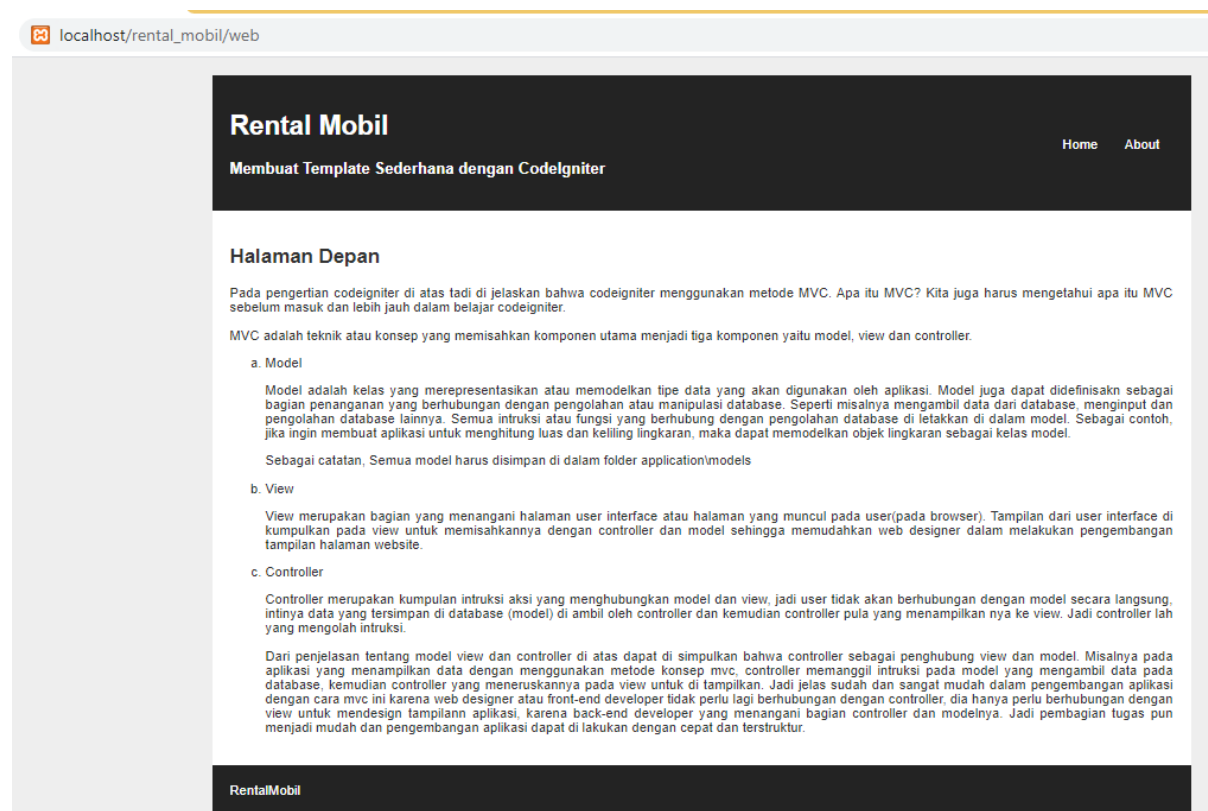
Kemudian diubah dengan ditambahkan kata 'url' menjadi seperti di bawah

```
$autoload['helper'] = array('url');
```

Setelah selesai, bisa dilihat hasilnya dengan menjalankan kontroler **Web.php**

[http://localhost/rental\\_mobil/web](http://localhost/rental_mobil/web)

hasilnya seperti gambar di bawah.



Demikian tampilan web sederhana sudah berhasil dibuat pada codeigniter. Template yang dibuat di atas merupakan template dinamis. Maksudnya tampilan header dan footer tidak akan berubah namun bagian konten bisa berubah-ubah.

Contoh untuk membuat halaman lainnya yaitu akan dibuat halaman **v\_about.php**. Yang mana link untuk menuju halaman about sudah dibuat sebelumnya pada file **v\_header.php**. Link yang sudah dibuat yaitu **Home** diseting untuk menuju atau mengakses method `index()` dan **About** diseting menuju atau mengakses method `about()`

```
<li><a href="<?php echo base_url(). 'index.php/web' ?>">Home</a></li>
    <li><a href="<?php echo
base_url(). 'index.php/web/about' ?>">About</a></li>
```

Sekarang buatlah file view lagi dan beri nama **v\_about.php** dan simpan di **application/views/v\_about.php**

```
<section>
    <h1><?php echo $judul ?></h1>
    <h4>Nama</h4>
    <ul type="disc">
        <li>Nama Depan : Imam</li>
        <li>Nama Belakang : Nawawi</li>
    </ul>
    <br>
    <h4>Alamat</h4>
    <ul type="none">
        <li> Jalan Ciledug Raya No. 168 Pesanggrahan</li>
    </ul>

    <h4>Tempat Lahir</h4>
    <ul type="none">
        <li>Cirebon</li>
    </ul>

    <h4>Olah Raga Favorit</h4>
    <ul type="square">
        <li>Bulutangkis</li>
        <li>Catur</li>
    </ul>
</section>
```

Kemudian tambahkan method baru yaitu method **about()** pada controller **Web.php** di bawah method **index()**

```
public function about()
{
```

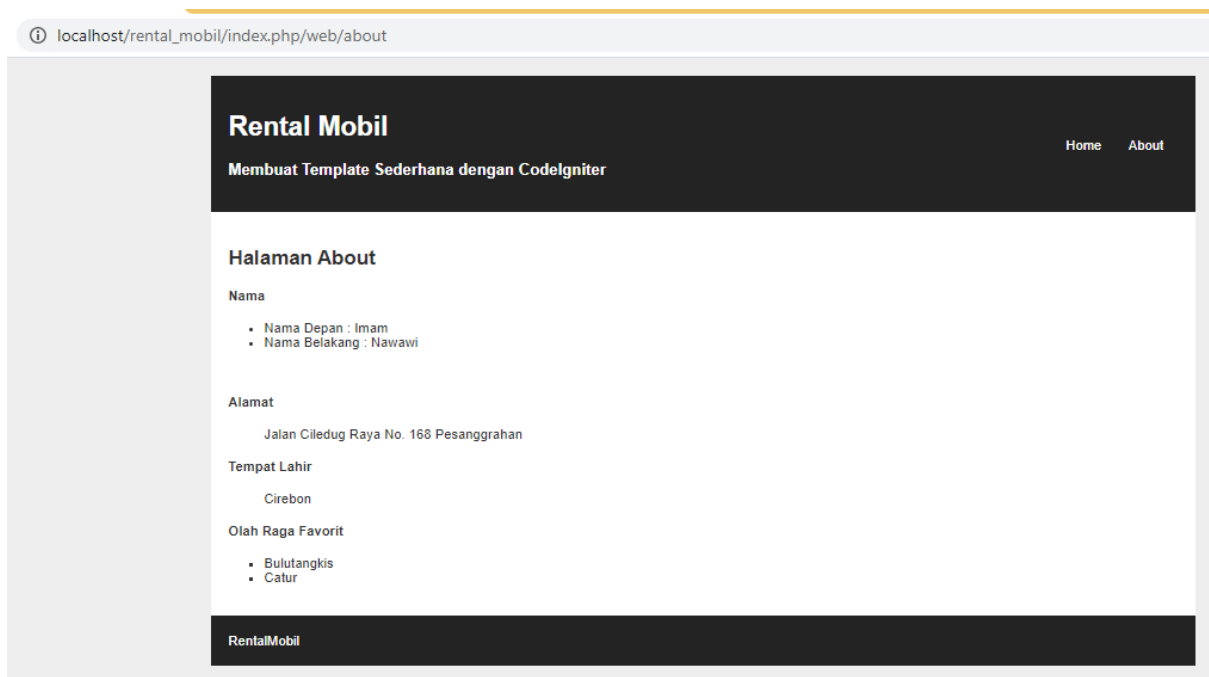
```

    $data['judul'] = "Halaman About";
    $this->load->view('v_header', $data);
    $this->load->view('v_about', $data);
    $this->load->view('v_footer', $data);
}

```

Selanjutnya jalankan kembali pada browser dan klik pada menu About atau bisa langsung mengakses ke method about.

[http://localhost/rental\\_mobil/web/about](http://localhost/rental_mobil/web/about)



#### Tugas 4

- c. Mahasiswa mengerjakan materi yang ada di pertemuan 5 menambahkan validasi data dengan form validasi pada controller matakuliah
- d. Point a dan b akan didemonstrasikan dan dikumpulkan di pertemuan selanjutnya

## PERTEMUAN 5

### MEMBUAT FORM VALIDASI PADA CODEIGNITER

#### 2.1 Form Validasi

Pada pertemuan ini mahasiswa diharapkan dapat memahami tentang validasi data dan bagaimana membuat form validasi data pada codeigniter dalam penginputan data. Seperti misalnya dalam menentukan form, apa saja yang wajib untuk diisi, form dengan format input tertentu, membuat konfirmasi password dan lainnya.

Pembuatan form validasi pada codeigniter diperlukan adanya pemanggilan library **form\_validation** pada codeigniter.

Pada saat membuat dan menangani form, proses validasi data merupakan bagian yang sangat penting untuk diperhatikan. Validasi data diperlukan untuk menjaga keabsahan dari data yang dikirim oleh user ke dalam aplikasi

#### 2.2 Mekanisme Validasi Data di dalam Form

- a. Form ditampilkan ke halaman web
- b. User mengisi data ke dalam form tersebut
- c. Jika terdapat satu atau beberapa data yang salah, maka form akan ditampilkan ulang, biasanya disertai dengan pesan kesalahan.
- d. Proses ini akan diulang sampai semua data yang diminta di dalam form diisi dengan benar.

Proses pemeriksaan data yang dikirim oleh user di dalam form dilakukan di dalam kontroler, demikian juga dengan pembuatan aturan (rule) dari datanya.

#### 2.3 Memuat Library Form\_validation

Untuk mengimplementasikan proses validasi data, yang diperlukan adalah memuat librari dari form\_validation terlebih dahulu. Proses pemuatan librari tersebut dapat dilakukan di bagian konstruktor controller.

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

Atau bisa juga dilakukan melalui file autoload yang ada dalam folder **application/config/autoload.php**

Buka file tersebut lalu cari baris sintak seperti berikut:

```
$autoload['libraries'] = array();
```

Kemudian diubah dengan ditambahkan kata `form_validation` seperti berikut:

```
$autoload['libraries'] = array('form_validation');
```

Contoh pembuatan form validasi kali ini akan menggunakan controller **Matakuliah.php** yang sudah dibuat pada pertemuan sebelumnya. Buka controller **Matakuliah.php** kemudian ubah menjadi seperti di bawah:

```
<?php
class Matakuliah extends CI_Controller
{

    public function index()
    {

        $this->load->view('view-form-matakuliah');
    }

    public function cetak()
    {
        $this->form_validation->set_rules('kode', 'Kode Matakuliah',
        'required|min_length[3]', [
            'required' => 'Kode Matakuliah Harus diisi',
            'min_lenght' => 'Kode terlalu pendek'
        ]);

        $this->form_validation->set_rules('nama', 'Nama Matakuliah',
        'required|min_length[3]', [
            'required' => 'Nama Matakuliah Harus diisi',
            'min_lenght' => 'Nama terlalu pendek'
        ]);

        if ($this->form_validation->run() != true) {
            $this->load->view('view-form-matakuliah');
        } else {
```



```

        $data = [
            'kode' => $this->input->post('kode'),
            'nama' => $this->input->post('nama'),
            'sks' => $this->input->post('sks')
        ];

        $this->load->view('view-data-matakuliah', $data);
    }
}
}

```

Pada controller Matakuliah di atas terlihat membuat rules dari form validasi. Rule nya berupa **required** dan **min\_length[3]** yang artinya jika required maka harus diisi. Sedangkan min\_length[3] maka minimal diisi 3 digit.

untuk selengkapnya tentang membuat form validation pada codeigniter anda dapat membacanya pada user\_guide yang sudah di sediakan oleh codeigniter pada project CI anda.

[http://localhost/codeigniter/user\\_guide/libraries/form\\_validation.html#the-form](http://localhost/codeigniter/user_guide/libraries/form_validation.html#the-form)

### Tugas 5

- a. Mahasiswa mengerjakan materi yang ada di pertemuan 6 yaitu setting base\_url, membuat database, membuat model user dan model mobil
- b. Point a dan b akan didemonstrasikan dan dikumpulkan di pertemuan selanjutnya

## **PERTEMUAN 6**

### **STUDI KASUS RENTAL MOBIL**

Pada pertemuan ini mahasiswa diharapkan dapat memahami persiapan dasar untuk memulai sebuah projek kasus. Mahasiswa juga mampu memahami desain dan pembuatan database

Pada tahap ini akan dimulai membuat sebuah project aplikasi berbasis web yang akan dibuat dengan menggunakan CodeIgniter. untuk memulai pembuatan aplikasi rental\_mobil berbasis web.

