

MODUL PERKULIAHAN

---

# PEMROGRAMAN WEB II

---

Berbasis **framework**

*Tim Penyusun Modul*

## KATA PENGANTAR

Dengan menyebut nama Allah SWT yang Maha Pengasih lagi Maha Panyayang, Penulis panjatkan puja dan puji syukur atas kehadiran-Nya, yang telah melimpahkan rahmat, hidayah, dan inayah-Nya kepada penulis, sehingga penulis dapat menyelesaikan Modul “Web Programming II” ini.

Penulis menyadari sepenuhnya masih ada kekurangan baik dari segi susunan kalimat maupun tata bahasa. Oleh karena itu, dengan tangan terbuka penulis menerima segala saran dan kritik dari pembaca agar menjadi lebih baik untuk ke depannya.

Akhir kata penulis berharap modul ini dapat memberikan manfaat maupun inspirasi terhadap pembaca

## DAFTAR ISI

Kata Pengantar .....	ii
Daftar Isi .....	iii
 PERTEMUAN 1 PENGENALAN CODEIGNITER .....	 1
1.1 Framework Web.....	1
1.2 CodeIgniter.....	2
1.3 Instalasi Software .....	2
 PERTEMUAN 2 MODEL, VIEW, CONTROLLER .....	 7
2.1 Pengertian MVC .....	7
2.2 Menentukan Kontroler Standar (Default) .....	12
2.3 Menghilangkan Index.php Pada CodeIgniter.....	13
2.4 Aturan Merancang Model, View, dan Controller .....	15
2.5 Parsing Data ke View.....	17
 PERTEMUAN 3 MEMBUAT TEMPLATE WEB SEDERHANA.....	 18
3.1 Membuat Template Sederhana.....	18
3.2 Seting Base_url .....	22
3.3 Membuat Halaman Template Dinamis .....	24
 PERTEMUAN 4 MEMBUAT FORM VALIDASI PADA CODEIGNITER.....	 30
4.1 Form Validasi.....	30
4.2 Mekanisme Validasi Data .....	30
4.3 Memuat Librari Form Validasi .....	30
 PERTEMUAN 5 STUDI KASUS PERPUSTAKAAN.....	 35
5.1 Persiapan Membuat Aplikasi Perpustakaan .....	35
5.2 Seting Base_url Perpustakaan .....	35
5.3 Merancang dan Membuat Database Aplikasi perpustakaan .....	36
5.4 Membuat Koneksi Database CodeIgniter dengan MySQL.....	39
5.5 Membuat Model Terstruktur dan Efisien .....	40
5.6 Integrasi CodeIgniter dan Bootstrap .....	42
 PERTEMUAN 6 MEMBUAT FORM LOGIN, MD5 DAN PESAN NOTIFIKASI.....	 43
6.1 Membuat Halaman Login .....	43
6.2 Membuat Halaman Admin dan Logout.....	48
 PERTEMUAN 7 REVIEW MATERI .....	 62
 PERTEMUAN 8 UJIAN TENGAH SEMESTER.....	 63
 PERTEMUAN 9 MEMBUAT CRUD DATA BUKU .....	 64
9.1 Menampilkan Data Buku .....	64
9.2 Input Data Buku .....	66

9.3 Update Data Buku .....	69
9.4 Menghapus Data Buku .....	71
PERTEMUAN 10 MEMBUAT PROSES TRANSAKSI PEMINJAMAN .....	72
10.1 Menampilkan Data Transaksi .....	73
10.2 Membuat Proses Transaksi .....	76
10.3 Membuat Pembatalan Transaksi .....	79
10.4 Transaksi Selesai .....	80
PERTEMUAN 11-14 PRESENTASI PROJEK .....	84
DAFTAR PUSTAKA	



## **PERTEMUAN 1**

### **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.

#### **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. 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).

## **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 (notepad, notepad++, sublime text, atom, dll)

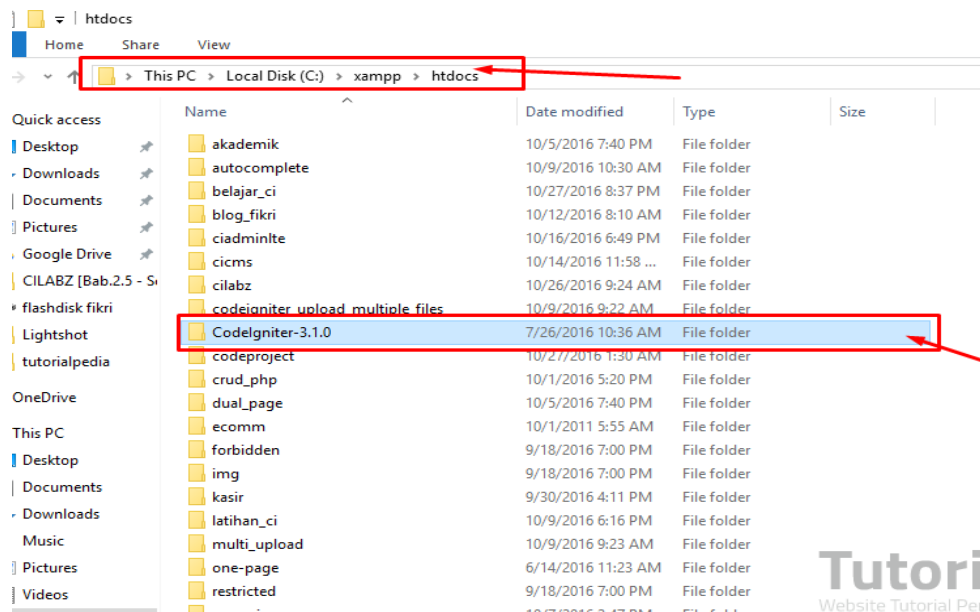
Pembelajaran kali ini akan menggunakan text editor Atom, karena bersifat Opensource. Bisa didownload di <https://atom.io/>

- c. Codeigniter <https://codeigniter.com/>

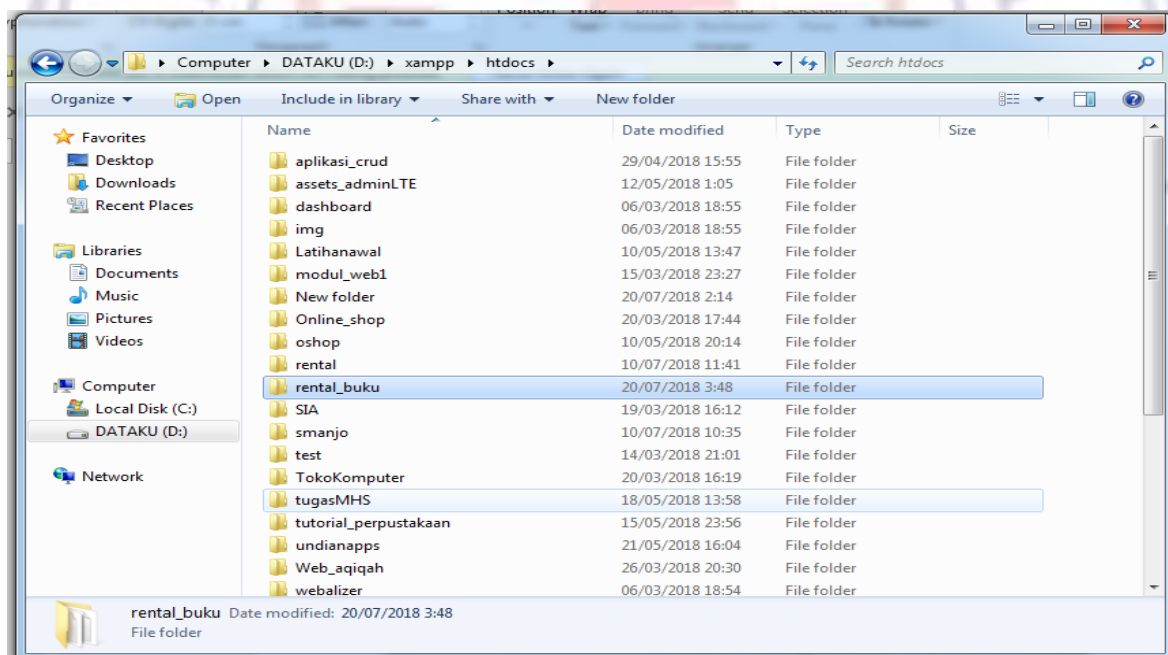
- d. Web Browser (Mozilla Firefox, google chrome, IE, Safari, UCBrowser, Opera, dll).

## **4. Instalasi Codeigniter**

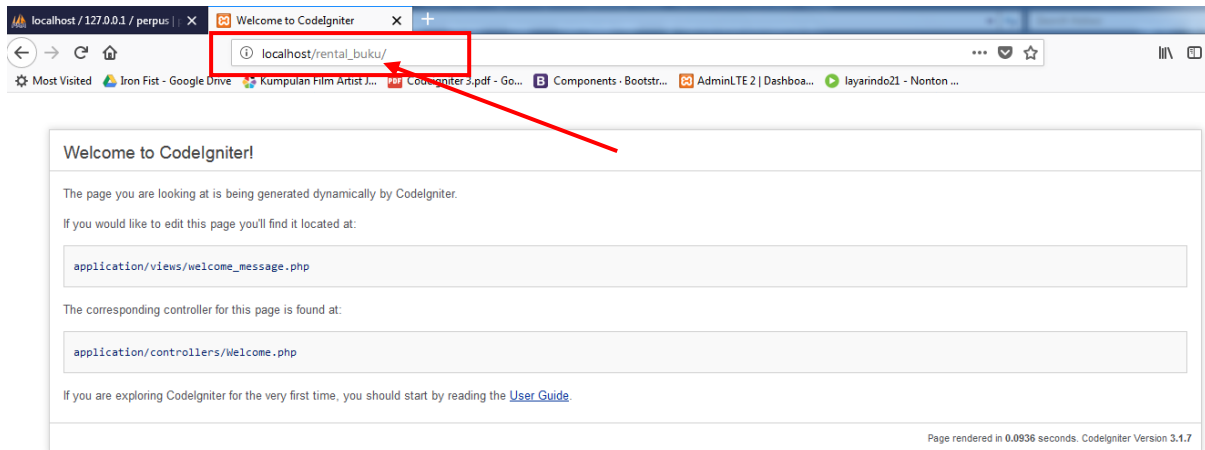
- a. Download Package codeigniter pada situ resminya yang sudah disebutkan di atas.
- b. Saat ini versi stabil dari codeigniter adalah 3.1.6
- c. Setelah di download ekstrak file codeigniter pada direktori **C:\xampp\htdocs**