#### **6.1 Persiapan Membuat Aplikasi rental mobil**

Rental Mobil merupakan aplikasi reservasi atau booking peminjaman mobil pada sebuah rental mobil yang dilakukan secara online dengan melibatkan anggota, admin dan aplikasi rental\_mobil. Aplikasi rental\_mobil ini dibagi menjadi 2 jenis tampilan yaitu tampilan bagian backend dan tampilan frontend. Untuk pembahasan Web Programming 2 ini hanya membahas tampilan backend.

Aplikasi rental mobil adalah aplikasi berbasis web yang dibuat untuk booking mobil untuk dipinjam. Tujuan dibuatkannya aplikasi ini agar memudahkan para pengguna dalam mencari mobil dan ingin meminjam mobil tersebut. Jadi sebelum di datang ke rental dan meminjam mobil. Pengguna bisa mencari terlebih dahulu mobil yang akan dipinjam, untuk memastikan bahwa di rental yang akan dia kunjungi benar bahwa mobil tersebut tersedia.

Alur logika sistem rental mobil ini yaitu seseorang yang ingin melakukan booking diharuskan mendaftar menjadi anggota terlebih dahulu, selanjutnya ketika sudah menjadi anggota, baru dapat melakukan booking terhadap mobil yang akan dipinjam. Setelah melakukan booking, anggota diharuskan mengambil mobil yang telah diboeking dengan cara datang langsung ke rental dalam waktu 1x24 jam. Kemudian konfirmasi ke petugas atau admin untuk diambikan mobil yang telah diboeking berdasarkan bukti booking melalui aplikasi.

Rental\_mobil secara utuh memiliki kebutuhan sebagai berikut:

#### Kebutuhan user:

1. Admin
  - a. Seorang admin dapat login ke dalam aplikasi rental\_mobil
  - b. Seorang admin dapat menambah, melihat, mengubah, dan menghapus, data mobil pada aplikasi rental\_mobil.
  - c. Seorang admin dapat melihat data booking dan melanjutkan proses booking sampai mobil dipinjam.
  - d. Seorang admin dapat memproses pengembalian mobil yang sudah selesai dipinjam.
2. Pengunjung
  - a. Pengunjung dapat melihat-lihat data mobil yang ada pada aplikasi rental\_mobil
  - b. Pengunjung dapat melakukan registrasi untuk menjadi anggota rental\_mobil
  - c. Pengunjung dapat memberikan komentar melalui buku tamu
3. Anggota
  - a. Anggota dapat login ke dalam sistem aplikasi rental\_mobil.
  - b. Anggota dapat melakukan booking terhadap mobil yang ada pada aplikasi rental\_mobil
  - c. Anggota dapat mencetak bukti booking mobil untuk dibawa ketika akan mengambil mobil yang diboeking.

#### Kebutuhan sistem:

- a. Sistem akan melakukan validasi username dan password ketika ada seorang user yang melakukan login
- b. Sistem juga melakukan validasi data mobil dan anggota yang akan diinput ke dalam database
- c. Sistem akan memblokir dan memberikan notif apabila ada seorang pengunjung yang meng-klik tombol booking tanpa melakukan login terlebih dahulu.
- d. Sistem akan menghapus secara otomatis data booking yang sudah lewat dari 1 x 24 jam
- e. Sistem dapat mengkalkulasi denda yang harus dibayarkan ketika ada anggota yang terlambat mengembalikan mobil.
- f. Sistem dapat menampilkan mobil yang dicari oleh user berdasarkan kriteria pencarian.

Persiapan untuk memulai membangun aplikasi rental\_mobil terdiri dari instalasi codeigniter, pemilihan tool pendukung, setting base\_url, load librari, helper, dan segala yang dibutuhkan. Tetapi disini tidak akan membahas instalasi codeigniter karena sudah sangat mudah dijelaskan di pertemuan awal. Disini hanya kan membahas mulai dari seting base\_url dan seterusnya.

Diperlukan beberapa penyesuaian atau pengaturan pada codeigniter sebelum mulai digunakan untuk membuat aplikasi rental\_mobil. Semua pengaturan atau penyesuaian yang dilakukan di sini sesuai dengan kebutuhan untuk membuat aplikasi rental\_mobil dengan codeigniter. diantaranya adalah:

## 6.2 Setting Base URL

Pengaturan base url adalah pengaturan URL dasar dalam sebuah project aplikasi yang dibuat dengan menggunakan codeigniter. Pengaturan base url di codeigniter terdapat di dalam file config.php di dalam folder project codeigniter.

### **rental\_mobil/application/config/config.php**

Helper dan Library bisa dibilang seperti coding dengan fungsi-fungsi yang berbeda dan memiliki kegunaan masing-masing. Tinggal memanggil dan menggunakannya saja. Kita juga bisa membuat helper atau library kita sendiri di codeigniter untuk memudahkan pekerjaan kita.

Di tahap ini kita akan men-load atau membuka beberapa library dan helper codeigniter yang kita butuhkan dalam pembuatan aplikasi rental\_mobil ini. Untuk menjalankan atau membuka library dan helper secara otomatis di codeiginter, kita bisa mengaturnya dalam file autoload.php dalam folder *application/config/autoload.php*.

### **rental\_mobil/application/config/autoload.php**

Library yang akan kita load secara otomatis adalah *database*, *session* dan *form\_validation*.

Cari sintak

```
$autoload['libraries'] = array();
```

Kemudian tambahkan database, session dan form\_validation. Sehingga menjadi seperti berikut:

```
$autoload['libraries'] = array('form_validation', 'session',  
'database');
```

Dan helper yang akan kita load secara otomatis adalah *url*:

```
$autoload['helper'] = array();
```

Kemudian tambahkan “url” (karena kita hanya membutuhkan helper url disini). Maka akan menjadi seperti berikut.

```
$autoload['helper'] = array('url');
```

Proses autoload library dan helper selesai. Selanjutnya kita akan membuat database untuk aplikasi rental\_mobil dan melakukan konfigurasi database pada codeigniter.

### 6.3 Merancang & Membuat Database Aplikasi rental\_mobil dan Konfigurasi database di Codeigniter

Berikut adalah gambaran struktur database yang akan kita buat untuk aplikasi rental\_mobil ini.

Buat sebuah database baru dengan nama “rental” kemudian buatlah table-table seperti berikut:

#### a. Tabel User (user)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
2	nama	varchar(128)	latin1_swedish_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
3	email	varchar(128)	latin1_swedish_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
4	image	varchar(128)	latin1_swedish_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
5	password	varchar(256)	latin1_swedish_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
6	role_id	int(11)			No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
7	is_active	int(1)			No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
8	tanggal_input	int(11)			No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

#### b. Tabel Role (role)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
2	role	varchar(128)	latin1_swedish_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

Isi tabel role

id	role
1	administrator
2	member

#### c. Tabel Kategori (kategori)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>id_kategori</b>	int(5)			No	None		AUTO_INCREMENT
2	<b>nama_kategori</b>	varchar(45)	latin1_swedish_ci		No	None		

Isi tabel kategori

id_kategori	nama_kategori
1	Sedan
2	MPV
3	SUV
4	PickUp
5	Truck
6	Mini Bus
7	Bus

#### d. Tabel Mobil (**mobil**)

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1 <b>id</b>	int(11)		Tidak	Tidak ada	AUTO_INCREMENT			Ubah Hapus Lainnya
<input type="checkbox"/>	2 <b>kode_mobil</b>	varchar(128)	latin1_swedish_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
<input type="checkbox"/>	3 <b>id_kategori</b>	int(11)		Tidak	Tidak ada				Ubah Hapus Lainnya
<input type="checkbox"/>	4 <b>merek</b>	varchar(64)	latin1_swedish_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
<input type="checkbox"/>	5 <b>warna</b>	varchar(64)	latin1_swedish_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
<input type="checkbox"/>	6 <b>tahun_pembuatan</b>	year(4)		Tidak	Tidak ada				Ubah Hapus Lainnya
<input type="checkbox"/>	7 <b>plat_no</b>	varchar(64)	latin1_swedish_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
<input type="checkbox"/>	8 <b>stok</b>	int(11)		Tidak	Tidak ada				Ubah Hapus Lainnya
<input type="checkbox"/>	9 <b>dipinjam</b>	int(11)		Tidak	Tidak ada				Ubah Hapus Lainnya
<input type="checkbox"/>	10 <b>dibooking</b>	int(11)		Tidak	Tidak ada				Ubah Hapus Lainnya
<input type="checkbox"/>	11 <b>image</b>	varchar(256)	latin1_swedish_ci	Tidak	Tidak ada				Ubah Hapus Lainnya

Isi tabel mobil

id	kode_mobil	id_kategori	merek	warna	tahun_pembuatan	plat_no	stok	dipinjam	dibooking	image
1	Honda All New City	1	Honda	Silver	2019	B 9999 XX	1	0	0	img1.jpg
2	Toyota Camry	1	Toyota	Putih	2017	D 8888 ZZ	1	0	0	img2.jpg
3	Daihatsu TERIOS	6	Daihatsu	putih	2018	B 1111 KL	1	0	0	img3.jpg
4	Toyota Avanza	2	Toyota	Hitam	2020	B 6666 LL	1	0	0	img4.jpg
5	Toyota Fortuner	3	Toyota	hitam	2020	B 1234 AB	1	0	0	img5.jpg

## 6.4 Membuat Koneksi Database Codeigniter dengan MySQL

Sebelum melakukan seting konfigurasi pada file database, terlebih dahulu harus disiapkan databasenya. Agar bisa langsung dipraktekkan.

Selanjutnya untuk menghubungkan codeigniter dengan database buka file *config database* codeigniter yang terletak di **application/config/database.php** kemudian cari baris sintak seperti berikut:

```

$active_group = 'default';
$query_builder = TRUE;

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => '',
    'password' => '',
    'database' => '',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);

```

Kemudian masukkan username, password, dan nama database, untuk password jika tidak ada maka bisa dibiarkan kosong. Sehingga akan menjadi.

```

$active_group = 'default';
$query_builder = TRUE;

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '',
    'database' => 'rental',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),

```

```

        'cache_on' => FALSE,
        'cachedir' => '',
        'char_set' => 'utf8',
        'dbcollat' => 'utf8_general_ci',
        'swap_pre' => '',
        'encrypt' => FALSE,
        'compress' => FALSE,
        'stricton' => FALSE,
        'failover' => array(),
        'save_queries' => TRUE
    );

```

## 6.5 Membuat Model untuk CRUD

Pada tahap ini akan dibuat fungsi-fungsi yang bisa digunakan berulang kali, tanpa harus repot banyak-banyak membuat fungsi pada model. contohnya untuk insert data ke database, hanya buat satu fungsi untuk insert data ke database. Dan bisa kita gunakan secara berulang-ulang kali. Begitu juga dengan fungsi edit, hapus dan update data pada database.

Buat 2 buah file model dengan nama **ModelUser.php** dan **ModelMobil.php**. perlu diperhatikan untuk Penamaan Controller dan Model harus diawali dengan huruf besar di codeigniter 3. Memang tidak masalah jika dibuat dengan nama kecil, tetapi akan menyalahi aturan penggunaan codeigniter yang benar dan akan terjadi trouble saat diupload dihosting.

### ModelUser.php

```

<?php
defined('BASEPATH') or exit('No direct script access allowed');

class ModelUser extends CI_Model
{
    public function simpanData($data = null)
    {
        $this->db->insert('user', $data);
    }

    public function cekData($where = null)
    {
        return $this->db->get_where('user', $where);
    }

    public function getUserWhere($where = null)
    {
        return $this->db->get_where('user', $where);
    }
}

```



```

    }

    public function cekUserAccess($where = null)
    {
        $this->db->select('*');
        $this->db->from('access_menu');
        $this->db->where($where);
        return $this->db->get();
    }

    public function getUserLimit()
    {
        $this->db->select('*');
        $this->db->from('user');
        $this->db->limit(10, 0);
        return $this->db->get();
    }
}

```

#### ModelMobil.php

```

<?php
defined('BASEPATH') or exit('No direct script access allowed');

class ModelMobil extends CI_Model
{
    //manajemen mobil
    public function getMobil()
    {
        return $this->db->get('mobil');
    }

    public function mobilWhere($where)
    {
        return $this->db->get_where('mobil', $where);
    }

    public function simpanMobil($data = null)
    {
        $this->db->insert('mobil', $data);
    }

    public function updateMobil($data = null, $where = null)
    {
        $this->db->update('mobil', $data, $where);
    }

    public function hapusMobil($where = null)

```

```

{
    $this->db->delete('mobil', $where);
}

public function total($field, $where)
{
    $this->db->select_sum($field);
    if(!empty($where) && count($where) > 0){
        $this->db->where($where);
    }
    $this->db->from('mobil');
    return $this->db->get()->row($field);
}

//manajemen kategori
public function getKategori()
{
    return $this->db->get('kategori');
}

public function kategoriWhere($where)
{
    return $this->db->get_where('kategori', $where);
}

public function simpanKategori($data = null)
{
    $this->db->insert('kategori', $data);
}

public function hapusKategori($where = null)
{
    $this->db->delete('kategori', $where);
}

public function updateKategori($where = null, $data = null)
{
    $this->db->update('kategori', $data, $where);
}

//join
public function joinKategoriMobil($where)
{
    $this->db->select('mobil.id_kategori,kategori.kategori');
    $this->db->from('mobil');
    $this->db->join('kategori', 'kategori.id
mobil.id_kategori');
    $this->db->where($where);
    return $this->db->get();
}

```

```
}
```

Dengan model seperti di atas, Untuk menginput data ke database kita bisa menggunakan function *simpan()*, untuk menampilkan data atau mengambil data dari database kita bisa menggunakan function *tampil()*, untuk menghapus data dari database kita bisa menggunakan function *hapus()*, dan untuk mengupdate data pada database kita bisa menggunakan function *ubah()*.