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



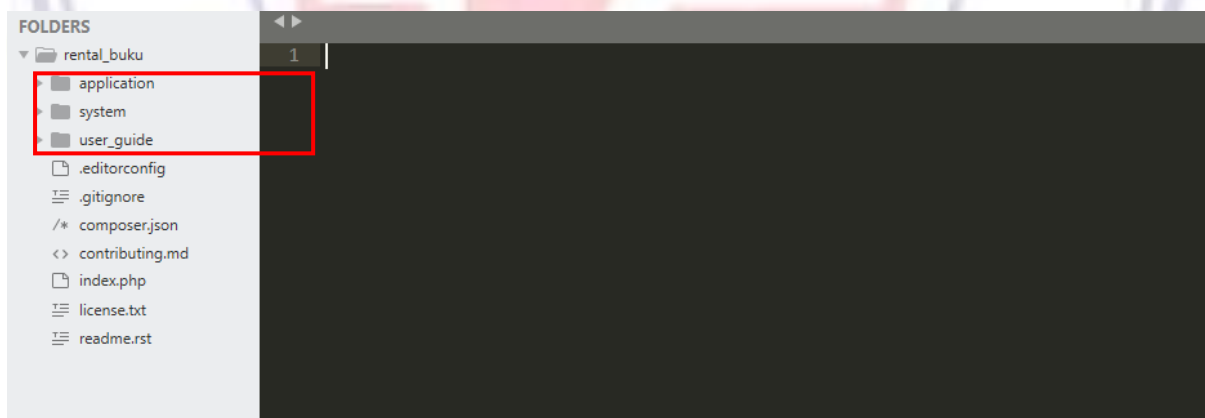
- e. Setelah itu, sekarang coba akses folder tersebut melalui browser.
- f. jangan lupa untuk menyalakan **XAMPP** terlebih dahulu.
- g. lalu akses dengan url **http://localhost/nama\_folder\_project** pada tutorial kali ini saya beri nama dengan **rental\_buku**



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

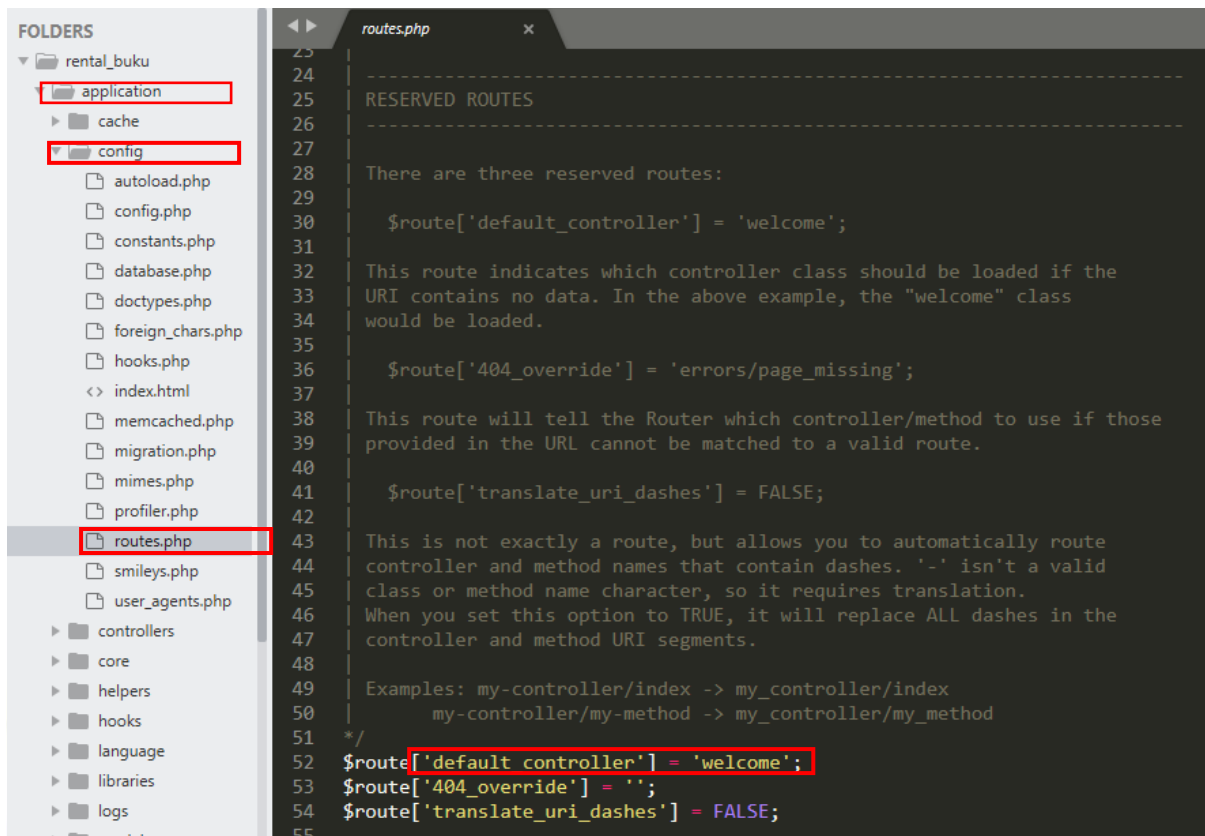
## 5. Struktur Folder Codeigniter

Di dalam folder codeigniter terdapat 3 folder utama yaitu **Application**, **System**, **User Guide**. Folder **application** berguna sebagai tempat untuk mendevlop 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).

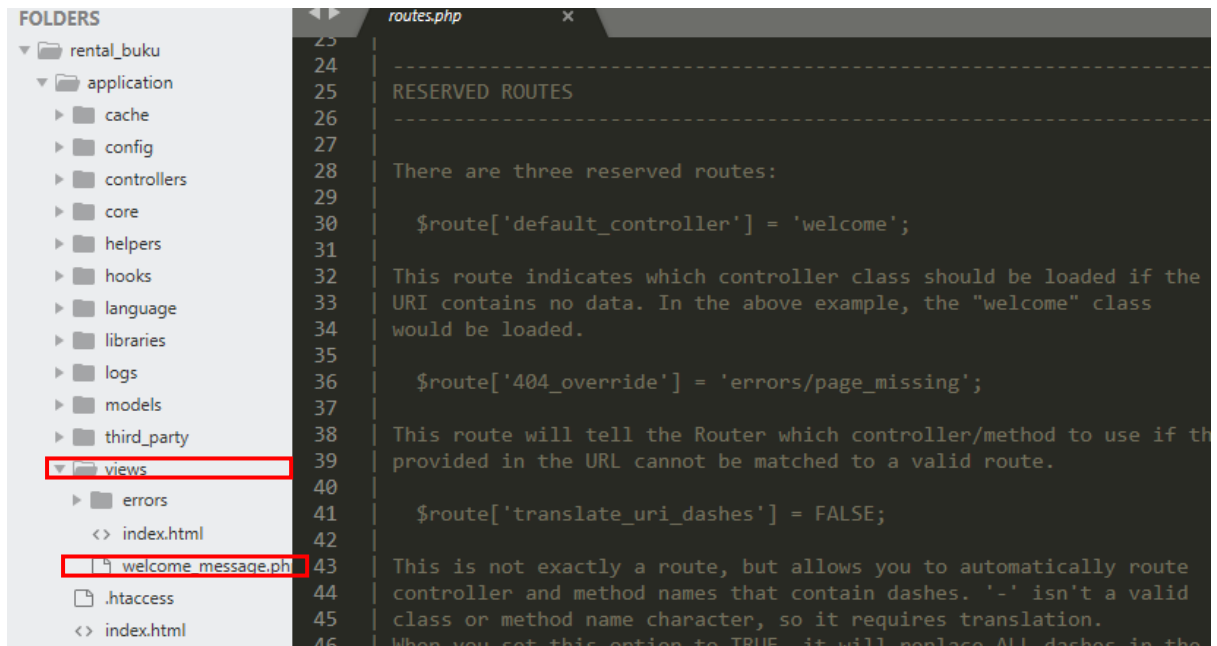




**Controller Welcome** itu sendiri bisa anda temukan pada folder **application/controllers**. Pada file **welcome\_controller** tersebut terdapat **fungsi index** yang gunanya **meload bagian view** atau memanggil halaman views **dengan nama welcome\_messages** sebagai berikut.

```
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.



## PERTEMUAN 2

### 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.

#### 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

```
<?php
class Hello1 extends CI_Controller {
    public function index() {
        echo "<h2> Hello World! (Versi 1) </h2>";
    }
}
```

## Contoh2 menggunakan Controller dan Model

Model simpan di application\model\Hello\_model.php

```
<?php
class Hello2_model extends CI_Model {
    //mendefinisikan properti dengan nama $str
    public $str = 'Hello World! (versi 2)';
}
```

Controller simpan di application\controllers\Hello2.php

```
class Hello2 extends CI_Controller {
    public function index() {
        //memuat model dengan nama Hello2_model
        $this->load->model('Hello2_model');

        //menangkap objek dari kelas Hello2_model yang telah dimuat ke variabel $model
        $model = $this->Hello2_model;

        echo "<html>
        <head><title>Controller dan Model</title></head>
        <body>";

        //memanggil properti $str milik objek $model
        echo "<h2>". $model->str."</h2>";

        echo "</body>";
        echo "</html>";
    }
}
```



### Contoh3 menggunakan Controller dan View

**View** Simpan di application\views\Hello3view.php

```
<html>
<head><title>Controller dan View</title></head>
<body>
<h2>Hello World! (Versi 3) </h2>
</body>
</html>
```

**Contoller** simpan di application\controllers\Hello3.php

```
<?php
class Hello3 extends CI_Controller{
    public function index() {
        //memuat view 'hello3view'
        $this->load->view('hello3view');
    }
}
```

Untuk melihat hasil dari contoh-contoh di atas bisa dengan ketik URI:

[http://localhost/rental\\_buku/index.php/hello1](http://localhost/rental_buku/index.php/hello1).

### Contoh4 menggunakan Controller, Model, dan View

**Model** Simpan di application\models\lingkaran\_model.php:

```
<?php
Class Lingkaran_model extends CI_model {
    //mendefinisikan konstanta untuk nilai PI
    const PI = 3.14;

    //atribut model
    private $jarijari;

    //metode untuk menentukan nilai $jarijari
    public function set_jarijari($r){
        $this->jarijari = $r;
    }
}
```

```

//metode untuk mengambil nilai $jarjari
public function get_jarijari(){
    return $this->jarijari;
}

//metode untuk menghitung luas lingkaran
public function hitung_luas(){
    return self::PI * $this->jarijari * $this->jarijari;
}

//metode untuk menghitung keliling lingkaran
public function hitung_keliling(){
    return 2 * self::PI * $this->jarijari;
}
}

```

**Controller** Simpan di application\controllers\lingkaran.php

```

<?php
Class Lingkaran extends CI_Controller {
    public function index() {
        //meload model Lingkaran_model
        $this->load->model('Lingkaran_model');

        //menangkap model yang telah dimuat oleh controller
        $model = $this->Lingkaran_model;

        //menentukan nilai jari jari
        $model->set_jarijari(3);

        //memuat view dan mengirimkan $model ke view
        $this->load->view('lingkaranview', array('model' => $model));
    }
}

```

**View** Simpan di application\views\lingkaranview.php

```

<html>
<head>
<title> Demo Model, View, Controller </title>

```

```

</head>
<body>
<h2> Model Lingkaran </h2>

<!-- memanggil metode get_jarijari() dari objek $model -->
Nilai Jari Jari : <?php echo $model->get_jarijari(); ?> <br>

<!-- memanggil metode hitung_luas() dari objek $model -->
Luas Lingkaran : <?php echo $model->hitung_luas(); ?> <br>