Kemudian kita harus mengatur autoload model di application/config/autoload.php. agar model-model yang sudah dibuat bisa digunakan dan di load secara otomatis. Jadi tidak perlu lagi kita load dari controller.

#### **Application/config/autoload.php**

Cari baris sintak berikut:

```
$autoload['model'] = array();
```

Kemudian ubah menjadi

```
$autoload['model'] = array('ModelUser', 'ModelMobil');
```

#### **Tugas 6**

- a. Kerjakan Soal yang ada di materi pertemuan 7
- b. Kumpulkan hasil pengerjaan pada pertemuan 7

## **PERTEMUAN 7**

### **REVIEW MATERI**

Untuk mengetahui sejauh mana pemahaman yang sudah anda dapat tentang materi dasar MVC dari pertemuan 1 sampai pertemuan 6, maka di pertemuan 7 ini dibuatlah sebuah studi kasus yang harus anda selesaikan. Dengan konsep MVC yang sudah dijelaskan di pertemuan 1 s/d 6. Dari membuat tampilan input sampai menampilkan hasil submit dari tampilan input tersebut. Berikut studi kasus yang harus anda selesaikan:

1. Ada sebuah toko sepatu yang ingin dibuatkan program untuk menghitung otomatis dari transaksi penjualan sepatunya yang mana harga sepatu ditentukan berdasarkan merk dari sepatu tersebut.
2. Buatlah tampilan form input transaksi di point 1 meliputi input nama pembeli menggunakan text, no hp menggunakan text, merk sepatu berupa pilihan combo/select berisi Nike, Adidas, Kickers, Eiger, Bucherri. Masing-masing memiliki harga sesuai urutan yaitu 375000, 300000, 250000, 275000, 400000., ukuran sepatu berupa combo/select juga berisi pilihan ukuran dari no 32 s/d nomor 44. Dan terakhir terdapat tombol submit untuk memproses inputan tersebut.
3. Buatlah sebuah controller dan model untuk memproses inputan form sehingga hasilnya di tampilkan ke file view yang berisi hasil dari proses submit dari form input.
4. Pada tampilan hasil, buatlah link untuk kembali menuju file input form pada point 2

#### **Tugas 7**

- a. Kerjakan kasus-kasus pada materi pertemuan 9
- b. Kumpulkan hasil pengerjaan pada pertemuan 9

**PERTEMUAN 8**  
**UJIAN TENGAH SEMESTER**

## PERTEMUAN 9

### MEMBUAT FORM LOGIN DAN PESAN NOTIFIKASI

Pada pertemuan ini mahasiswa diharapkan mampu membuat form Login dengan enkripsi password MD5 dan memahami pembuatan notifikasi kesalahan dalam inputan form.

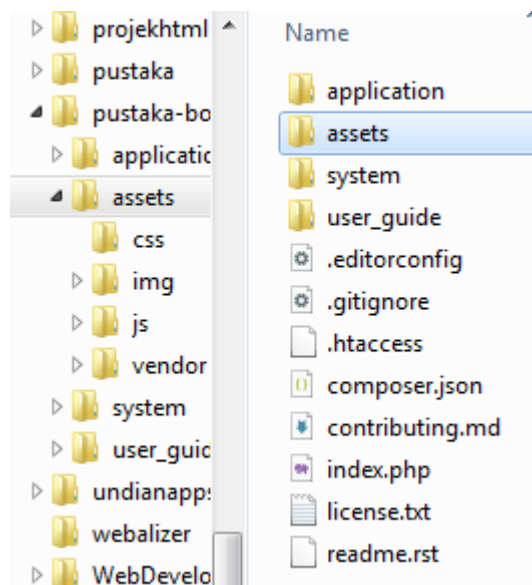
#### 9.1 Membuat Halaman Login

Halaman login akan kita buat di halaman utama pada saat aplikasi rental\_mobil bagian admin diakses.

Form login akan kita buat pada controller Autentifikasi. Controller ini akan kita jadikan controller default. Cara untuk mengaturnya sudah dijelaskan di pertemuan-pertemuan awal.

Untuk membuat tampilan-tampilan dari aplikasi rental mobil akan menggunakan template dari bootstrap yaitu **sb admin 2** kenapa menggunakan template sb admin 2 ini, karena template sb admin 2 ini sudah menggunakan bootstrap terbaru yaitu versi 4. Untuk itu kita harus mempersiapkan terlebih dahulu template sb admin 2 tersebut. Sb admin 2 sudah kami sediakan di folder tools.

Langsung saja extract file rar **startbootstrap-sb-admin-2-gh-pages.zip** kemudian cari folder css, img js, vendor kemudian copy paste ke dalam folder project kalian. Dalam hal ini adalah folder rental\_mobil. Di dalamnya kita buat folder baru dengan nama **assets**. Sehingga hasilnya seperti berikut:



Kembali ke pembuatan form login. Untuk membuatnya kita akan menggunakan tampilan form login yang ada di template sb admin 2. Untuk itu sekarang coba kembali ke folder hasil extract template sb admin 2, kemudian cari file **login.html** kemudian buka file login tersebut bisa menggunakan web browser atau menggunakan editor php atau html yang sedang kamu gunakan. Ketika sudah terbuka, copy isi script dari file login tersebut kemudian buat file view yang baru pada proyek kamu dengan nama **login.php** simpan dalam folder **Application/views/admin** lalu paste. Maka hasilnya seperti berikut:

```
<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>SB Admin 2 - Login</title>

    <!-- Custom fonts for this template-->
    <link href="vendor/fontawesome-free/css/all.min.css"
rel="stylesheet" type="text/css">
    <link
href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,30
0i,400,400i,600,600i,700,700i,800,800i,900,900i" rel="stylesheet">

    <!-- Custom styles for this template-->
    <link href="css/sb-admin-2.min.css" rel="stylesheet">

</head>

<body class="bg-gradient-primary">

    <div class="container">

        <!-- Outer Row -->
        <div class="row justify-content-center">
```

```

<div class="col-xl-10 col-lg-12 col-md-9">

  <div class="card o-hidden border-0 shadow-lg my-5">
    <div class="card-body p-0">
      <!-- Nested Row within Card Body -->
      <div class="row">
        <div class="col-lg-6 d-none d-lg-block bg-login-
image"></div>
        <div class="col-lg-6">
          <div class="p-5">
            <div class="text-center">
              <h1 class="h4 text-gray-900 mb-4">Welcome
Back!</h1>
            </div>
            <form class="user">
              <div class="form-group">
                <input type="email" class="form-control form-
control-user" id="exampleInputEmail" aria-describedby="emailHelp"
placeholder="Enter Email Address...">
              </div>
              <div class="form-group">
                <input type="password" class="form-control
form-control-user" id="exampleInputPassword" placeholder="Password">
              </div>
              <div class="form-group">
                <div class="custom-control custom-checkbox
small">
                  <input type="checkbox" class="custom-
control-input" id="customCheck">
                  <label class="custom-control-label"
for="customCheck">Remember Me</label>
                </div>
              </div>
              <a href="index.html" class="btn btn-primary btn-
user btn-block">
                Login
              </a>
              <hr>
              <a href="index.html" class="btn btn-google btn-
user btn-block">

```





```
</html>
```

Tampilan script di atas dibagi menjadi 3 file view yaitu **aute\_header.php**, **login.php**, dan **aute\_footer.php**. script bagian atas sampai tanda merah dijadikan file **aute\_header.php** simpan dalam folder **application/views/templates**, script bagian tengah dijadikan file **login.php** simpan dalam folder **application/views/autentifikasi**, dan script bagian dari tanda merah ke bawah dijadikan file **aute\_footer.php** simpan di dalam folder **application/views/templates**. Selanjutnya sesuaikan scriptnya menjadi seperti di bawah:

#### **aute\_header.php**

```
<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>WP Univ. BSI | <?= $judul; ?></title>

    <!-- Custom fonts for this template-->
    <link href="<?= base_url('assets/'); ?>vendor/fontawesome-
free/css/all.min.css" rel="stylesheet" type="text/css">
    <link
href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,30
0i,400,400i,600,600i,700,700i,800,800i,900,900i" rel="stylesheet">

    <!-- Custom styles for this template-->
    <link href="<?= base_url('assets/'); ?>css/sb-admin-2.min.css"
rel="stylesheet">

</head>

<body class="bg-gradient-primary">
```

#### **login.php**

```

<div class="container">

    <!-- Outer Row -->
    <div class="row justify-content-center">

        <div class="col-lg-7">

            <div class="card o-hidden border-0 shadow-lg my-5">
                <div class="card-body p-0">
                    <!-- Nested Row within Card Body -->
                    <div class="row">
                        <div class="col-lg">
                            <div class="p-5">
                                <div class="text-center">
                                    <h1 class="h4 text-gray-900 mb-4">Halaman Login!!</h1>
                                </div>
                                <?=$this->session->flashdata('pesan'); ?>
                                <form class="user" method="post"
action="<?=$base_url('autentifikasi'); ?>">
                                    <div class="form-group">
                                        <input type="text"
class="form-control form-control-user" value="<?=$set_value('email'); ?>" id="email" placeholder="Masukkan Alamat Email" name="email">
                                        <?=$form_error('email',
'<small class="text-danger pl-3">', '</small>'); ?>
                                        </div>
                                        <div class="form-group">
                                            <input type="password"
class="form-control form-control-user" id="password"
placeholder="Password" name="password">
                                            <?=$form_error('password',
'<small class="text-danger pl-3">', '</small>'); ?>
                                            </div>
                                            <button type="submit" class="btn btn-primary btn-user btn-block">
                                                Masuk
                                            </button>
                                        </form>
                                    <hr>

```

```
<div class="text-center">  
    <a class="small" href="{base_url('autentifikasi/lupaPassword')}">"Lupa Password?"</a>  
</div>  
    <div class="text-center">  
        <a class="small" href="{base_url('autentifikasi/registrasi')}">"Daftar Member!"</a>  
    </div>  
</div>  
</div>  
</div>  
</div>
```

**aute\_footer.php**

```

<!-- Bootstrap core JavaScript-->
<script src="<?= base_url('assets/');
?>vendor/jquery/jquery.min.js"></script>
<script src="<?= base_url('assets/');
?>vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<!-- Core plugin JavaScript-->
<script src="<?= base_url('assets/'); ?>vendor/jquery-
easing/jquery.easing.min.js"></script>

<!-- Custom scripts for all pages-->
<script src="<?= base_url('assets/'); ?>js/sb-admin-
2.min.js"></script>
<script>
    $('.alert-message').alert().delay(3000).slideUp('slow');
</script>

</body>

```

```
</html>
```

Setelah selesai membuat tampilan form login, seperti yang sudah disinggung di atas. Kita akan membuat controller Autentifikasi.php.