<!-- memanggil metode hitung_keliling() dari objek $model -->
Keliling Lingkaran : <?php echo $model->hitung_keliling(); ?>

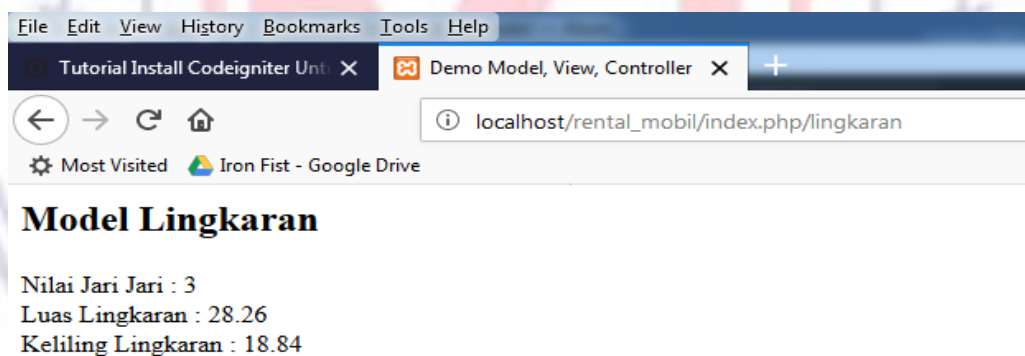
</body>
</html>

```

Jalankan aplikasi di atas menggunakan URI berikut:

[http://localhost/rental\\_buku/index.php/lingkaran](http://localhost/rental_buku/index.php/lingkaran)

hasil yang ditampilkan adalah seperti berikut:



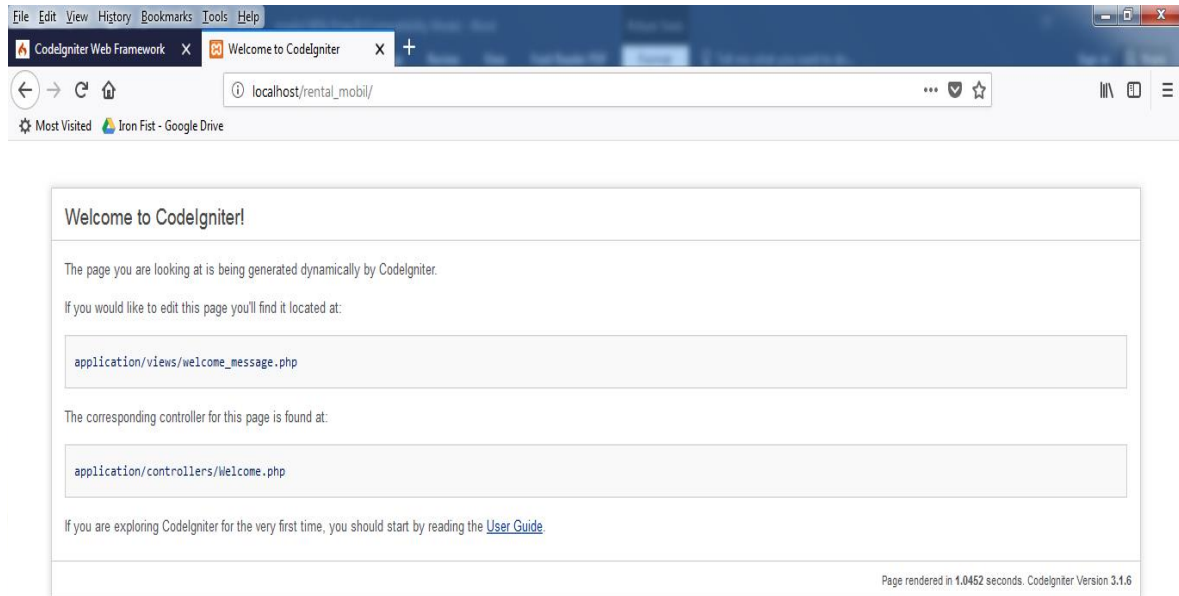
## 2. Menentukan Kontroler Standar (Default)

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\\_buku/index.php](http://localhost/rental_buku/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 `application\config\routes.php`. cari baris code berikut:

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

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

### 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 pengaturan “index\_page” codeigniter pada *application/config/config.php*, kemudian cari pengaturan 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\_buku/.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 akan dibuat sebuah method dengan nama pemrograman pada file controller Belajar.php yang sudah pernah dibuat.

**application/controllers/belajar.php**

```
1 <?php
2 class Belajar extends CI_Controller{
3     function __construct(){
4         parent::__construct();
5     }
6
7     public function index(){
8         echo "ini method index pada controller belajar | cara membuat controller pada codeigniter";
9     }
10
11     public function halo(){
12         $data['nama_web'] = 'Codeigniter.com';
13         $this->load->view('view_belajar', $data);
14     }
15
16     function pemrograman(){
17         echo 'Menghilangkan index.php pada codeigniter | RentalBuku';
18     }
19 }
```



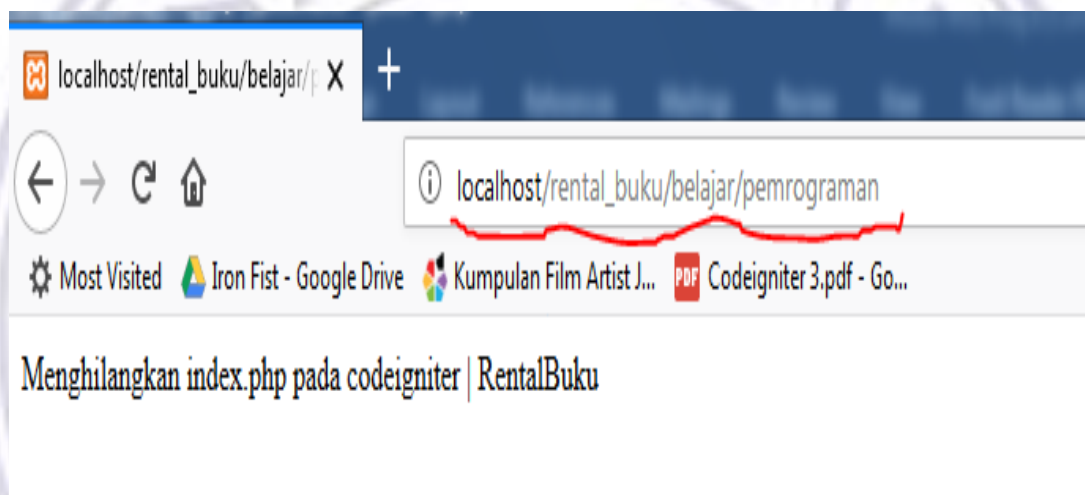
Jika biasanya untuk mengakses method “pemrograman” dalam kontroler belajar diatas harus dengan mengetikkan

```
http://localhost/rental_buku/index.php/belajar/pemrograman
```

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

```
http://localhost/rental_buku/belajar/pemrograman
```

berikut hasil dalam browser



#### 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 tampilan, 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.

## 5. Parsing Data Ke View

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

Perhatikan contoh cara memarsing data ke view codeigniter berikut ini. Buka controller belajar.

**Contoh5** parsing data dari controller ke view

**Controller** simpan di application\controllers\belajar.php

```
<?php
class Belajar extends CI_Controller {
    function __construct() {
        parent::__construct();
    }

    public function index() {
        echo "ini method index pada controller Belajar | cara membuat controller pada codeigniter";
    }

    public function halo() {
        //memparsing data ke view
        $data['nama_web'] = "Codeigniter.com";
        $this->load->view('view_belajar', $data);
    }
}
```

**View** simpan di application\views\view\_belajar.php

```
<!DOCTYPE html>
<html>
<head>
<title>Cara Membuat View Pada CodeIgniter </title>
</head>
<body>
<h1><?php echo $nama_web; ?></h1>
</body>
</html>
```

## PERTEMUAN 3

### 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.

#### 1. Membuat Template Sederhana

Untuk membuat dan menampilkan template sederhana yang akan dibuat, 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";

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

    }

}
```

Kemudian buatlah sebuah **v\_index.php** simpan pada **application/views/v\_index.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() ?&gt;assets/css/stylebuku.css"&gt;</pre
```

```

</head>
<body>
    <div id="wrapper">
        <header>
            <hgroup>
                <h1>RentalBuku.net</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>
        <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 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.</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>

<footer>

<a href="http://www.RentalBuku.com">RentalBuku</a>

</footer>

</div>

</body>

</html>

Kemudian buatlah **stylebuku.css** untuk membuat tampilan lebih bagus dan simpan di folder root seperti berikut: **rental\_buku/assets/css/stylebuku.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 3 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.

## 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_buku/';
```

Setingan 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/stylebuku.css">
```

Hasil perintah di atas akan sama seperti ini

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

Hal tersebut di atas dikarenakan `base_url()` sudah terlebih dahulu di seting menjadi “localhost/rental\_buku”. 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";
```

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

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

```
92 $autoload['helper'] = array();  
93
```

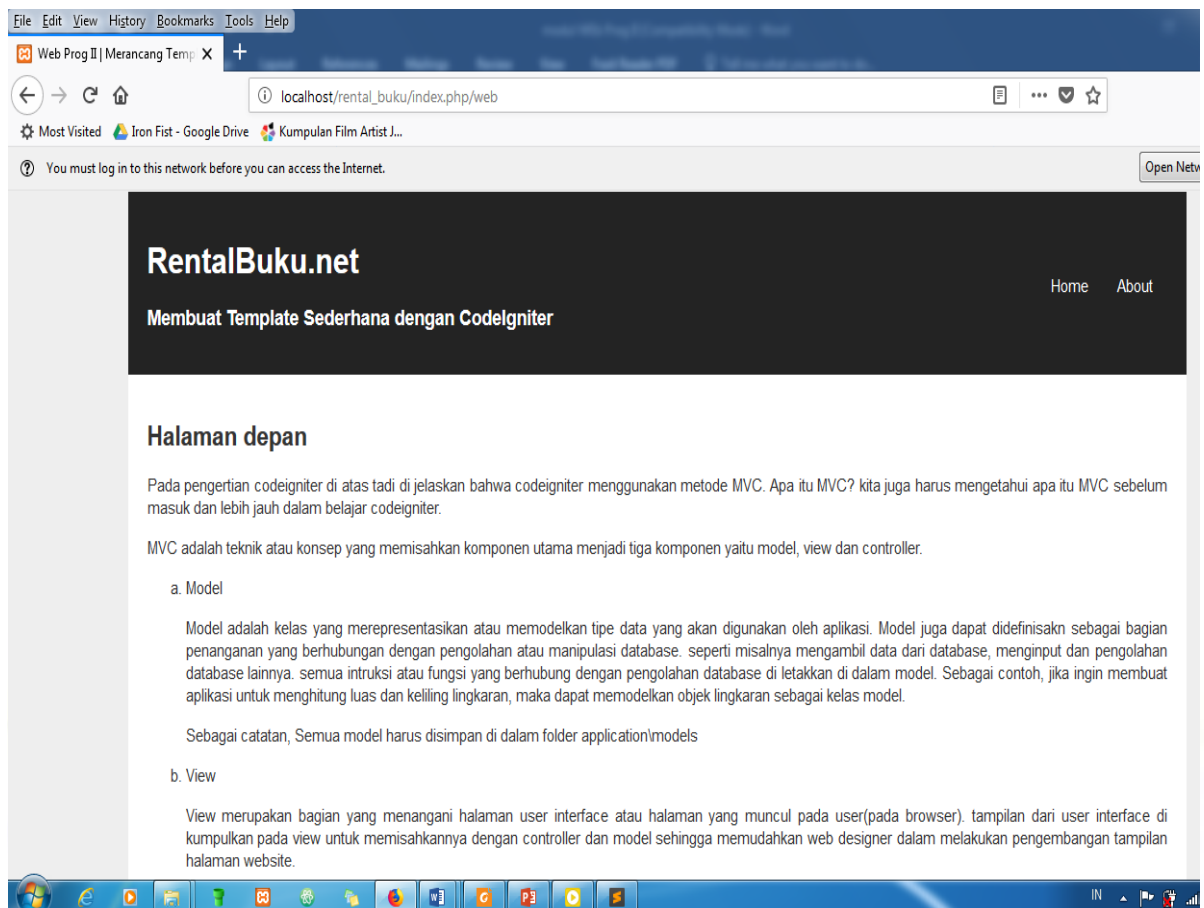
Kemudian diubah dengan ditambahkan kata 'url' menjadi seperti di bawah