Buatlah controller **Autentifikasi.php** simpan dalam folder **Application/controllers** lalu buat scriptnya seperti berikut:

```
<?php

class Autentifikasi extends CI_Controller
{

    public function index()
    {
        //jika statusnya sudah login, maka tidak bisa mengakses
        halaman login alias dikembalikan ke tampilan user
        if($this->session->userdata('email')){
            redirect('user');
        }

        $this->form_validation->set_rules('email', 'Alamat Email',
        'required|trim|valid_email', [
            'required' => 'Email Harus diisi!!',
            'valid_email' => 'Email Tidak Benar!!'
        ]);
        $this->form_validation->set_rules('password', 'Password',
        'required|trim', [
            'required' => 'Password Harus diisi'
        ]);
        if ($this->form_validation->run() == false) {
            $data['judul'] = 'Login';
            $data['user'] = '';
            //kata 'login' merupakan nilai dari variabel judul dalam
            array $data dikirimkan ke view aute_header
            $this->load->view('templates/aute_header', $data);
            $this->load->view('autentifikasi/login');
            $this->load->view('templates/aute_footer');
        } else {
            $this->_login();
        }
    }
}
```

```

    }

private function _login()
{
    $email = htmlspecialchars($this->input->post('email',
true));
    $password = $this->input->post('password', true);

    $user = $this->ModelUser->cekData(['email' => $email])-
>row_array();

    //jika usernya ada
    if ($user) {
        //jika user sudah aktif
        if ($user['is_active'] == 1) {
            //cek password
            if (password_verify($password, $user['password'])) {
                $data = [
                    'email' => $user['email'],
                    'role_id' => $user['role_id']
                ];

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

                if ($user['role_id'] == 1) {
                    redirect('admin');
                } else {
                    if ($user['image'] == 'default.jpg') {
                        $this->session->set_flashdata('pesan',
'<div class="alert alert-info alert-message" role="alert">Silahkan
Ubah Profile Anda untuk Ubah Photo Profil</div>');
                    }
                    redirect('user');
                }
            } else {
                $this->session->set_flashdata('pesan', '<div
class="alert alert-danger alert-message" role="alert">Password
salah!!</div>');
                redirect('autentifikasi');
            }
        } else {

```

```

        $this->session->set_flashdata('pesan', '<div
class="alert alert-danger alert-message" role="alert">User belum
diaktifasi!!</div>');
        redirect('autentifikasi');
    }
} else {
    $this->session->set_flashdata('pesan', '<div
class="alert alert-danger alert-message" role="alert">Email tidak
terdaftar!!</div>');
    redirect('autentifikasi');
}
}
}

```

Kemudian tambahkan method **cekData()** dalam ModelUser.php yang sebelumnya sudah dibuat. Di bawah ini script tambahan method **cekData()**

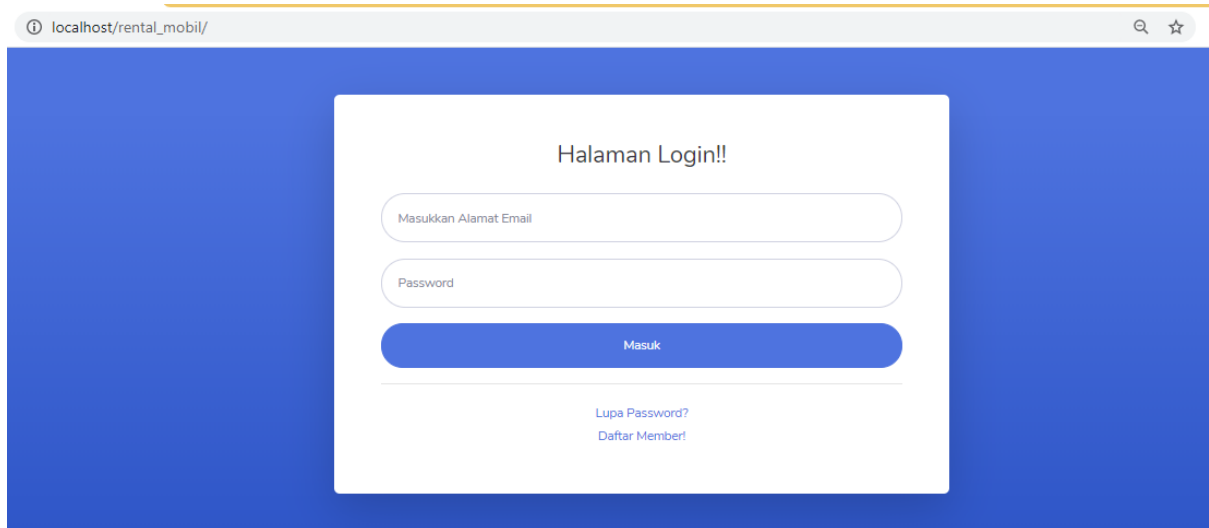
```

public function cekData($where = null)
{
    return $this->db->get_where('user', $where);
}

```

Sebelum melihat hasilnya, ubah pengaturan controller default terlebih dahulu yang sebelum controller welcome ubah menjadi controller autentifikasi. Pada folder **Application/config/routes.php**

Untuk melihat hasilnya silahkan ketik URL seperti ini **localhost/rental\_mobil/**, maka hasilnya seperti dibawah ini:



## 9.2 Membuat Halaman Admin

Halaman admin adalah halaman backend yang digunakan untuk mengelola data-data master seperti mobil, kategori mobil dll. Untuk membuatnya kita akan melibatkan controller autentifikasi.php

Dalam pembuatan halaman admin, template yang digunakan tetep template dari sb admin 2. Untuk itu coba kita buka index.html dari sb admin 2 menggunakan editor yang kamu miliki. Selanjutnya copy isi script dari index.html, kemudian buat view baru dalam folder **Application/views/admin/** beri nama **index.php** lalu paste scriptnya. Selanjutnya file index ini akan dipecah menjadi 5 file view yaitu: **header.php**, **index.php**, **footer.php**, **sidebar.php**, dan **topbar.php**. jadi isi dari file index.php yang baru saja di buat, disesuaikan dengan 5 file di bawah ini:

### header.php

```
<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1, shrink-to-fit=no">
    <meta name="description" content="">
```



```

<meta name="author" content="">

<title>Rental Mobil | <?= $judul; ?></title>

<!-- Custom fonts for this template-->
<link href="<?= base_url('assets/'); ?>vendor/fontawesome-free/css/all.min.css" rel="stylesheet" type="text/css">
<link href="<?= base_url('assets/'); ?>datatable/datatables.css" rel="stylesheet" type="text/css">
<script type="text/javascript" src="<?= base_url('assets/'); ?>datatable/jquery.dataTables.js"; ?>></script>
<script type="text/javascript" src="<?= base_url('assets/'); ?>datatable/datatables.js"; ?>></script>
<link href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400i,600,600i,700,700i,800,800i,900,900i" rel="stylesheet">

<!-- Custom styles for this template-->
<link href="<?= base_url('assets/'); ?>css/sb-admin-2.min.css" rel="stylesheet">

</head>

<body id="page-top">

    <!-- Page Wrapper -->
    <div id="wrapper">

```

## index.php

```

<!-- Begin Page Content -->
<div class="container-fluid">

    <!-- row ux-->
    <div class="row">
        <div class="col-xl-3 col-md-6 mb-4">
            <div class="card border-left-danger shadow h-100 py-2 bg-primary">
                <div class="card-body">
                    <div class="row no-gutters align-items-center">
                        <div class="col mr-2">

```

```

        <div class="text-md font-weight-bold text-white text-
uppercase mb-1">Jumlah Anggota</div>
        <div class="h1 mb-0 font-weight-bold text-white"><?=
$this->ModelUser->getUserWhere(['role_id' => 1])->num_rows();
?></div>
    </div>
    <div class="col-auto">
        <a href="<?= base_url('user/anggota'); ?>"><i
class="fas fa-users fa-3x text-warning"></i></a>
    </div>
</div>
</div>
</div>
</div>
</div>

<div class="col-xl-3 col-md-6 mb-4">
    <div class="card border-left-primary shadow h-100 py-2 bg-
warning">
        <div class="card-body">
            <div class="row no-gutters align-items-center">
                <div class="col mr-2">
                    <div class="text-md font-weight-bold text-white text-
uppercase mb-1">Stok Mobil Tendaftar</div>
                    <div class="h1 mb-0 font-weight-bold text-white">
                        <?php
                            $where = ['stok != 0'];
                            $totalstok = $this->ModelMobil->total('stok',
$where);
                            echo $totalstok;
                        ?>
                    </div>
                </div>
                <div class="col-auto">
                    <a href="<?= base_url('mobil'); ?>"><i class="fas fa-
book fa-3x text-primary"></i></a>
                </div>
            </div>
        </div>
    </div>
</div>
</div>

<div class="col-xl-3 col-md-6 mb-4">

```

```

        <div class="card border-left-success shadow h-100 py-2 bg-
danger">
        <div class="card-body">
            <div class="row no-gutters align-items-center">
                <div class="col mr-2">
                    <div class="text-md font-weight-bold text-white text-
uppercase mb-1">Mobil yang dipinjam</div>
                    <div class="h1 mb-0 font-weight-bold text-white">
                        <?php
                            $where = ['dipinjam != 0'];
                            $totaldipinjam = $this->ModelMobil-
>total('dipinjam', $where);
                            echo $totaldipinjam;
                        ?>
                    </div>
                </div>
                <div class="col-auto">
                    <a href="<?= base_url('user'); ?>"><i class="fas fa-
user-tag fa-3x text-success"></i></a>
                </div>
            </div>
        </div>
    </div>

    <div class="col-xl-3 col-md-6 mb-4">
        <div class="card border-left-warning shadow h-100 py-2 bg-
success">
            <div class="card-body">
                <div class="row no-gutters align-items-center">
                    <div class="col mr-2">
                        <div class="text-md font-weight-bold text-white text-
uppercase mb-1">Mobil yang dibooking</div>
                        <div class="h1 mb-0 font-weight-bold text-white">
                            <?php
                                $where = ['dibooking !=0'];
                                $totaldibooking = $this->ModelMobil-
>total('dibooking', $where);
                                echo $totaldibooking;
                            ?>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

        <div class="col-auto">
            <a href="<?= base_url('user'); ?>"><i class="fas fa-
shopping-cart fa-3x text-danger"></i></a>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
<!-- end row ux-->

<!-- Divider -->
<hr class="sidebar-divider">

<!-- row table-->
<div class="row">
    <div class="table-responsive table-bordered col-sm-5 ml-auto mr-
auto mt-2">
        <div class="page-header">
            <span class="fas fa-users text-primary mt-2 "> Data
User</span>
            <a class="text-danger" href="<?php echo
base_url('user/data_user'); ?>"><i class="fas fa-search mt-2 float-
right"> Tampilkan</i></a>
        </div>
        <table class="table mt-3">
            <thead>
                <tr>
                    <th>#</th>
                    <th>Nama Anggota</th>
                    <th>Email</th>
                    <th>Role ID</th>
                    <th>Aktif</th>
                    <th>Member Sejak</th>
                </tr>
            </thead>
            <tbody>
                <?php
                $i = 1;
                foreach ($anggota as $a) { ?>
                    <tr>
                        <td><?= $i++; ?></td>

```

```

        <td><?= $a['nama']; ?></td>
        <td><?= $a['email']; ?></td>
        <td><?= $a['role_id']; ?></td>
        <td><?= $a['is_active']; ?></td>
        <td><?= date('Y', $a['tanggal_input']); ?></td>
    </tr>
<?php } ?>
</tbody>
</table>
</div>

<div class="table-responsive table-bordered col-sm-5 ml-auto mr-
auto mt-2">
    <div class="page-header">
        <span class="fas fa-book text-warning mt-2"> Data
Mobil</span>
        <a href="<?= base_url('mobil'); ?>"><i class="fas fa-search
text-primary mt-2 float-right"> Tampilkan</i></a>
    </div>
    <div class="table-responsive">
        <table class="table mt-3" id="table-datatable">
            <thead>
                <tr>
                    <th>#</th>
                    <th>Nama Mobil</th>
                    <th>Merek</th>
                    <th>Warna</th>
                    <th>Tahun Mobil</th>
                    <th>Nomor Plat</th>
                    <th>Stok</th>
                </tr>
            </thead>
            <tbody>
                <?php
                $i = 1;
                foreach ($mobil as $b) { ?>
                    <tr>
                        <td><?= $i++; ?></td>
                        <td><?= $b['kode_mobil']; ?></td>
                        <td><?= $b['merek']; ?></td>
                        <td><?= $b['warna']; ?></td>

```

```

        <td><?= $b['tahun_pembuatan']; ?></td>
        <td><?= $b['plat_no']; ?></td>
        <td><?= $b['stok']; ?></td>
    </tr>
    <?php } ?>
</tbody>
</table>
</div>
</div>

</div>
<!-- end of row table-->

</div>
<!-- /.container-fluid -->

</div>
<!-- End of Main Content -->

```

## footer.php

```

<!-- Footer -->
<footer class="sticky-footer bg-white">
    <div class="container my-auto">
        <div class="copyright text-center my-auto">
            <span>Copyright &copy; Web Programming SMTIK Nusa
Mandiri with Bootstrap SB Admin 2 <?= date('Y'); ?></span>
        </div>
    </div>
</footer>
<!-- End of Footer -->

</div>
<!-- End of Content Wrapper -->

</div>
<!-- End of Page Wrapper -->

<!-- Scroll to Top Button-->
<a class="scroll-to-top rounded" href="#page-top">

```

```

        <i class="fas fa-angle-up"></i>
    </a>

<!-- Logout Modal-->
<div class="modal fade" id="logoutModal" tabindex="-1" role="dialog"
aria-labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="exampleModalLabel">Yakin
mau keluar?</h5>
                <button class="close" type="button" data-
dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">x</span>
                </button>
            </div>
            <div class="modal-body">Pilih "Logout" di bawah jika
kamu yakin sudah selesai.</div>
            <div class="modal-footer">
                <button class="btn btn-secondary" type="button"
data-dismiss="modal">Cancel</button>
                <a class="btn btn-primary" href="{?=
base_url('autentifikasi/logout'); ?}">Logout</a>
            </div>
        </div>
    </div>
</div>

<!-- Bootstrap core JavaScript-->
<script src="{?= base_url('assets/');
?}>vendor/jquery/jquery.min.js"></script>
<script src="{?= base_url('assets/');
?}>vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<!-- Core plugin JavaScript-->
<script src="{?= base_url('assets/'); ?}>vendor/jquery-
easing/jquery.easing.min.js"></script>

<!-- Custom scripts for all pages-->
<script src="{?= base_url('assets/'); ?}>js/sb-admin-
2.min.js"></script>