```
91  
92 $autoload['helper'] = array('url');  
93
```

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

[http://localhost/rental\\_buku/index.php/web](http://localhost/rental_buku/index.php/web)

hasilnya seperti gambar di bawah.



Demikian tampilan web sederhana sudah berhasil dibuat pada codeigniter. Tapi halaman ini belum dinamis. Masih perlu membuat bagian header dan footer secara berulang-ulang pada halaman-halamn lainnya.

### 3. Membuat Halaman Template Dinamis

Pembuatan template web yang dinamis hanya cukup dengan cara memisah-misahkan file **v\_index.php** yang sudah dibuat tadi menjadi beberapa bagian file view. Yaitu akan dibuat menjadi tiga file view menjadi file **v\_header.php**, **v\_index.php**, **v\_footer.php**. file-file tersebut disimpan di tempat yang sama seperti file **v\_index.php** sebelumnya.

File **application/views/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() ?&gt;assets/css/stylebuku.css"&gt;

&lt;/head&gt;
&lt;body&gt;

    &lt;div id="wrapper"&gt;

        &lt;header&gt;

            &lt;hgroup&gt;

                &lt;h1&gt;RentalBuku.net&lt;/h1&gt;

                &lt;h3&gt;Membuat Template Sederhana dengan CodeIgniter&lt;/h3&gt;

            &lt;/hgroup&gt;

            &lt;nav&gt;

                &lt;ul&gt;

                    &lt;li&gt;&lt;a href="<?php echo base_url(). 'index.php/web' ?&gt;"&gt;Home&lt;/a&gt;&lt;/li&gt;

                    &lt;li&gt;&lt;a href="<?php echo base_url(). 'index.php/web/about'

?&gt;"&gt;About&lt;/a&gt;&lt;/li&gt;

                &lt;/ul&gt;

            &lt;/nav&gt;

            &lt;div class="clear"&gt;&lt;/div&gt;

        &lt;/header&gt;
</pre

```

Kemudian file **application/views/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 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.</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>

```

### File application/views/v\_footer.php

```

<footer>

        <a href="http://www.RentalBuku.com">RentalBuku</a>

</footer>

</div>

</body>

</html>

```

Setelah file dipecah menjadi tiga file, selanjutnya cara untuk memanggil file-file tersebut agar bisa tampil menyatu adalah dengan cara memanggilnya secara berurutan pada controller

### 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);
    }
}

```

Hasil yang ditampilkan akan sama seperti sebelum dipisah menjadi 3 file. Tetapi kelebihanannya adalah dapat dengan mudah untuk membuat halaman lainnya dengan tampilan header dan footer yang tetap sama. Tinggal memanggil **v\_header.php** dan **v\_footer.php** dan mengganti file **v\_index.php** dengan file halaman lain yang ingin ditampilkan dengan header dan footer tersebut.

Contoh untuk membuat halaman lainnya yaitu akan dibuat dalam **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 metode **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>
    <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
berhubung 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.</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>

Kemudian buat method baru yaitu method about() pada controller **Web.php**

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

class Web extends CI_Controller{

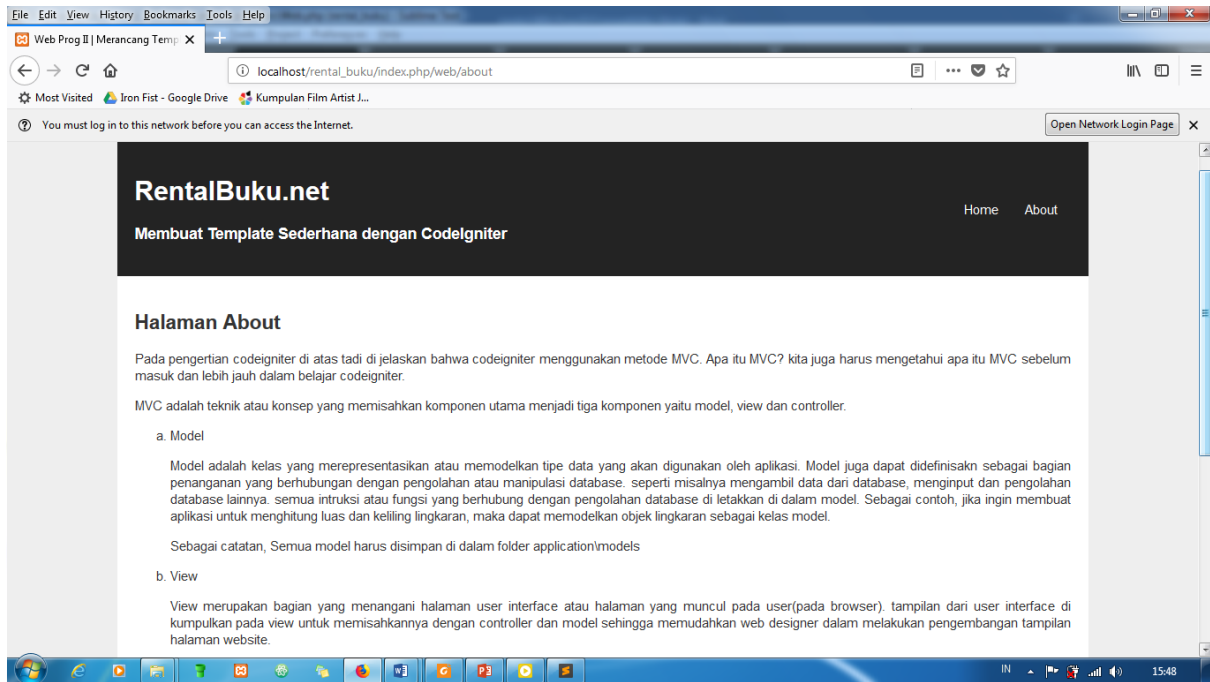
    function __construct(){
        parent::__construct();
    }

    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);
    }

    public function about(){
        $data['judul'] = "Halaman About";
        $this->load->view('v_header',$data);
        $this->load->view('v_index',$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_buku/index.php/web/about`



## **PERTEMUAN 4**

### **MEMBUAT FORM VALIDASI PADA CODEIGNITER**

#### **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. 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 pendefinisian aturan (rule) dari datanya.

#### **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:

```
61 $autoload['libraries'] = array();
```

Kemudian diubah dengan ditambahkan kata form\_validation seperti berikut:

```
61 $autoload['libraries'] = array('form_validation');
```

Contoh pembuatan form validasi kali ini akan membuat kontroler dengan nama **Form.php** simpan pada **application/controllers/Form.php**



```
1 <?php
2 define('BASEPATH') or exit ('No Direct Script Access Allowed');
3
4 class Form extends CI_Controller{
5
6     function __construct(){
7         parent::__construct();
8     }
9
10    public function index(){
11        $this->load->view('v_form');
12    }
13
14    public function aksi(){
15        $this->form_validation->set_rules('nama','Nama','required');
16        $this->form_validation->set_rules('email','Email','required');
17        $this->form_validation->set_rules('konfirmasi_email','Konfirmasi Email','required');
18
19        if($this->form_validation->run() != false){
20            echo "Form Validation Oke";
21        }else{
22            $this->load->view('v_form');
23        }
24    }
25 }
```

Berdasarkan kontroler Form.php di atas, maka perlu dibuat terlebih dahulu sebuah form pada view yang akan dipanggil pada method index agar dipanggil pertama kali.

**application/views/v\_form.php**

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Membuat Form Validation pada CodeIgniter | RentalBuku</title>
5 </head>
6 <body>
7     <table border="1" cellpadding="5" align="center">
8         <tr>
9             <td>
10                <h1>Membuat Form Validasi dengan CodeIgniter</h1>
11                <?php echo validation_errors(); ?>
12                <?php echo form_open('form/aksi'); ?>
13                <table border="0" cellspacing="0" cellpadding="5">
14                    <form>
15                        <tr>
16                            <td>
17                                <label>Nama</label>
18                            </td>
19                            <td>
20                                <input type="text" name="nama">
21                            </td>
22                        </tr>
23                        <tr>
24                            <td>
25                                <label>Email</label>
26                            </td>
27                            <td>
28                                <input type="text" name="email">
29                            </td>
30                        </tr>
31                        <tr>
32                            <td>
33                                <label>Konfirmasi Email</label>
34                            </td>
35                            <td>
36                                <input type="text" name="konfirm_email"><br>
37                            </td>
38                        </tr>
39                        <tr>
40                            <td colspan="2" align="right">
41                                <input type="submit" name="submit" value="Simpan">
42                            </td>
43                        </tr>
44                    </form>
45                </table>
46            </td>
47        </tr>
48    </table>
49 </body>
50 </html>

```

Untuk membuat form validation, sebelumnya harus menentukan dulu form yang akan di berikan validasi. Seperti contoh di atas telah membuat form validation pada form input nama, email dan konfir\_email. function set\_rules di atas berarti telah menetapkan peraturan untuk form. cara penulisannya pada parameter pertama berikan nama form yang ingin di beri validasi, pada parameter kedua berikan kata yang di munculkan pada saat validasi. dan parameter ketiga isikan peraturan form. required berarti wajib, berarti form tersebut wajib di isi. sekarang coba jalankan form validation yang sudah dibuat dengan alamat [http://localhost/rental\\_buku/form](http://localhost/rental_buku/form). lalu klik tombol submit jangan mengisi form nya maka akan muncul peringatan untuk mengisi form.

## Membuat Form Validasi dengan CodeIgniter

The Nama field is required.

The Email field is required.

The Konfirmasi Email field is required.

Nama	<input type="text"/>
Email	<input type="text"/>
Konfirmasi Email	<input type="text"/>
<input type="button" value="Simpan"/>	

letak pesan peringatan form validationnya di tampilkan dengan fungsi berikut pada view form.