```

```

<script>
    $('.custom-file-input').on('change', function() {
        let fileName = $(this).val().split('\\').pop();
        $(this).next('.custom-file-label').addClass("selected").html(fileName);
    });

    $(document).ready(function() {
        $("#table-datatable").dataTable();
    });
    $('.alert-message').alert().delay(3500).slideUp('slow');
</script>

</body>

</html>

```

## sidebar.php

```

<!-- Sidebar -->
<ul class="navbar-nav bg-gradient-primary sidebar sidebar-dark accordion" id="accordionSidebar">

    <!-- Sidebar - Brand -->
    <a class="sidebar-brand d-flex align-items-center justify-content-center" href="index.html">
        <div class="sidebar-brand-icon rotate-n-15">
            <i class="fas fa-book"></i>
        </div>
        <div class="sidebar-brand-text mx-3">Rental
Mobil</div>
    </a>

    <!-- Divider -->
    <hr class="sidebar-divider">

    <!-- Looping Menu-->

    <!-- Heading -->
    <div class="sidebar-heading">

```



```

        Master Data
    </div>

    <!-- Nav Item - Dashboard -->
    <li class="nav-item active">
        <!-- Nav Item - Dashboard -->
        <li class="nav-item">
            <a class="nav-link pb-0" href="{%=
base_url('mobil'); %}">
                <i class="fa fa-fw fa book"></i>
                <span>Data Mobil</span></a>
            </li>
            <li class="nav-item">
                <a class="nav-link pb-0" href="{%=
base_url('user/anggota'); %}">
                    <i class="fa fa-fw fa book"></i>
                    <span>Data Anggota</span></a>
                </li>
            </li>
        </li>
        <!-- Divider -->
        <hr class="sidebar-divider mt-3">

        <!-- Sidebar Toggler (Sidebar) -->
        <div class="text-center d-none d-md-inline">
            <button class="rounded-circle border-0"
id="sidebarToggle"></button>
        </div>

    </ul>
    <!-- End of Sidebar -->

```

## topbar.php

```

<!-- Content Wrapper -->
<div id="content-wrapper" class="d-flex flex-column">

    <!-- Main Content -->
    <div id="content">

        <!-- Topbar -->

```

```

        <nav class="navbar navbar-expand navbar-light bg-white
topbar mb-4 static-top shadow">

        <!-- Sidebar Toggle (Topbar) -->
        <button id="sidebarToggleTop" class="btn btn-link d-md-
none rounded-circle mr-3">
            <i class="fa fa-bars"></i>
        </button>

        <!-- Page Heading -->
        <h1 class="h3 mb-2 text-gray-800"><?= $judul; ?></h1>

        <!-- Topbar Navbar -->
        <ul class="navbar-nav ml-auto">

            <div class="topbar-divider d-none d-sm-block"></div>

            <!-- Nav Item - User Information -->
            <li class="nav-item dropdown no-arrow">
                <a class="nav-link dropdown-toggle" href="#"
id="userDropdown" role="button" data-toggle="dropdown" aria-
haspopup="true" aria-expanded="false">
                    <span class="mr-2 d-none d-lg-inline text-
gray-600 small"><?= $user['nama']; ?> </span>
                    
                    <!-- Dropdown - User Information -->
                    <div class="dropdown-menu dropdown-menu-right
shadow animated--grow-in" aria-labelledby="userDropdown">
                        <a class="dropdown-item" href="<?=
base_url('user'); ?>"
                            <i class="fas fa-user fa-sm fa-fw mr-2
text-gray-400"></i>
                            Profile Saya
                        </a>
                        <div class="dropdown-divider"></div>
                        <a class="dropdown-item" href="<?=
base_url('autentifikasi/logout'); ?>" data-toggle="modal" data-
target="#logoutModal">
                            <i class="fas fa-sign-out-alt fa-sm fa-
fw mr-2 text-gray-400"></i>

```

```

                Logout
            </a>
        </div>
    </li>

</ul>

</nav>
<!-- End of Topbar -->

```

Setelah membuat file view pecahan dari tampilan admin, maka selanjutnya kita harus membuat controller **Admin.php** simpan dalam folder **Application/controllers** lalu buatlah scriptnya seperti berikut:

```

<?php
defined('BASEPATH') or exit('No direct script access allowed');

class Admin extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        cek_login();
    }

    public function index()
    {
        $data['judul'] = 'Dashboard';
        $data['user'] = $this->ModelUser->cekData(['email' => $this->session->userdata('email')])->row_array();
        $data['anggota'] = $this->ModelUser->getUserLimit()->result_array();
        $data['mobil'] = $this->ModelMobil->getMobil()->result_array();

        $this->load->view('templates/header', $data);
        $this->load->view('templates/sidebar', $data);
        $this->load->view('templates/topbar', $data);
        $this->load->view('admin/index', $data);
        $this->load->view('templates/footer');
    }
}

```

}

### **Tugas 8**

- a. Mahasiswa mengerjakan latihan yang ada di materi pertemuan 10 yaitu membuat helper dan membuat halaman registrasi member, membuat halaman my profile dan ubah profile. Lihat modul halaman
- b. Kumpulkan hasil pengerjaan pada pertemuan 10

## PERTEMUAN 10

### MEMBUAT FORM REGISTRASI USER

Pada pertemuan ini mahasiswa diharapkan dapat memahami konsep registrasi dan membuat form input registrasi.

Dalam kasus ini, sementara form registrasi bisa diakses melalui halaman login dengan link registrasi user. Dikarenakan pada materi web programming 2 ini pembahasannya belum sampai membuat halaman frontend. Yaitu halaman member dan pengunjung. Jika sudah sampai halaman frontend, barulah link untuk mengakses form registrasi akan diletakkan pada bagian frontend.

#### 10.1 Membuat helper

File helper ini dibuat untuk membuat pengecekan status login dari user. Karena akan digunakan di setiap form, agar lebih efektif maka dibuatlah helper. Aturan penamaan file helper ini harus diakhiri dengan kata helper contoh **booking\_helper.php**, **rental\_helper.php** dan lain-lain. Ketika helper ini akan digunakan atau di load, pemanggilannya cukup kata depannya saja. Seperti contoh helper diatas yaitu di load **booking** dan **rental** tanpa menyertakan kata **\_helper** nya.

Buatlah file baru beri nama **rental\_helper.php** kemudian simpan dalam folder **application/helpers** dan berikut script dari **rental\_helper.php**:

```
<?php

function cek_login()
{
    $ci = get_instance();

    if (!$ci->session->userdata('email')) {
        $ci->session->set_flashdata('pesan', '<div class="alert
alert-danger" role="alert">Akses ditolak. Anda belum login!!
</div>');
        redirect('autentifikasi');
    } else {
        $role_id = $ci->session->userdata('role_id');
```

```
}  
}
```

Dari tampilan script di atas, terlihat ada dua function yaitu **cek\_login()** untuk memastikan sudah login atau belum.

Selanjutnya buat file view **blok.php** dan **gagal.php** kedua file ini digunakan untuk menampilkan tampilan gagal dan akses di blok. Untuk membuatnya kita bisa menggunakan template yang sudah disediakan oleh sb admin 2. Cari file **404.html** kemudian buka dengan editor lalu copy isi scriptnya dan paste pada file blok.php dan file gagal.php. selanjutnya samakan scriptnya seperti berikut:

#### **blok.php**

```
<!DOCTYPE html>  
<html lang="en">  
  
<head>  
  
    <meta charset="utf-8">  
    <meta http-equiv="X-UA-Compatible" content="IE=edge">  
    <meta name="viewport" content="width=device-width, initial-  
scale=1, shrink-to-fit=no">  
    <meta name="description" content="">  
    <meta name="author" content="">  
  
    <title>Akses diblock </title>  
  
    <!-- Custom fonts for this template-->  
    <link href="<?= base_url('assets/');?>vendor/fontawesome-  
free/css/all.min.css" rel="stylesheet" type="text/css">  
    <link  
href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,30  
0i,400,400i,600,600i,700,700i,800,800i,900,900i" rel="stylesheet">  
  
    <!-- Custom styles for this template-->  
    <link href="<?= base_url('assets/');?>css/sb-admin-2.min.css"  
rel="stylesheet">  
  
</head>  
  
<body id="page-top">
```

```

<!-- Page Wrapper -->
<div id="wrapper">

    <!-- Content Wrapper -->
    <div id="content-wrapper" class="d-flex flex-column">

        <!-- Main Content -->
        <div id="content">

            <!-- Begin Page Content -->
            <div class="container-fluid mt-5">

                <!-- 404 Error Text -->
                <div class="text-center">
                    <div class="error mx-auto" data-text="403">403</div>
                    <p class="lead text-gray-800 mb-5">Akses tidak
diizinkan</p>
                    <p class="text-gray-500 mb-0">Sepertinya kamu megakses
halaman yang salah</p>
                    <a href="{?= base_url('user');?}">&larr; Kembali ke
Dashboard</a>
                </div>

            </div>
            <!-- /.container-fluid -->

        </div>
        <!-- End of Main Content -->

    </div>
    <!-- End of Content Wrapper -->

</div>
<!-- End of Page Wrapper -->

<!-- Scroll to Top Button-->
<a class="scroll-to-top rounded" href="#page-top">
    <i class="fas fa-angle-up"></i>
</a>

<!-- Logout Modal-->

```

```

    <div class="modal fade" id="logoutModal" tabindex="-1"
role="dialog" aria-labelledby="exampleModalLabel" aria-
hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="exampleModalLabel">Ready to
Leave?</h5>
                <button class="close" type="button" data-dismiss="modal"
aria-label="Close">
                    <span aria-hidden="true">x</span>
                </button>
            </div>
            <div class="modal-body">Select "Logout" below if you are
ready to end your current session.</div>
            <div class="modal-footer">
                <button class="btn btn-secondary" type="button" data-
dismiss="modal">Cancel</button>
                <a class="btn btn-primary" href="login.html">Logout</a>
            </div>
        </div>
    </div>
</div>

<!-- Bootstrap core JavaScript-->
<script src="<?=
base_url('assets/');?>vendor/jquery/jquery.min.js"></script>
<script src="<?=
base_url('assets/');?>vendor/bootstrap/js/bootstrap.bundle.min.js"><
/script>

<!-- Core plugin JavaScript-->
<script src="<?= base_url('assets/');?>vendor/jquery-
easing/jquery.easing.min.js"></script>

<!-- Custom scripts for all pages-->
<script src="<?= base_url('assets/');?>js/sb-admin-
2.min.js"></script>

</body>

</html>

```



## gagal.php

```
<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>Gagal Aktivasi Akun </title>

    <!-- Custom fonts for this template-->
    <link href="<?= base_url('assets/'); ?>vendor/fontawesome-
free/css/all.min.css" rel="stylesheet" type="text/css">
    <link
href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,30
0i,400,400i,600,600i,700,700i,800,800i,900,900i" rel="stylesheet">

    <!-- Custom styles for this template-->
    <link href="<?= base_url('assets/'); ?>css/sb-admin-2.min.css"
rel="stylesheet">

</head>

<body id="page-top">

    <!-- Page Wrapper -->
    <div id="wrapper">

        <!-- Content Wrapper -->
        <div id="content-wrapper" class="d-flex flex-column">

            <!-- Main Content -->
            <div id="content">
```

```

        <!-- Begin Page Content -->
        <div class="container-fluid mt-5">

            <!-- 404 Error Text -->
            <div class="text-center">
                <?= $this->session->flashdata('pesan'); ?>
                <a href="<?= base_url('autentifikasi'); ?>"
class="btn btn-secondary ">&larr; Close </a>
            </div>

            </div>
            <!-- /.container-fluid -->

        </div>
        <!--End of Main Content -->

    </div>
    <!--End of Content Wrapper -->
</div>
<!--End of Page Wrapper -->

<!--Scroll to Top Button -->
<a class="scroll-to-top rounded" href="#page-top"><i
class="fas fa-angle-up"></i></a>

<!-- Bootstrap core JavaScript -->
<script src="<?= base_url('assets/');
?>vendor/jquery/jquery.min.js"></script>
<script src="<?= base_url('assets/');
?>vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<!-- Core plugin JavaScript -->
<script src="<?= base_url('assets/'); ?>vendor/jquery-
easing/jquery.easing.min.js"></script>

<!-- Custom scripts for all pages-->
<script src="<?= base_url('assets/'); ?>js/sb-admin-
2.min.js"></script>
<script>
    $('<code>.alert-message</code>').alert().delay(3500).slideUp('slow');
</script>

```

```
</body>
```

```
</html>
```

Selanjutnya tambahkan method untuk mengontrol blok.php dan gagal php di atas dalam controller autentifikasi.php berikut scriptnya:

```
public function blok()
{
    $this->load->view('autentifikasi/blok');
}

public function gagal()
{
    $this->load->view('autentifikasi/gagal');
}
```

## 10.2 Membuat Form Registrasi

Setelah membuat tampilan-tampilan blok dan gagal. Barulah membuat tampilan registrasi.