```
<?php echo validation_errors(); ?>
```

dan jika ingin menampilkan pesan dengan manual anda dapat menggunakan fungsi berikut

```
<?php //echo validation_errors();  
echo form_error('nama');  
echo form_error('email');  
echo form_error('konfirmasi_email');  
?>
```

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



## **PERTEMUAN 5**

### **STUDI KASUS PERPUSATAKAAN**

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 perpustakaan berbasis web.

#### **1. Persiapan Membuat Aplikasi Perpustakaan**

Pada tahap persiapan ini 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 perpustakaan. Semua pengaturan atau penyesuaian yang dilakukan di sini sesuai dengan kebutuhan untuk membuat aplikasi perpustakaan dengan codeigniter. diantaranya adalah:

#### **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.

**perpustakaan/application/config/config.php**

Helper dan Library bisa dibilang seperti perpustakaan 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 perpustakaan ini. Untuk menjalankan atau membuka library dan helper secara otomatis di codeigniter, kita bisa mengaturnya dalam file `autoload.php` dalam folder `application/config/autoload.php`.

### **Perpustakaan/application/config/autoload.php**

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

Cari sintak

```
61 $autoload['libraries'] = array();
```

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

```
61 $autoload['libraries'] = array('database','session','form_validation');
```

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

```
92 $autoload['helper'] = array();
```

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

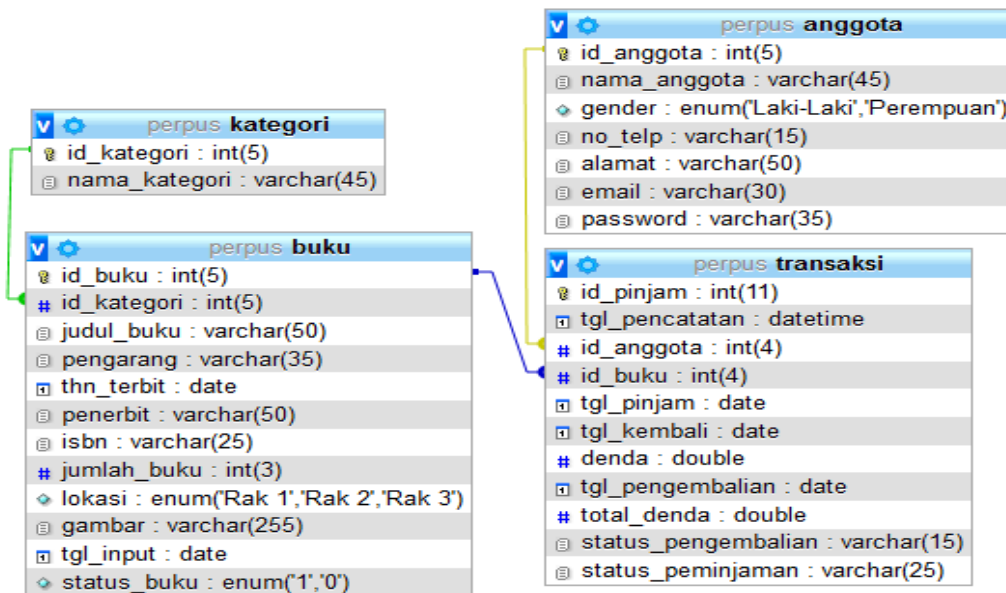
```
92 $autoload['helper'] = array('url');
```

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



### 3. Merancang & Membuat Database Aplikasi Perpustakaan dan Konfigurasi database di Codeigniter

Berikut adalah gambaran struktur database yang akan kita buat untuk aplikasi perpustakaan ini.



Buat sebuah database baru dengan nama “perpus”.

Ada 5 buah tabel yang akan kita buat dalam database dan 1 tabel tambahan. Yaitu:

1. Kategori
2. Buku
3. Anggota
4. Peminjaman
5. Detail\_pinjam
6. Admin
7. Transaksi

Buatlah sebuah database melalui localhost/phpmyadmin. Beri nama database dengan nama **perpus**. Kemudian buat tabel-tabelnya. Seperti berikut:

#### Admin

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra	Tindakan
1	id_admin	int(5)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Kunci Utama Unik Lainnya
2	nama_admin	varchar(25)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Kunci Utama Unik Lainnya
3	username	varchar(25)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Kunci Utama Unik Lainnya
4	password	varchar(35)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Kunci Utama Unik Lainnya

## kategori

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	<b>id_kategori</b>	int(5)		Tidak	Tidak ada		AUTO_INCREMENT	Ubah  Hapus  Kunci Utama  Unik  Lainnya
<input type="checkbox"/>	2	<b>nama_kategori</b>	varchar(45) latin1_swedish_ci		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Unik  Lainnya

## buku

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	<b>id_buku</b>	int(5)		Tidak	Tidak ada		AUTO_INCREMENT	Ubah  Hapus  Kunci Utama  Unik  Lainnya
<input type="checkbox"/>	2	<b>id_kategori</b>	int(5)		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Unik  Lainnya
<input type="checkbox"/>	3	<b>judul_buku</b>	varchar(50) latin1_swedish_ci		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Unik  Lainnya
<input type="checkbox"/>	4	<b>pengarang</b>	varchar(35) latin1_swedish_ci		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Unik  Lainnya
<input type="checkbox"/>	5	<b>thn_terbit</b>	date		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Unik  Lainnya
<input type="checkbox"/>	6	<b>penerbit</b>	varchar(50) latin1_swedish_ci		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Unik  Lainnya
<input type="checkbox"/>	7	<b>isbn</b>	varchar(25) latin1_swedish_ci		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Unik  Lainnya
<input type="checkbox"/>	8	<b>jumlah_buku</b>	int(3)		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Unik  Lainnya
<input type="checkbox"/>	9	<b>lokasi</b>	enum('Rak 1', 'Rak 2', 'Rak 3') latin1_swedish_ci		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Unik  Lainnya
<input type="checkbox"/>	10	<b>gambar</b>	varchar(255) latin1_swedish_ci		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Unik  Lainnya
<input type="checkbox"/>	11	<b>tgl_input</b>	date		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Unik  Lainnya
<input type="checkbox"/>	12	<b>status_buku</b>	enum('1', '0') latin1_swedish_ci		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Unik  Lainnya

## anggota

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	<b>id_anggota</b>	int(5)		Tidak	Tidak ada		AUTO_INCREMENT	Ubah  Hapus  Kunci Utama  Unik  Lainnya
<input type="checkbox"/>	2	<b>nama_anggota</b>	varchar(45) latin1_swedish_ci		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Unik  Lainnya
<input type="checkbox"/>	3	<b>gender</b>	enum('Laki-Laki', 'Perempuan') latin1_swedish_ci		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Unik  Lainnya
<input type="checkbox"/>	4	<b>no_telp</b>	varchar(15) latin1_swedish_ci		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Unik  Lainnya
<input type="checkbox"/>	5	<b>alamat</b>	varchar(50) latin1_swedish_ci		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Unik  Lainnya
<input type="checkbox"/>	6	<b>email</b>	varchar(30) latin1_swedish_ci		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Unik  Lainnya
<input type="checkbox"/>	7	<b>password</b>	varchar(35) latin1_swedish_ci		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Unik  Lainnya

## transaksi

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	<b>id_pinjam</b>	int(11)		Tidak	Tidak ada		AUTO_INCREMENT	Ubah  Hapus  Kunci Utama  Lainnya
<input type="checkbox"/>	2	<b>tgl_pencatatan</b>	datetime		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Lainnya
<input type="checkbox"/>	3	<b>id_anggota</b>	int(4)		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Lainnya
<input type="checkbox"/>	4	<b>id_buku</b>	int(4)		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Lainnya
<input type="checkbox"/>	5	<b>tgl_pinjam</b>	date		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Lainnya
<input type="checkbox"/>	6	<b>tgl_kembali</b>	date		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Lainnya
<input type="checkbox"/>	7	<b>denda</b>	double		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Lainnya
<input type="checkbox"/>	8	<b>tgl_pengembalian</b>	date		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Lainnya
<input type="checkbox"/>	9	<b>total_denda</b>	double		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Lainnya
<input type="checkbox"/>	10	<b>status_pengembalian</b>	varchar(15) latin1_swedish_ci		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Lainnya
<input type="checkbox"/>	11	<b>status_peminjaman</b>	varchar(25) latin1_swedish_ci		Tidak	Tidak ada			Ubah  Hapus  Kunci Utama  Lainnya

#### 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:

```
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 ▼ $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost',
79     'username' => '',
80     'password' => '',
81     'database' => '',
82     'dbdriver' => 'mysqli',
83     'dbprefix' => '',
84     'pconnect' => FALSE,
85     'db_debug' => (ENVIRONMENT !== 'production'),
86     'cache_on' => FALSE,
87     'cachedir' => '',
88     'char_set' => 'utf8',
89     'dbcollat' => 'utf8_general_ci',
90     'swap_pre' => '',
91     'encrypt' => FALSE,
92     'compress' => FALSE,
93     'stricton' => FALSE,
94     'failover' => array(),
95     'save_queries' => TRUE
96 );
97
```

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

```

73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost',
79     'username' => 'root',
80     'password' => '',
81     'database' => 'perpus',
82     'dbdriver' => 'mysqli',
83     'dbprefix' => '',
84     'pconnect' => FALSE,
85     'db_debug' => (ENVIRONMENT !== 'production'),
86     'cache_on' => FALSE,
87     'cachedir' => '',
88     'char_set' => 'utf8',
89     'dbcollat' => 'utf8_general_ci',
90     'swap_pre' => '',
91     'encrypt' => FALSE,
92     'compress' => FALSE,
93     'stricton' => FALSE,
94     'failover' => array(),
95     'save_queries' => TRUE
96 );
97

```

## 5. Membuat Model Yang Terstruktur dan Efisien untuk CRUD

Pada tahap ini akan dibuat fungsi-fungsi yang bisa gunakan 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 sebuah file model dengan nama M\_perpus.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.

```

1  <?php
2  defined('BASEPATH') or exit ('No Direct Script Access Allowed');
3
4  class M_perpus extends CI_Model{
5      function edit_data($where,$table){
6          return $this->db->get_where($table,$where);
7      }
8
9      function get_data($table){
10         return $this->db->get($table);
11     }
12
13     function insert_data($data,$table){
14         $this->db->insert($table,$data);
15     }
16
17     function update_data($where,$data,$table){
18         $this->db->where($where);
19         $this->db->update($table,$data);
20     }
21
22     function delete_data($where,$table){
23         $this->db->where($where);
24         $this->db->delete($table);
25     }
26 }

```

Dengan model seperti di atas, kita bisa menggunakannya secara berulang-ulang tanpa membuat 1 function model untuk 1 function pengelolaan database. Untuk menginput data ke database kita bisa menggunakan function *insert\_data()*, untuk menampilkan data atau mengambil data dari database kita bisa menggunakan function *get\_data()*. Untuk mengambil data tertentu bisa menggunakan *edit\_data()*, untuk menghapus data dari database kita bisa menggunakan function *delete\_data()*, dan untuk mengupdate data pada database kita bisa menggunakan function *update\_data()*.

Kemudian kita harus mengatur autoload model di application/config/autoload.php. agar model *m\_rental* di load secara otomatis. Jadi tidak perlu lagi kita load dari controller.

#### Application/config/autoload.php

Cari baris sintak berikut:

```

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

```

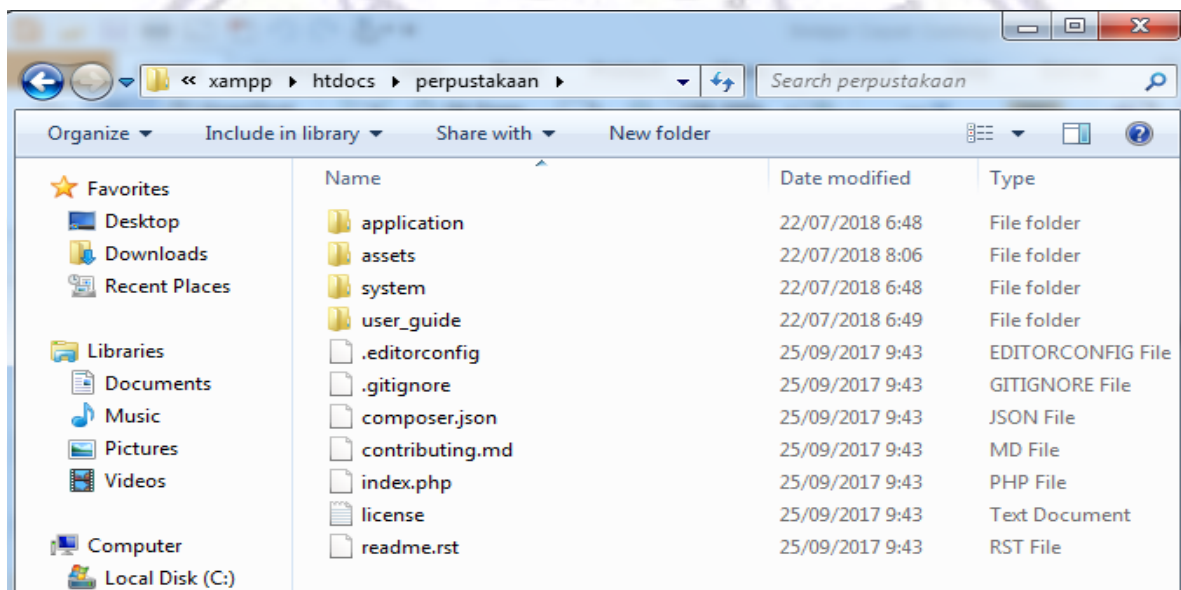


Kemudian ubah menjadi

```
135 $autoload['model'] = array('m_perpus');
```

## 6. Integrasi CodeIgniter dan Bootstrap

Untuk bisa menggunakan bootstrap, kita bisa mendownload bootstrap ini dari situs resminya di <http://getbootstrap.com/>, atau bisa menggunakan bootstrap yang telah di sertakan di dalam folder pendukung. ekstrak bootstrap ke directory root project perpustakaan ini. Lalu rename nama folder menjadi “assets”.





## PERTEMUAN 6

### MEMBUAT FORM LOGIN, MD5 DAN PESAN NOTIFIKASI

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

#### 1. Membuat Halaman Login

Halaman login akan kita buat di halaman utama pada saat aplikasi perpustakaan ini diakses. sebagai percobaan, kita harus menginput data username dan password admin terlebih dulu di tabel admin. Caranya bisa klik **tambahkan** pada tabel admin. Kemudian isi username dengan “nama kamu”, password dengan “admin123”. Jangan lupa pilih fungsi **md5** pada password. Seperti contoh pada gambar berikut ini:



id_admin	int(5)	<input type="text"/>	<input type="text"/>
nama_admin	varchar(25)	<input type="text"/>	<input type="text" value="Administrator"/>
username	varchar(25)	<input type="text"/>	<input type="text" value="admin"/>
password	varchar(35)	<div>MD5 <input type="text"/></div>	<input type="text" value="1234"/>

Kemudian klik *irim* untuk menyimpan data. Oke selesai, akun admin sudah kita buat, selanjutnya kita buat sistem loginnya.

Form login akan kita buat pada controller Welcome.php. karena controller Welcome.php adalah controller yang default di gunakan pada codeigniter. Untuk mengubah default controllernya teman-teman bisa mengubahnya di *application/config/routes.php*. tapi di sini kita tidak akan mengubahnya, kita akan langsung menggunakan controller *Welcome.php* sebagai controller default. Halaman login pun akan kita buat di controller ini. Buka Controller welcome.php dan tuliskan syntax berikut.

## Application/controller/Welcome.php

```
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Welcome extends CI_Controller {
5     function __construct(){
6         parent::__construct();
7     }
8
9     public function index(){
10         $this->load->view('login');
11     }
12
13     function login(){
14         $username = $this->input->post('admin_username');
15         $password = $this->input->post('admin_password');
16         $this->form_validation->set_rules('admin_username','Username','trim|required');
17         $this->form_validation->set_rules('admin_password','Password','trim|required');
18
19         if($this->form_validation->run() != false){
20             $where = array('username' => $username, 'password' => md5($password) );
21
22             $data = $this->m_perpus->edit_data($where, 'admin');
23             $d = $this->m_perpus->edit_data($where, 'admin')->row();
24             $cek = $data->num_rows();
25
26             if($cek > 0){
27                 $session = array('id' => $d->id_admin, 'nama' => $d->nama_admin, 'status' => 'login');
28                 $this->session->set_userdata($session);
29                 redirect(base_url(). 'admin');
30             }else{
31                 redirect(base_url(). 'welcome?pesan=gagal');
32             }
33         }else{
34             $this->load->view('login');
35         }
36     }
37 }
38
```

Setelah selesai menulis syntax di atas pada controller Welcome.php, akan sedikit dijelaskan tentang function construct(), function construct seperti contoh di atas adalah function yang pertama kali akan di jalankan pada saat sebuah class dijalankan. Setelah function construct() di jalankan, selanjutnya yang di jalankan adalah function index(). Ini merupakan ketentuan dari codeigniter. Coba perhatikan pada function construct() di atas, pada function construct() ada pemanggilan model 'm\_perpus'. kenapa kita perlu memanggil model ini? karena kita akan berhubungan dengan model m\_perpus di dalam controller Welcome ini. Selanjutnya perhatikan juga pada function index() di atas.

```
9     public function index(){
10         $this->load->view('login');
11     }
```

Pada function index() kita membuat perintah untuk membuka view login. Syntax `$this->load->view()` adalah perintah untuk menampilkan view (tampilan). Di sini kita memerintahkan untuk menampilkan view login (login.php). ekstensi .php nya tidak perlu lagi kita tuliskan saat memanggil view. Cukup seperti perintah di atas.

Pada saat controller welcome dijalankan, otomatis function index juga dijalankan, dan akan menampilkan halaman login (view login.php). Karena pada function index kita ingin menampilkan view login.php, maka sekarang kita buat dulu view *login.php* nya.

*Application/view/login.php*

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Login - Aplikasi Perpustakaan Berbasis WEB</title>
5 <link rel="stylesheet" type="text/css" href="<?php echo base_url(). 'assets/css/bootstrap.css' ?>">
6 <script type="text/javascript" src="<?php echo base_url(). 'assets/js/jquery.js'; ?>"></script>
7 <script type="text/javascript" src="<?php echo base_url(). 'assets/js/bootstrap.js'; ?>"></script>
8 </head>
9 <body>
10 <div class="col-md-4 col-md-offset-4" style="margin-top:50px">
11 <center>
12 <h2>APLIKASI PERPUSTAKAAN</h2>
13 <h3>LOGIN</h3>
14 </center>
15 <br/>
16 <?php
17 if(isset($_GET['pesan'])){
18     if($_GET['pesan'] == "gagal"){
19         echo "<div class='alert alert-danger alert-danger'>";
20         echo $this->session->flashdata('alert');
21         echo "</div>";
22     }else if($_GET['pesan'] == "logout"){
23         if($this->session->flashdata())
24         {
25             echo "<div class='alert alert-danger alert-success'>";
26             echo $this->session->flashdata('Anda Telah Logout');
27             echo "</div>";
28         }
29     }else if($_GET['pesan'] == "belumlogin"){
30         if($this->session->flashdata())
31         {
32             echo "<div class='alert alert-danger alert-primary'>";
33             echo $this->session->flashdata('alert');
34             echo "</div>";
35         }
36         //echo "<div class='alert alert-primary'>Silahkan login dulu.</div>";
37     }
38 }
39 }else{
40     if($this->session->flashdata())
41     {
42         echo "<div class='alert alert-danger alert-message'>";
43         echo $this->session->flashdata('alert');
44         echo "</div>";
45     }
46 }
47 ?>
48 <br/>
49 <div class="panel panel-default">
50 <div class="panel-body">
51 <br/>
52 <br/>
53 <form method="post" action="<?php echo base_url(). 'welcome/login' ?>">
54 <div class="form-group">
55 <input type="text" name="admin_username" placeholder="username" class="form-control">
56 <?php echo form_error('username'); ?>
57 </div>
58
59 <div class="form-group">
60 <input type="password" name="admin_password" placeholder="password" class="form-control">
61 <?php echo form_error('password'); ?>
62 </div>
```

```

63
64         <div class="form-group">
65             <input type="submit" value="Login" class="btn btn-primary">
66         </div>
67     </form>
68     <br/>
69     <br/>
70 </div>
71 </div>
72 </div>
73 <script type="text/javascript">
74     $('.alert-message').alert().delay(3000).slideUp('slow');
75 </script>
76 </body>
77 </html>
78

```

Perhatikan pada tag <head> ... </head> di atas, Disinilah terlihat kegunaan dari `base_url()` yang sudah di jelaskan di atas tadi.

```

5     <link rel="stylesheet" type="text/css" href="<?php echo base_url(). 'assets/css/bootstrap.css' ?>">
6     <script type="text/javascript" src="<?php echo base_url(). 'assets/js/jquery.js'; ?>"></script>
7     <script type="text/javascript" src="<?php echo base_url(). 'assets/js/bootstrap.js'; ?>"></script>

```

Pada pengaturan `config['base_url']` sebelumnya kita telah menentukannya dengan `http://localhost/perpustakaan/`. Jadi saat kita gunakan seperti contoh menghubungkan template dengan bootstrap seperti di atas, sama seperti

```

<link rel="stylesheet" type="text/css" href="http://localhost/perpustakaan/assets/css/bootstrap.css" ?>

```

Karena `base_url()` sudah sama dengan `http://localhost/perpustakaan/`. Kemudian lagi perhatikan pada form yang sudah kita buat pada view `login.php` di atas.

```

<form method="post" action="<?php echo base_url(). 'welcome/login' ?>">
    <div class="form-group">
        <input type="text" name="admin_username" placeholder="username" class="form-control">
        <?php echo form_error('username'); ?>
    </div>

    <div class="form-group">
        <input type="password" name="admin_password" placeholder="password" class="form-control">
        <?php echo form_error('password'); ?>
    </div>

    <div class="form-group">
        <input type="submit" value="Login" class="btn btn-primary">
    </div>
</form>

```

Action dari form ini telah diatur agar ke function/method `login` di controller `welcome`.