Buatlah view baru beri nama **registrasi.php** kemudian simpan dalam folder **Application/views/autentifikasi** lalu buka file view **login.php** copy semua script login dan paste pada file **registrasi.php**. Selanjutnya tambahkan inputan sehingga hasilnya seperti berikut:

```
<div class="container">

    <div class="card o-hidden border-0 shadow-lg my-5 col-lg-7 mx-
auto">

        <div class="card-body p-0">
            <!-- Nested Row within Card Body -->
            <div class="row">
                <div class="col-lg">
                    <div class="p-5">
                        <div class="text-center">
                            <h1 class="h4 text-gray-900 mb-
4">Daftar Menjadi Member!</h1>
                        </div>
                        <form class="user" method="post"
action="<?= base_url('autentifikasi/registrasi'); ?>">
```

```

        <div class="form-group">
            <input type="text" class="form-
control form-control-user" id="nama" name="nama" placeholder="Nama
Lengkap" value="<?= set_value('nama'); ?>">
            <?= form_error('nama', '<small
class="text-danger pl-3">', '</small>'); ?>
        </div>
        <div class="form-group">
            <input type="text" class="form-
control form-control-user" id="email" name="email"
placeholder="Alamat Email" value="<?= set_value('email'); ?>">
            <?= form_error('email', '<small
class="text-danger pl-3">', '</small>'); ?>
        </div>
        <div class="form-group row">
            <div class="col-sm-6 mb-3 mb-sm-
0">
                <input type="password"
class="form-control form-control-user" id="password1"
name="password1" placeholder="Password">
                <?= form_error('password1',
'<small class="text-danger pl-3">', '</small>'); ?>
            </div>
            <div class="col-sm-6">
                <input type="password"
class="form-control form-control-user" id="password2"
name="password2" placeholder="Ulangi Password">
                <?= form_error('password2',
'<small class="text-danger pl-3">', '</small>'); ?>
            </div>
        </div>
        <button type="submit" class="btn btn-
primary btn-user btn-block">
            Daftar Menjadi Member
        </button>
    </form>
    <hr>
    <div class="text-center">
        <a class="small" href="<?=
base_url('autentifikasi/lupaPassword'); ?>">Lupa Password?</a>
    </div>
    <div class="text-center">

```

```

                Sudah Menjadi Member?<a class="small"
href="<?= base_url('autentifikasi'); ?>"> Login!</a>
            </div>
        </div>
    </div>
</div>
</div>
</div>
</div>

```

Selanjutnya setelah membuat view registrasi. Kita tambahkan method pada controller autentifikasi. Nama methodnya **registrasi()** seperti berikut:

```

public function registrasi()
{
    if ($this->session->userdata('email')) {
        redirect('user');
    }
    //membuat rule untuk inputan nama agar tidak boleh kosong
    dengan membuat pesan error dengan
    //bahasa sendiri yaitu 'Nama Belum diisi'
    $this->form_validation->set_rules('nama', 'Nama Lengkap',
    'required', [
        'required' => 'Nama Belum diis!!'
    ]);
    //membuat rule untuk inputan email agar tidak boleh kosong,
    tidak ada spasi, format email harus valid
    //dan email belum pernah dipakai sama user lain dengan
    membuat pesan error dengan bahasa sendiri
    //yaitu jika format email tidak benar maka pesannya 'Email
    Tidak Benar!!'. jika email belum diisi,
    //maka pesannya adalah 'Email Belum diisi', dan jika email
    yang diinput sudah dipakai user lain,
    //maka pesannya 'Email Sudah dipakai'
    $this->form_validation->set_rules('email', 'Alamat Email',
    'required|trim|valid_email|is_unique[user.email]', [
        'valid_email' => 'Email Tidak Benar!!',
        'required' => 'Email Belum diisi!!',
        'is_unique' => 'Email Sudah Terdaftar!'
    ]);
}

```

```

        //membuat rule untuk inputan password agar tidak boleh
        kosong, tidak ada spasi, tidak boleh kurang dari
        //dari 3 digit, dan password harus sama dengan repeat
        password dengan membuat pesan error dengan
        //bahasa sendiri yaitu jika password dan repeat password
        tidak diinput sama, maka pesannya
        //'Password Tidak Sama'. jika password diisi kurang dari 3
        digit, maka pesannya adalah
        //'Password Terlalu Pendek'.
        $this->form_validation->set_rules('password1', 'Password',
        'required|trim|min_length[3]|matches[password2]', [
            'matches' => 'Password Tidak Sama!!',
            'min_length' => 'Password Terlalu Pendek'
        ]);
        $this->form_validation->set_rules('password2', 'Repeat
        Password', 'required|trim|matches[password1]');
        //jika jika disubmit kemudian validasi form diatas tidak
        berjalan, maka akan tetap berada di
        //tampilan registrasi. tapi jika disubmit kemudian validasi
        form diatas berjalan, maka data yang
        //diinput akan disimpan ke dalam tabel user
        if ($this->form_validation->run() == false) {
            $data['judul'] = 'Registrasi Member';
            $this->load->view('templates/aut_header', $data);
            $this->load->view('autentifikasi/registrasi');
            $this->load->view('templates/aut_footer');
        } else {
            $email = $this->input->post('email', true);
            $data = [
                'nama' => htmlspecialchars($this->input-
>post('nama', true)),
                'email' => htmlspecialchars($email),
                'image' => 'default.jpg',
                'password' => password_hash($this->input-
>post('password1'), PASSWORD_DEFAULT),
                'role_id' => 2,
                'is_active' => 0,
                'tanggal_input' => time()
            ];

            $this->ModelUser->simpanData($data); //menggunakan model

```

```

        $this->session->set_flashdata('pesan', '<div
class="alert alert-success alert-message" role="alert">Selamat!!
akun member anda sudah dibuat. Silahkan Aktivasi Akun anda</div>');
        redirect('autentifikasi');
    }
}

```

Setelah controller dibuat, selanjutnya buatlah sebuah model baru beri nama **ModelUser.php** kemudian simpan dalam folder **Application/models/** lalu buatlah script di bawah:

```

<?php
defined('BASEPATH') or exit('No direct script access allowed');

class ModelUser extends CI_Model
{
    public function simpanData($data = null)
    {
        $this->db->insert('user', $data);
    }

    public function cekData($where = null)
    {
        return $this->db->get_where('user', $where);
    }

    public function getUserWhere($where = null)
    {
        return $this->db->get_where('user', $where);
    }

    public function cekUserAccess($where = null)
    {
        $this->db->select('*');
        $this->db->from('access_menu');
        $this->db->where($where);
        return $this->db->get();
    }

    public function getUserLimit()
    {
        $this->db->select('*');
    }
}

```

```

        $this->db->from('user');
        $this->db->limit(10, 0);
        return $this->db->get();
    }
}

```

### 10.3 Membuat Halaman My Profile dan Ubah Profile

Halaman my profile ini akan di tampilkan setelah login. Link untk mengaksesnya berada di pojok kanan atas tampilan halaman utama admin.

Buatlah controller baru beri nama **User.php** kemudian simpan dalam folder **Application/controlles/** lalu buatlah script berikut:

```

<?php
defined('BASEPATH') or exit('No direct script access allowed');

class User extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        cek_login();
    }

    public function index()
    {
        $data['judul'] = 'Profil Saya';
        $data['user'] = $this->ModelUser->cekData(['email' => $this->session->userdata('email')])>row_array();

        $this->load->view('templates/header', $data);
        $this->load->view('templates/sidebar', $data);
        $this->load->view('templates/topbar', $data);
        $this->load->view('user/index', $data);
        $this->load->view('templates/footer');
    }
}

```



```

public function anggota()
{
    $data['judul'] = 'Data Anggota';
    $data['user'] = $this->ModelUser->cekData(['email' => $this->session->userdata('email')])>row_array();
    $this->db->where('role_id', 1);
    $data['anggota'] = $this->db->get('user')>result_array();

    $this->load->view('templates/header', $data);
    $this->load->view('templates/sidebar', $data);
    $this->load->view('templates/topbar', $data);
    $this->load->view('user/anggota', $data);
    $this->load->view('templates/footer');
}

public function ubahProfil()
{
    $data['judul'] = 'Ubah Profil';
    $data['user'] = $this->ModelUser->cekData(['email' => $this->session->userdata('email')])>row_array();

    $this->form_validation->set_rules('nama', 'Nama Lengkap',
    'required|trim', [
        'required' => 'Nama tidak Boleh Kosong'
    ]);

    if ($this->form_validation->run() == false) {
        $this->load->view('templates/header', $data);
        $this->load->view('templates/sidebar', $data);
        $this->load->view('templates/topbar', $data);
        $this->load->view('user/ubah-profile', $data);
        $this->load->view('templates/footer');
    } else {
        $nama = $this->input->post('nama', true);
        $email = $this->input->post('email', true);

        //jika ada gambar yang akan diupload
        $upload_image = $_FILES['image']['name'];

        if ($upload_image) {
            $config['upload_path'] = './assets/img/profile/';

```

```

        $config['allowed_types'] = 'gif|jpg|png';
        $config['max_size']      = '3000';
        $config['max_width']    = '1024';
        $config['max_height']    = '1000';
        $config['file_name']     = 'pro' . time();

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

        if ($this->upload->do_upload('image')) {
            $gambar_lama = $data['user']['image'];
            if ($gambar_lama != 'default.jpg') {
                unlink(FCPATH . 'assets/img/profile/' .
$gambar_lama);
            }

            $gambar_baru = $this->upload->data('file_name');
            $this->db->set('image', $gambar_baru);
        } else { }
    }

    $this->db->set('nama', $nama);
    $this->db->where('email', $email);
    $this->db->update('user');

    $this->session->set_flashdata('pesan', '<div
class="alert alert-success alert-message" role="alert">Profil
Berhasil diubah </div>');
    redirect('user');
}
}
}

```

kemudian setelah membuat controller, selanjutnya kita buat tampilan (view ) dari my profile dan ubah profile.

Buatlah file view baru beri nama **index.php** dan **ubah-profile.php** lalu simpan dalam folder **Application/views/user/** kemudian buatlah script berikut:

### Index.php

```

<!-- Begin Page Content -->
<div class="container-fluid">

```

```

<div class="row">
    <div class="col-lg-6 justify-content-x">
        <?= $this->session->flashdata('pesan'); ?>
    </div>
</div>
<div class="card mb-3" style="max-width: 540px;">
    <div class="row no-gutters">
        <div class="col-md-4">
            
        </div>
        <div class="col-md-8">
            <div class="card-body">
                <h5 class="card-title"><?= $user['nama'];
?></h5>
                <p class="card-text"><?= $user['email']; ?></p>
                <p class="card-text"><small class="text-
muted">Jadi member sejak: <br><b><?= date('d F Y',
$user['tanggal_input']); ?></b></small></p>
            </div>
            <div class="btn btn-info ml-3 my-3">
                <a href="<?= base_url('user/ubahprofil'); ?>"
class="text text-white"><i class="fas fa-user-edit"></i> Ubah
Profil</a>
            </div>
        </div>
    </div>
</div>
<!-- /.container-fluid -->

</div>
<!-- End of Main Content -->

```

### ubah-profile.php

```

<!-- Begin Page Content -->
<div class="container-fluid">

```

```

<div class="row">
  <div class="col-lg-9">
    <?= form_open_multipart('user/ubahprofil'); ?>
    <div class="form-group row">
      <label for="email" class="col-sm-2 col-form-label">Email</label>
      <div class="col-sm-10">
        <input type="text" class="form-control"
id="email" name="email" value="<?= $user['email']; ?>" readonly>
      </div>
    </div>
    <div class="form-group row">
      <label for="nama" class="col-sm-2 col-form-label">Nama Lengkap</label>
      <div class="col-sm-10">
        <input type="text" class="form-control"
id="nama" name="nama" value="<?= $user['nama']; ?>">
        <?= form_error('nama', '<small class="text-
danger pl-3">', '</small>'); ?>
      </div>
    </div>
    <div class="form-group row">
      <div class="col-sm-2">Gambar</div>
      <div class="col-sm-10">
        <div class="row">
          <div class="col-sm-3">
            
          </div>
          <div class="col-sm-9">
            <div class="custom-file">
              <input type="file" class="custom-
file-input" id="image" name="image">
              <label class="custom-file-label"
for="image">Pilih file</label>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

```

        <div class="form-group row justify-content-end">
            <div class="col-sm-10">
                <button type="submit" class="btn btn-
primary">Ubah</button>
                <button class="btn btn-dark"
onclick="window.history.go(-1)"> Kembali</button>
            </div>
        </div>