method/function *login* pada controller welcome yang telah dibuat. yang akan di eksekusi sebagai **form handler** dari form login.

```
public function login(){
    $username = $this->input->post('admin_username');
    $password = $this->input->post('admin_password');
    $this->form_validation->set_rules('admin_username', 'Username', 'trim|required');
    $this->form_validation->set_rules('admin_password', 'Password', 'trim|required');

    if($this->form_validation->run() != false){
        $where = array('username' => $username, 'password' => md5($password) );

        $data = $this->m_perpus->edit_data($where, 'admin');
        $d = $this->m_perpus->edit_data($where, 'admin')->row();
        $cek = $data->num_rows();

        if($cek > 0){
            $session = array('id' => $d->id_admin, 'nama' => $d->nama_admin, 'status' => 'login');
            $this->session->set_userdata($session);
            redirect(base_url().'admin');
        }else{
            $this->session->set_flashdata('alert', 'Login gagal! Username atau password salah.');
```

Pertama kita menangkap data yang dikirim dari form. Dan kita tetapkan validasinya, bahwa username dan password harus wajib di isi (required), kemudian kita cek validasinya. Jika tidak ada masalah, kita cek username dan password yang diinput sesuai atau tidak dengan username dan password pada tabel admin. Jika sesuai maka kita buat session id, dan nama. Yang mana session id kita simpan id admin yang melakukan login, session nama kita menyimpan nama admin yang login. Setelah dibuat sessionnya, maka selanjutnya mengalihkan halamannya ke controller admin. Karena controller admin ini yang akan dijadikan sebagai halaman admin.

```
redirect(base_url().'admin');
```

Jika login gagal maka dialihkan ke controller welcome kembali. Yang mana halaman login ada di controller welcome. Sambil mengirimkan pesan melalui url untuk membuat pesan notifikasi bahwa login gagal.

```
$this->session->set_flashdata('alert', 'Login gagal! Username atau password salah.');
```



view *login.php* yang sudah dibuat. Di sana telah dibuat pesan notifikasi. Jika terdapat data pesan yang dikirim dari url, dan jika pesannya adalah “gagal”, maka akan di tampilkan pesan notifikasi “ login gagal! Username dan password salah ”.

```
<?php
if(isset($_GET['pesan'])){
    if($_GET['pesan'] == "gagal"){
        echo "<div class='alert alert-danger alert-danger'>";
        echo $this->session->flashdata('alert');
        echo "</div>";
    }else if($_GET['pesan'] == "logout"){
        if($this->session->flashdata())
        {
            echo "<div class='alert alert-danger alert-success'>";
            echo $this->session->flashdata('Anda Telah Logout');
            echo "</div>";
        }
        //echo "<div class='alert alert-success'>Anda telah logout.</div>";
    }else if($_GET['pesan'] == "belumlogin"){
        if($this->session->flashdata())
        {
            echo "<div class='alert alert-danger alert-primary'>";
            echo $this->session->flashdata('alert');
            echo "</div>";
        }
        //echo "<div class='alert alert-primary'>Silahkan login dulu.</div>";
    }
}
else{
    if($this->session->flashdata())
    {
        echo "<div class='alert alert-danger alert-message'>";
        echo $this->session->flashdata('alert');
        echo "</div>";
    }
}
?>
```

Jika pesannya adalah logout, maka akan di tampilkan “anda telah logout“. Jika pesannya adalah “belumlogin”. Maka akan di tampilkan notifikasi “silahkan login dulu”. Sampai di sini, sistem login belum rampung 100%, karena belum menyiapkan halaman admin. Oleh sebab itu kita akan membuat dan menyiapkan halaman admin dulu. Baru setelah itu kita coba apakah login sudah berjalan sebagai mana yang kita perintahkan atau belum.

## 2. Membuat Halaman Admin dan logout

### a. Membuat Halaman Admin

Jika login berhasil, halaman akan dialihkan ke controller admin. Maka dari itu harus disiapkan dulu controller admin nya. Buat sebuah controller baru dengan nama Admin.php  
*Application/controller/Admin.php*



```

Admin.php
1 <?php
2 defined('BASEPATH') or exit ('NO Direct Script Access Allowed');
3
4 class Admin extends CI_Controller{
5     function __construct(){
6         parent::__construct();
7         // cek login
8         if($this->session->userdata('status') != "login"){
9             redirect(base_url().'welcome?pesan=belumlogin');
10        }
11    }
12
13    function index(){
14        $data['peminjaman'] = $this->db->query("select * from peminjaman order by id_pinjam desc limit 10")->result();
15        $data['anggota'] = $this->db->query("select * from anggota order by id_anggota desc limit 10")->result();
16        $data['buku'] = $this->db->query("select * from buku order by id_buku desc limit 10")->result();
17
18        $this->load->view('admin/header');
19        $this->load->view('admin/index',$data);
20        $this->load->view('admin/footer');
21    }
22 }
23

```

Pada function construct() kita membuat pengecekan jika si admin belum login, maka akan di alihkan ke halaman login. Ini bertujuan untuk melindungi controller admin dari pengguna yang belum melakukan login.

```

if($this->session->userdata('status') != "login"){
    redirect(base_url().'welcome?pesan=belumlogin');
}

```

Dan pada function index() di controller admin, kita akan membuat halaman dashboard.

```

function index(){
    $data['peminjaman'] = $this->db->query("select * from peminjaman order by id_pinjam desc limit 10")->result();
    $data['anggota'] = $this->db->query("select * from anggota order by id_anggota desc limit 10")->result();
    $data['buku'] = $this->db->query("select * from buku order by id_buku desc limit 10")->result();

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

```

Pada function index() kita menampilkan view header, index dan footer. Yang mana pada view index kita memparsing data yang kita ambil dari database.

```

$data['peminjaman'] = $this->db->query("select * from peminjaman order by id_pinjam desc limit 10")->result();
$data['anggota'] = $this->db->query("select * from anggota order by id_anggota desc limit 10")->result();
$data['buku'] = $this->db->query("select * from buku order by id_buku desc limit 10")->result();

```

\$this->db->query() adalah fungsi untuk menjalankan query database di codeigniter. bisa dibaca panduannya pada user\_guide codeigniter.

[http://localhost/perpustakaan/user\\_guide/database/queries.html](http://localhost/perpustakaan/user_guide/database/queries.html)

Disarankan untuk membaca tutorial membuat template web sederhana dengan codeigniter dan tutorial cara memarsing data ke view codeigniter terlebih dulu sebelum melanjutkan ke depan. Tutorial membuat tampilan atau template web sederhana dengan codeigniter dan cara memarsing data ke view sudah dijelaskan pada pertemuan sebelumnya. Pada variabel array peminjaman di atas bertujuan untuk mengambil data transaksi dari tabel peminjaman sebanyak 10 record. Untuk di tampilkan di halaman admin. Variabel array anggota kita buat untuk mengambil 10 record data kostumer terakhir, dan variabel array buku kita buat untuk mengambil 10 record terakhir data buku dari tabel buku. Semua data yang kita ambil ini akan kita tampilkan pada view index (halaman dashboard admin). Buat sebuah folder dengan nama 'admin' di dalam folder view. kemudian buat 3 buah view baru dengan nama index.php, header.php dan footer.php di dalam folder admin. pada view index.php ini kita akan membuat halaman dashboard admin.

#### application/views/admin/header.php

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Dashboard - Aplikasi Perpustakaan</title>
5 <link rel="stylesheet" type="text/css" href="php echo base_url(). 'assets/css/bootstrap.css' ?">
6 <link rel="stylesheet" type="text/css" href="php echo base_url(). 'assets/datatables/datatables.css' ?">
7 <script type="text/javascript" src="php echo base_url(). 'assets/js/jquery.js' ?"></script>
8 <script type="text/javascript" src="php echo base_url(). 'assets/js/bootstrap.js' ?"></script>
9 <script type="text/javascript" src="php echo base_url(). 'assets/datatables/jquery.dataTables.js' ?"></script>
10 <script type="text/javascript" src="php echo base_url(). 'assets/datatables/datatables.js' ?"></script>
11 </head>
12 <body>
13 <nav class="navbar navbar-default">
14 <div class="container">
15 <!-- Brand and toggle get grouped for better mobile display -->
16 <div class="navbar-header">
17 <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
18 <span class="sr-only">Toggle navigation</span>
19 <span class="icon-bar"></span>
20 <span class="icon-bar"></span>
21 <span class="icon-bar"></span>
22 </button>
23 <a class="navbar-brand" href="php echo base_url(). 'admin'; ?">Perpustakaan</a>
24 </div>
25 <div class="collapse navbar-collapse" id="bs-examplenavbar-collapse-1">
26 <ul class="nav navbar-nav">
27 <li class="active"><a href="php echo base_url(). 'admin'; ?"><span class="glyphicon glyphicon-home"></span> Dashboard <span class="sr-only">(current)</span></a></li>
28 <li><a href="php echo base_url(). 'admin/buku'; ?"><span class="glyphicon glyphicon-folder-open"></span> Data Buku</a></li>
29 <li><a href="php echo base_url(). 'admin/anggota'; ?"><span class="glyphicon glyphicon-user"></span> Data Anggota</a></li>
30 <li><a href="php echo base_url(). 'admin/peminjaman'; ?"><span class="glyphicon glyphicon-sort"></span>
```

```

31         </span> Transaksi Peminjaman</a></li>
32         <li><a href="<?php echo base_url().'.admin/laporan'; ?>"><span class="glyphicon glyphicon-list-alt">
33         </span> Laporan</a></li>
34     </ul>
35     <ul class="nav navbar-nav navbar-right">
36         <li><a href="<?php echo base_url().'.admin/logout'; ?>"><span class="glyphicon glyphicon-log-out"></
37         span> Logout</a></li>
38         <li class="dropdown"><a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button"
39         aria-haspopup="true" aria-expanded="false"><?php echo "Halo, <b>
40         ".$this->session->userdata('nama'); ?></b> <span class="caret"></span></a>
41         <ul class="dropdown-menu">
42             <li><a href="<?php echo base_url().'.admin/ganti_password' ?>"><i class="glyphicon
43             glyphicon-lock"></i> Ganti Password</a></li>
44         </ul>
45     </li>
46 </ul>
47 </div><!-- /.navbar-collapse -->
48 </div><!-- /.container-fluid -->
49 </nav>
50 <div class="container">

```

## application/views/admin/index.php