    </form>
</div>
</div>
</div>
<!-- /.container-fluid -->

</div>
<!-- End of Main Content -->

```

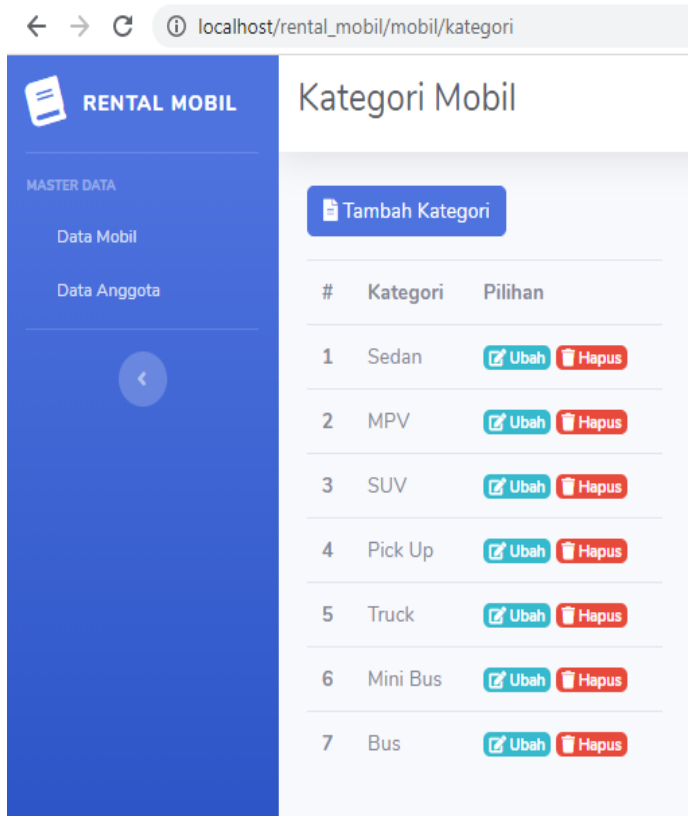
### Tugas 9

- a. Mahasiswa mengerjakan latihan yang ada di materi pertemuan 11 membuat CRUD kategori mobil
- b. Kumpulkan hasil pengerjaan pada pertemuan 11

## PERTEMUAN 11

### MEMBUAT KATEGORI MOBIL

Kategori mobil akan ditampilkan seperti berikut:



#### 11.1 Membuat Tampil Data Kategori

Berdasarkan tampilan di atas, berarti pertama kita buat tampilan data kategori mobil.

Buatlah file view baru beri nama **kategori.php** kemudian simpan dalam folder **Application/views/mobil** lalu buatlah script berikut

```
<!-- Begin Page Content -->
<div class="container-fluid">

    <?= $this->session->flashdata('pesan'); ?>
    <div class="row">
        <div class="col-lg-3">
            <?php if(validation_errors()){?>
                <div class="alert alert-danger" role="alert">
```

```

        <?= validation_errors();?>
    </div>
    <?php }?>
    <?= $this->session->flashdata('pesan'); ?>
    <a href="" class="btn btn-primary mb-3" data-
toggle="modal" data-target="#kategoriBaruModal"><i class="fas fa-
file-alt"></i> Tambah Kategori</a>
    <table class="table table-hover">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Kategori</th>
                <th scope="col">Pilihan</th>
            </tr>
        </thead>
        <tbody>

            <?php
                $a = 1;
                foreach ($kategori as $k) { ?>
                    <tr>
                        <th scope="row"><?= $a++; ?></th>
                        <td><?= $k['kategori']; ?></td>
                        <td>
                            <a href="<?=
base_url('mobil/ubahMobil/').$k['id'];?>" class="badge badge-
info"><i class="fas fa-edit"></i> Ubah</a>
                            <a href="<?=
base_url('mobil/hapusmobil/').$k['id'];?>" onclick="return
confirm('Kamu yakin akan menghapus <?= $judul.' ' . $k['kategori'];?>
?');" class="badge badge-danger"><i class="fas fa-trash"></i>
Hapus</a>
                        </td>
                    </tr>
                <?php } ?>
            </tbody>
        </table>
    </div>
</div>
<!-- /.container-fluid -->

```

```

</div>
<!-- End of Main Content -->

<!-- Modal Tambah kategori baru-->
<div class="modal fade" id="kategoriBaruModal" tabindex="-1"
role="dialog" aria-labelledby="kategoriBaruModalLabel" aria-
hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title"
id="kategoriBaruModalLabel">Tambah Kategori</h5>
                <button type="button" class="close" data-
dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <form action="<?= base_url('mobil/kategori'); ?>"
method="post">
                <div class="modal-body">
                    <div class="form-group">
                        <select name="kategori" class="form-control
form-control-user">
                            <option value="">Pilih Kategori</option>
                            <?php
                                $k = ['Sedan','MPV','SUV','Pick
Up','Truck','Mini Bus','Bus'];
                                for ($i=0;$i<7;$i++) { ?>
                                    <option value="<?= $k[$i];?>"><?=
$k[$i];?></option>
                                <?php } ?>
                            </select>
                        </div>
                    </div>
                    <div class="modal-footer">
                        <button type="button" class="btn btn-secondary"
data-dismiss="modal"><i class="fas fa-ban"></i> Close</button>
                        <button type="submit" class="btn btn-primary"><i
class="fas fa-plus-circle"></i> Tambah</button>
                    </div>
                </form>
            </div>
        </div>
    </div>

```



```

        </div>
    </div>
</div>
<!-- End of Modal Tambah Mneu -->

```

## 11.2 Hapus data kategori

Selanjutnya setelah view dibuat, maka kita buat function untuk menampilkan data kategori hapus data kategori dalam controller **Mobil.php** berikut script dari function **kategori()** dan **hapusKategori()**

```

<?php
defined('BASEPATH') or exit('No direct script access allowed');

class Mobil extends CI_Controller
{
    public function kategori()
    {
        $data['judul'] = 'Kategori Mobil';
        $data['user'] = $this->ModelUser->cekData(['email' => $this->session->userdata('email')])>row_array();
        $data['kategori'] = $this->ModelMobil->getKategori()->result_array();

        $this->form_validation->set_rules('kategori', 'Kategori', 'required', [
            'required' => 'Nama Mobil harus diisi'
        ]);

        if ($this->form_validation->run() == false) {
            $this->load->view('templates/header', $data);
            $this->load->view('templates/sidebar', $data);
            $this->load->view('templates/topbar', $data);
            $this->load->view('mobil/kategori', $data);
            $this->load->view('templates/footer');
        } else {
            $data = [
                'kategori' => $this->input->post('kategori')
            ];

```

```
        $this->ModelMobil->simpanKategori($data);  
        redirect('mobil/kategori');  
    }  
}  
  
public function hapusKategori()  
{  
    $where = ['id' => $this->uri->segment(3)];  
    $this->ModelMobil->hapusKategori($where);  
    redirect('mobil/kategori');  
}  
}
```

#### **Tugas 10**

- a. Mahasiswa mengerjakan latihan yang ada di materi pertemuan 12
- b. Kumpulkan hasil pengerjaan pada pertemuan 12

## PERTEMUAN 12

### MEMBUAT CRUD DATA MOBIL

Pada pertemuan ini mahasiswa diharapkan dapat memahami dan membuat aplikasi CRUD untuk data-data master.

Dalam kasus ini akan membuat CRUD untuk data-data mobil, sesuai dengan gambaran perancangan database di pertemuan sebelumnya. Data-data mobil yang akan disimpan adalah data id mobil, kode mobil, merek, warna, tahun pembuatan, plat nomor, jumlah, lokasi, gambar dan tanggal input.

#### 12.1 Menampilkan Data Mobil dan tambah Mobil

Langkah pertama akan dibuatkan file untuk menampilkan data mobil. Yaitu controller **Mobil.php** dengan beberapa method untuk tampil, simpan, ubah, dan hapus data.

Buatlah controller baru beri nama **Mobil.php** lalu simpan dalam folder **Application/controllers** dan buatlah script seperti di bawah ini:

```
<?php
defined('BASEPATH') or exit('No direct script access allowed');

class Mobil extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        cek_login();
    }

    //manajemen Mobil
    public function index()
    {
        $data['judul'] = 'Data Mobil';
        $data['user'] = $this->ModelUser->cekData(['email' => $this->session->userdata('email')])>row_array();
        $data['mobil'] = $this->ModelMobil->tampil()->result_array();
    }
}
```

```

        $data['kategori'] = $this->ModelMobil->getKategori()-
>result_array();

        $this->form_validation->set_rules('kode_mobil', 'Nama
Mobil', 'required|min_length[3]', [
            'required' => 'Nama Mobil harus diisi',
            'min_length' => 'Nama Mobil terlalu pendek'
        ]);
        $this->form_validation->set_rules('id_kategori', 'Kategori',
'required', [
            'required' => 'Nama Kategori harus diisi',
        ]);
        $this->form_validation->set_rules('merek', 'Merek',
'required|min_length[3]', [
            'required' => 'Merek harus diisi',
            'min_length' => 'Merek terlalu pendek'
        ]);
        $this->form_validation->set_rules('warna', 'Warna',
'required|min_length[3]', [
            'required' => 'Warna harus diisi',
            'min_length' => 'Warna terlalu pendek'
        ]);
        $this->form_validation->set_rules('tahun_pembuatan', 'Tahun
Pembuatan', 'required|min_length[3]|max_length[4]|numeric', [
            'required' => 'Tahun pembuatan harus diisi',
            'min_length' => 'Tahun pembuatan terlalu pendek',
            'max_length' => 'Tahun pembuatan terlalu panjang',
            'numeric' => 'Hanya boleh diisi angka'
        ]);
        $this->form_validation->set_rules('plat_no', 'Plat Nomor',
'required|min_length[3]', [
            'required' => 'Plat Nomor harus diisi',
            'min_length' => 'Plat Nomor terlalu pendek'
        ]);
        $this->form_validation->set_rules('stok', 'Stok',
'required|numeric', [
            'required' => 'Stok harus diisi',
            'numeric' => 'Yang anda masukan bukan angka'
        ]);

        //konfigurasi sebelum gambar diupload
        $config['upload_path'] = './assets/img/upload/';

```

```

$config['allowed_types'] = 'jpg|png|jpeg';
$config['max_size'] = '3000';
$config['max_width'] = '1024';
$config['max_height'] = '1000';
$config['file_name'] = 'img' . time();

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

if ($this->form_validation->run() == false) {
    $this->load->view('templates/header', $data);
    $this->load->view('templates/sidebar', $data);
    $this->load->view('templates/topbar', $data);
    $this->load->view('mobil/index', $data);
    $this->load->view('templates/footer');
} else {
    if ($this->upload->do_upload('image')) {
        $image = $this->upload->data();
        $gambar = $image['file_name'];
    } else {
        $gambar = '';
    }

    $data = [
        'kode_mobil' => $this->input->post('kode_mobil',
true),
        'id_kategori' => $this->input->post('id_kategori',
true),
        'merek' => $this->input->post('merek', true),
        'warna' => $this->input->post('warna', true),
        'tahun_pembuatan' => $this->input-
>post('tahun_pembuatan', true),
        'plat_no' => $this->input->post('plat_no', true),
        'stok' => $this->input->post('stok', true),
        'dipinjam' => 0,
        'dibooking' => 0,
        'image' => $gambar
    ];

    $this->ModelMobil->simpanMobil($data);
    redirect('mobil');
}
}

```

```
}
```

Setelah membuat controller mobil, selanjutnya kita buat model mobil untuk memodelkan data mobil dari tabel mobil yang nanti ditampilkan pada view index.php dari tampilan mobil.

Buatlah model baru dengan nama **ModelMobil.php** simpan dalam folder **Application/models/** kemudian buatlah script berikut:

```
<?php
defined('BASEPATH') or exit('No direct script access allowed');

class ModelMobil extends CI_Model
{
    //manajemen mobil
    public function getMobil()
    {
        return $this->db->get('mobil');
    }

    public function mobilWhere($where)
    {
        return $this->db->get_where('mobil', $where);
    }

    public function simpanMobil($data = null)
    {
        $this->db->insert('mobil', $data);
    }

    public function updateMobil($data = null, $where = null)
    {
        $this->db->update('mobil', $data, $where);
    }

    public function hapusMobil($where = null)
    {
        $this->db->delete('mobil', $where);
    }

    public function total($field, $where)
    {
        $this->db->select_sum($field);
        if(!empty($where) && count($where) > 0){
            $this->db->where($where);
        }
    }
}
```

```

    }
    $this->db->from('mobil');
    return $this->db->get()->row($field);
}

//manajemen kategori
public function getKategori()
{
    return $this->db->get('kategori');
}

public function kategoriWhere($where)
{
    return $this->db->get_where('kategori', $where);
}

public function simpanKategori($data = null)
{
    $this->db->insert('kategori', $data);
}

public function hapusKategori($where = null)
{
    $this->db->delete('kategori', $where);
}

public function updateKategori($where = null, $data = null)
{
    $this->db->update('kategori', $data, $where);
}

//join
public function joinKategoriMobil($where)
{
    $this->db->select('mobil.id_kategori,kategori.kategori');
    $this->db->from('mobil');
    $this->db->join('kategori','kategori.id =
mobil.id_kategori');
    $this->db->where($where);
    return $this->db->get();
}
}

```

Selanjutnya buatlah **index.php** untuk tampilan data mobil disimpan dalam folder **application/views/mobil** lalu buatlah script berikut:

```
<!-- Begin Page Content -->
<div class="container-fluid">

    <?= $this->session->flashdata('pesan'); ?>
    <div class="row">
        <div class="col-lg-12">
            <?php if(validation_errors()){?>
                <div class="alert alert-danger" role="alert">
                    <?= validation_errors();?>
                </div>
            <?php }?>
            <?= $this->session->flashdata('pesan'); ?>
            <a href="" class="btn btn-primary mb-3" data-
toggle="modal" data-target="#mobilBaruModal"><i class="fas fa-file-
alt"></i> Mobil Baru</a>
            <table class="table table-hover">
                <thead>
                    <tr>
                        <th scope="col">#</th>
                        <th scope="col">Nama Mobil</th>
                        <th scope="col">Merek</th>
                        <th scope="col">Warna</th>
                        <th scope="col">Tahun Pembuatan</th>
                        <th scope="col">Plat Nomor</th>
                        <th scope="col">Stok</th>
                        <th scope="col">DiPinjam</th>
                        <th scope="col">DiBooking</th>
                        <th scope="col">Gambar</th>
                        <th scope="col">Pilihan</th>
                    </tr>
                </thead>
                <tbody>

                    <?php
                        $a = 1;
                        foreach ($mobil as $b) { ?>
                            <tr>
```



```

        <th scope="row"><?= $a++; ?></th>
        <td><?= $b['kode_mobil']; ?></td>
        <td><?= $b['merek']; ?></td>
        <td><?= $b['warna']; ?></td>
        <td><?= $b['tahun_pembuatan']; ?></td>
        <td><?= $b['plat_no']; ?></td>
        <td><?= $b['stok']; ?></td>
        <td><?= $b['dipinjam']; ?></td>
        <td><?= $b['dibooking']; ?></td>
        <td>
            <picture>
                <source srcset=""
type="image/svg+xml">
                
            </picture></td>
        <td>
            <a href="<?=
base_url('mobil/ubahMobil/').$b['id'];?>" class="badge badge-
info"><i class="fas fa-edit"></i> Ubah</a>
            <a href="<?=
base_url('mobil/hapusmobil/').$b['id'];?>" onclick="return
confirm('Kamu yakin akan menghapus <?= $judul.'
'.$b['kode_mobil'];?> ?');" class="badge badge-danger"><i class="fas
fa-trash"></i> Hapus</a>
        </td>
    </tr>
</tbody>
</table>
</div>
</div>

<!-- /.container-fluid -->

</div>
<!-- End of Main Content -->

<!-- Modal Tambah mobil baru-->

```

```

<div class="modal fade" id="mobilBaruModal" tabindex="-1"
role="dialog" aria-labelledby="mobilBaruModallabel" aria-
hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title"
id="mobilBaruModallabel">Tambah Mobil</h5>
                <button type="button" class="close" data-
dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <form action="<?= base_url('mobil'); ?>" method="post"
enctype="multipart/form-data">
                <div class="modal-body">
                    <div class="form-group">
                        <input type="text" class="form-control form-
control-user" id="kode_mobil" name="kode_mobil"
placeholder="Masukkan Nama Mobil">
                    </div>
                    <div class="form-group">
                        <select name="id_kategori" class="form-
control form-control-user">
                            <option value="">Pilih Kategori</option>
                            <?php
                            foreach ($kategori as $k) { ?>
                                <option value="<?= $k['id']; ?>"><?=
$k['kategori']; ?></option>
                            <?php } ?>
                        </select>
                    </div>
                    <div class="form-group">
                        <input type="text" class="form-control form-
control-user" id="merek" name="merek" placeholder="Masukkan nama
merek">
                    </div>
                    <div class="form-group">
                        <input type="text" class="form-control form-
control-user" id="warna" name="warna" placeholder="Masukkan Warna">
                    </div>
                    <div class="form-group">

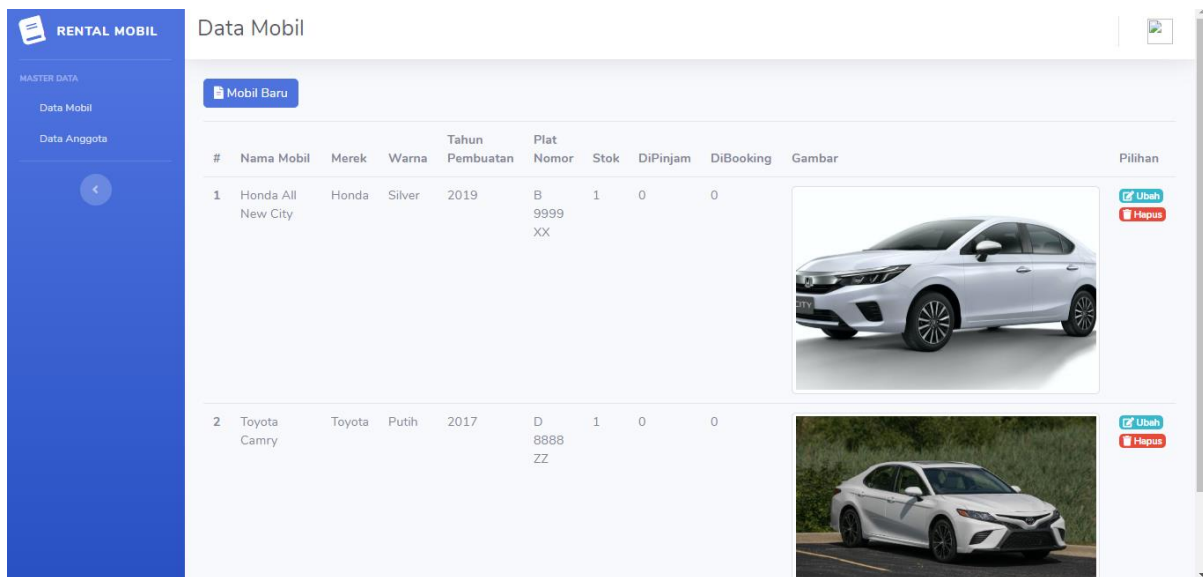
```

```

        <select name="tahun_pembuatan" class="form-
control form-control-user">
            <option value="">Pilih Tahun</option>
            <?php
            for ($i=date('Y'); $i > 2000 ; $i--) {
?>
                <option value="<?= $i;?>"><?=
$i;?></option>
            <?php } ?>
        </select>
    </div>
    <div class="form-group">
        <input type="text" class="form-control form-
control-user" id="plat_no" name="isbn" placeholder="Masukkan Plat
Nomor">
    </div>
    <div class="form-group">
        <input type="text" class="form-control form-
control-user" id="stok" name="stok" placeholder="Masukkan nominal
stok">
    </div>
    <div class="form-group">
        <input type="file" class="form-control form-
control-user" id="image" name="image">
    </div>
    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-secondary"
data-dismiss="modal"><i class="fas fa-ban"></i> Close</button>
        <button type="submit" class="btn btn-primary"><i
class="fas fa-plus-circle"></i> Tambah</button>
    </div>
</form>
</div>
</div>
</div>
<!-- End of Modal Tambah Mneu -->

```

Sekarang coba kita klik pada menu data mobil. Maka halaman akan diarahkan ke method/function index(). Dan data mobil pun berhasil ditampilkan.



## 12.2 Update Data Mobil

Untuk mengubah data mobil, akan kita buat sebuah form baru lagi pada function **ubahmobil()** di controller **Mobil.php**. seperti yang telah kita arahkan pada tombol ubah di data mobil.

```
public function ubahMobil()
{
    $data['judul'] = 'Ubah Data Mobil';
    $data['user'] = $this->ModelUser->cekData(['email' => $this->session->userdata('email')])>row_array();
    $data['admin'] = $this->ModelMobil->mobilWhere(['id' => $this->uri->segment(3)])>result_array();
    $kategori = $this->ModelMobil->joinKategoriMobil(['mobil.id' => $this->uri->segment(3)])>result_array();
    foreach ($kategori as $k) {
        $data['id'] = $k['id_kategori'];
        $data['k'] = $k['kategori'];
    }
    $data['kategori'] = $this->ModelMobil->getKategori()->result_array();

    $this->form_validation->set_rules('kode_mobil', 'Nama Mobil', 'required|min_length[3]', [
        'required' => 'Nama Mobil harus diisi',
        'min_length' => 'Nama Mobil terlalu pendek'
    ]);
}
```

```

        $this->form_validation->set_rules('id_kategori', 'Kategori',
        'required', [
            'required' => 'Nama kategori harus diisi',
        ]);
        $this->form_validation->set_rules('merek', 'Merek',
        'required|min_length[3]', [
            'required' => 'Merek harus diisi',
            'min_length' => 'Merek terlalu pendek'
        ]);
        $this->form_validation->set_rules('warna', 'Warna',
        'required|min_length[3]', [
            'required' => 'Warna harus diisi',
            'min_length' => 'Warna terlalu pendek'
        ]);
        $this->form_validation->set_rules('tahun_pembuatan', 'Tahun
        Pembuatan', 'required|min_length[3]|max_length[4]|numeric', [
            'required' => 'Tahun pembuatan harus diisi',
            'min_length' => 'Tahun pembuatan terlalu pendek',
            'max_length' => 'Tahun pembuatan terlalu panjang',
            'numeric' => 'Hanya boleh diisi angka'
        ]);
        $this->form_validation->set_rules('plat_no', 'Plat Nomor',
        'required|min_length[3]', [
            'required' => 'Plat Nomor harus diisi',
            'min_length' => 'Plat Nomor terlalu pendek'
        ]);
        $this->form_validation->set_rules('stok', 'Stok',
        'required|numeric', [
            'required' => 'Stok harus diisi',
            'numeric' => 'Yang anda masukan bukan angka'
        ]);

        //konfigurasi sebelum gambar diupload
        $config['upload_path'] = './assets/img/upload/';
        $config['allowed_types'] = 'jpg|png|jpeg';
        $config['max_size'] = '3000';
        $config['max_width'] = '1024';
        $config['max_height'] = '1000';
        $config['file_name'] = 'img' . time();

        //memuat atau memanggil library upload
        $this->load->library('upload', $config);

```

```

        if ($this->form_validation->run() == false) {
            $this->load->view('templates/header', $data);
            $this->load->view('templates/sidebar', $data);
            $this->load->view('templates/topbar', $data);
            $this->load->view('mobil/ubah_mobil', $data);
            $this->load->view('templates/footer');
        } else {
            if ($this->upload->do_upload('image')) {
                $image = $this->upload->data();
                unlink('assets/img/upload/' . $this->input-
>post('old_pict', TRUE));
                $gambar = $image['file_name'];
            } else {
                $gambar = $this->input->post('old_pict', TRUE);
            }

            $data = [
                'kode_mobil' => $this->input->post('kode_mobil',
true),
                'id_kategori' => $this->input->post('id_kategori',
true),
                'merek' => $this->input->post('merek', true),
                'warna' => $this->input->post('warna', true),
                'tahun_pembuatan' => $this->input-
>post('tahun_pembuatan', true),
                'plat_no' => $this->input->post('plat_no', true),
                'stok' => $this->input->post('stok', true),
                'image' => $gambar
            ];

            $this->ModelMobil->updateMobil($data, ['id' => $this-
>input->post('id')]);
            redirect('mobil');
        }
    }
}

```

Sekarang kita akan mencoba mengubah/mengedit salah satu data mobil. Klik pada tombol edit. Lalu ubah yang dibutuhkan kemudian simpan/update

### 12.3 Menghapus Data Mobil.

Sesuai dengan link yang sudah kita tentukan pada tombol hapus di table data mobil seperti di atas.

```
<a href="<?= base_url('mobil/hapusmobil/').$b['id'];?>"
onclick="return confirm('Kamu yakin akan menghapus <?= $judul.'
'.$b['kode_mobil'];?> ?');" class="badge badge-danger"><i class="fas
fa-trash"></i> Hapus</a>
```

Untuk aksi dari penghapusan data mobil kita akan membuatnya di function/method hapusmobil() dalam controller **Mobil.php**. Buat sebuah method baru dengan nama hapusmobil() di controller **Mobil.php**.

```
public function hapusMobil()
{
    $where = ['id' => $this->uri->segment(3)];
    $this->ModelMobil->hapusMobil($where);
    redirect('mobil');
}
```

Sekarang untuk mencobanya, bisa klik tombol hapus pada mobil yang ingin di hapus. Maka data mobil pun terhapus.

**PERTEMUAN 13-15**  
**PRESENTASI PROJEK**



**PERTEMUAN 16**  
**UJIAN AKHIR SEMESTER**

## DAFTAR PUSTAKA

Hadi, Diki Alfarabi. E-book Belajar Cepat CodeIgniter Lengkap dengan Studi Kasus Membuat Aplikasi Rental Mobil.

<https://www.tutorialpedia.net/tutorial-install-codeigniter-untuk-pemula/> diakses tanggal 17 juli 2018:13.30

Rahajo, Budi. Belajar Otodidak Framework CodeIgniter Teknik Pemrograman Web dengan PHP 7 dan Framework CodeIgniter 3. Bandung. Informatika