```

1 <div class="page-header">
2     <h3>Dashboard</h3>
3 </div>
4 <div class="row">
5     <div class="col-lg-3 col-md-6">
6         <div class="panel panel-primary">
7             <div class="panel-heading">
8                 <div class="row">
9                     <div class="col-xs-3">
10                         <i class="glyphicon glyphicon-folder-open"></i>
11                     </div>
12                     <div class="col-xs-9 text-right">
13                         <div class="huge">
14                             <font size="18"><b><?php echo $this->m_perpus->get_data('buku')->num_rows(); ?></b></font>
15                         </div>
16                         <div><b>Jumlah Buku yang Terdaftar</b></div>
17                     </div>
18                 </div>
19             </div>
20             <a href="<?php echo base_url().'.admin/buku' ?>">
21                 <div class="panel-footer">
22                     <span class="pull-left">View Details</span>
23                     <span class="pull-right"><i class="glyphicon glyphicon-arrow-right"></i></span>
24                     <div class="clearfix"></div>
25                 </div>
26             </a>
27         </div>
28     </div>
29
30     <div class="col-lg-3 col-md-6">
31         <div class="panel panel-success">
32             <div class="panel-heading">
33                 <div class="row">
34                     <div class="col-xs-3">
35                         <i class="glyphicon glyphicon-user"></i>
36                     </div>
37                     <div class="col-xs-9 text-right">
38                         <div class="huge">
39                             <font size="18"><b><?php echo $this->m_perpus->get_data('anggota')->num_rows(); ?></b></font>
40                         </div>
41                         <div><b>Jumlah Anggota yang Terdaftar</b></div>
42                     </div>
43                 </div>
44             </div>
45             <a href="<?php echo base_url().'.admin/anggota' ?>">
46                 <div class="panel-footer">
47                     <span class="pull-left">View Details</span>
48                     <span class="pull-right"><i class="glyphicon glyphicon-arrow-right"></i></span>
49                     <div class="clearfix"></div>
50                 </div>
51             </a>
52         </div>
53     </div>
54 </div>

```

```

54
55 <div class="col-lg-3 col-md-6">
56   <div class="panel panel-warning">
57     <div class="panel-heading">
58       <div class="row">
59         <div class="col-xs-3">
60           <i class="glyphicon glyphiconsort"></i>
61         </div>
62         <div class="col-xs-9 text-right">
63           <div class="huge">
64             <font size="18"><b><?php echo $this->m_perpus->edit_data(array('status_peminjaman'=>0),'
65               transaksi')->num_rows(); ?></b></font>
66           </div>
67           <div><b>Peminjaman Belum Selesai</b></div>
68         </div>
69       </div>
70     <a href="<?php echo base_url(). 'admin/peminjaman'; ?>">
71       <div class="panel-footer">
72         <span class="pull-left">View Details</span>
73         <span class="pull-right"><i class="glyphicon glyphicon-arrow-right"></i></span>
74       <div class="clearfix"></div>
75     </div>
76   </a>
77 </div>
78 </div>
79
80 <div class="col-lg-3 col-md-6">
81   <div class="panel panel-danger">
82     <div class="panel-heading">
83       <div class="row">
84         <div class="col-xs-3">
85           <i class="glyphicon glyphiconok"></i>
86         </div>
87         <div class="col-xs-9 text-right">
88           <div class="huge">
89             <font size="18"><b><?php echo $this->m_perpus->edit_data(array('status_peminjaman'=>1),'
90               transaksi')->num_rows(); ?></b></font>
91           </div>
92           <div><b>Peminjaman Sudah Selesai</b></div>
93         </div>
94       </div>
95     <a href="<?php echo base_url(). 'admin/peminjaman'; ?>">
96       <div class="panel-footer">
97         <span class="pull-left">View Details</span>
98         <span class="pull-right"><i class="glyphicon glyphicon-arrow-right"></i></span>
99       <div class="clearfix"></div>
100     </div>
101   </a>
102 </div>
103 </div>
104 </div>
105
106 <hr>
107
108 <div class="row">
109   <div class="col-lg-4">
110     <div class="panel panel-default">
111       <div class="panel-heading">
112         <h3 class="panel-title" style="font-size: 18px; font-weight: bold;"><i class="glyphicon
113           glyphicon-random arrow-right"></i> Buku</h3>
114       </div>
115       <div class="panel-body">
116         <div class="list-group">
117           <?php foreach($buku as $b){ ?>
118             <a href="#" class="list-group-item">
119               <span class="badge"><?php if($b->status_buku == 1){echo "Tersedia";}else{echo "
120                 Dipinjam";}?></span>
121               <i class="glyphicon glyphiconuser"></i> <?php echo $b->judul_buku; ?>
122             </a>
123           <?php } ?>
124         </div>
125         <div class="text-right">
126           <a href="<?php echo base_url(). 'admin/buku' ?>">Lihat Semua Buku <i class="glyphicon
127             glyphicon-arrow-right"></i></a>
128         </div>
129       </div>
130     </div>
131 </div>

```

```

129
130     <div class="col-lg-3">
131         <div class="panel panel-default">
132             <div class="panel-heading">
133                 <h3 class="panel-title" style="font-size: 18px; font-weight: bold;"><i class="glyphicon glyphicon-user o"></i> Anggota Terbaru</h3>
134             </div>
135             <div class="panel-body">
136                 <div class="list-group">
137                     <?php foreach($anggota as $a){ ?>
138                         <a href="#" class="list-group-item">
139                             <span class="badge"><?php echo $a->gender; ?></span>
140                             <i class="glyphicon glyphicon-user"></i> <?php echo $a->nama_anggota; ?>
141                         </a>
142                     <?php } ?>
143                 </div>
144                 <div class="text-right">
145                     <a href="<?php echo base_url(). 'admin/anggota' ?>">Lihat Semua Anggota <i class="glyphicon glyphicon-arrow-right"></i></a>
146                 </div>
147             </div>
148         </div>
149     </div>
150
151     <div class="col-lg-5">
152         <div class="panel panel-default">
153             <div class="panel-heading">
154                 <h3 class="panel-title" style="font-size: 18px; font-weight: bold;"><i class="glyphicon glyphicon-sort"></i> Pinjaman Terakhir</h3>
155             </div>
156             <div class="panel-body">
157                 <div class="table-responsive">
158                     <table class="table table-bordered table-hover table-striped">
159                         <thead>
160                             <tr>
161                                 <th>Tgl. Transaksi</th>
162                                 <th>Tgl. Pinjam</th>
163                                 <th>Tgl. Kembali</th>
164                                 <th>Total Denda</th>
165                             </tr>
166                         </thead>
167                         <tbody>
168                             <?php
169                             <foreach($peminjaman as $p){
170                                 ?>
171                                 <tr>
172                                     <td><?php echo date('d/m/Y',strtotime($p->tgl_pencatatan)); ?></td>
173                                     <td><?php echo date('d/m/Y',strtotime($p->tgl_pinjam)); ?></td>
174                                     <td><?php echo date('d/m/Y',strtotime($p->tgl_kembali)); ?></td>
175                                     <td><?php echo "Rp.".number_format($p->total_denda). " , -"; ?></td>
176                                 </tr>
177                             <?php } ?>
178
179                             <div class="text-right">
180                                 <a href="<?php echo base_url(). 'admin/transaksi' ?>">Lihat Semua Transaksi <i class="glyphicon glyphicon-arrow-right"></i></a>
181                             </div>
182                         </tbody>
183                     </table>
184                 </div>
185             </div>
186         </div>
187     </div>
188     <!-- /.row -->
189 </div>
190 </div>
191 <!-- /.row -->

```

## Application/views/admin/footer.php

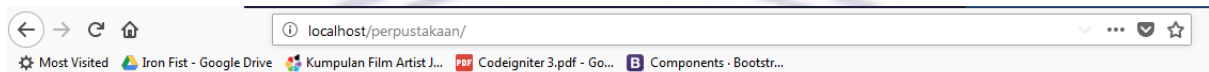
```

1 </div>
2 <script type="text/javascript">
3     $(document).ready(function(){
4         $("#table-datatable").dataTable();
5     });
6 </script>
7 </body>
8 </html>

```

Silahkan buka controller welcome pada browser untuk melihat hasilnya.  
**http://localhost/perpustakaan/**

perlu diperhatikan di sini, saat kita mengakses http://localhost/perpustakaan/ sama saja seperti kita mengakses http://localhost/perpustakaan/welcome. karena controller welcome telah kita set sebagai controller default.



## APLIKASI PERPUSTAKAAN

### LOGIN

Login

Coba memasukkan username dan password yang salah. Untuk memastikan sistem login yang kita buat sudah berjalan sebagaimana mestinya. Di sini saya mencoba memasukkan username : samsul, dan passwordnya samsul123.



## APLIKASI PERPUSTAKAAN

### LOGIN

Login gagal! Username dan password salah.

Login

Dan coba untuk akses halaman admin tanpa login. Dengan langsung mengakses controller admin. Seharusnya jika belum login, maka tidak bisa mengakses halaman admin (controller admin). Coba kita akses langsung halaman admin dengan mengakses url berikut.

<http://localhost/perpustakaan/admin>

maka akan muncul pesan notifikasi yang mengharuskan kita untuk melakukann login terlebih dulu.

## APLIKASI PERPUSTAKAAN

### LOGIN

Silahkan login dulu.

Login

Setelah kita mencoba masuk langsung ke halaman admin tanpa login, dan mencoba memasukkan username password yang salah, sekarang kita akan mencoba memasukkan username dan password yang benar. Coba kita masukkan username = imam, dan password = imam. Maka hasilnya kita berhasil login dan masuk ke halaman dashboard admin.

### Dashboard

12 Jumlah Buku View Details →	10 Jumlah Anggota View Details →	0 Jumlah Transaksi View Details →	0 Peminjaman Selesai View Details →
-------------------------------------	--	---	---

Buku

Pemrograman dan Hack Android untuk pemula dan adva

Tersedia

JavaScript uncover

Tersedia

24 Jam Belajar FrameWork

Dipinjam

Mikrokontroler

Dipinjam

Anggota Terbaru

Zaenal Abidin

Laki-Laki

Hisbu Utomo

Laki-Laki

Mus Dalifa

Perempuan

Tiwie Andrawati

Perempuan

Indah Dipo

Perempuan

Peminjaman Terakhir

Tgl. Transaksi	Tgl. Pinjam	Tgl. Kembali	Total
<a href="#">Lihat Semua Transaksi</a>			

## b. Membuat Logout

Setelah berhasil membuat login, selanjutnya kita akan membuat logout. Atau link keluar dari sistem. Perhatikan pada menu logout pada view header.php. kita telah membuat link menu logout.

```
<li><a href="<?php echo base_url(). 'admin/logout'; ?>"><span class="glyphicon glyphicon-log-out"></span> Logout</a></li>
```

Link logout di atas, mengarahkan ke function/method *logout* di controller *admin*. Oleh karena itu kita akan membuat function *logout* pada controller *admin*.

Tambahkan function logout berikut ini ke controller admin

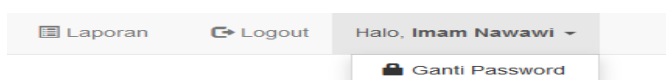
```
function logout(){  
    $this->session->sess_destroy();  
    redirect(base_url(). 'welcome?pesan=logout');
```

Syntax `$this->session->sess_destroy()`; merupakan fungsi codeigniter untuk menghapus semua *session*. Setelah menghapus session, kita alihkan halaman ke halaman login. Sekarang coba klik pada menu logtout. Maka halaman akan dialihkan kembali ke halaman login.

## c. Membuat Fitur Ganti password

Selanjutnya akan dibuat sebuah fitur untuk admin, yaitu fitur untuk mengganti password. Perhatikan pada menu ganti password di header admin. Kita telah membuat link menu untuk mengganti password. Dimana link tersebut sudah kita arahkan ke function *ganti\_password* di controller *admin*.

```
<li><a href="<?php echo base_url(). 'admin/ganti_password' ?>"><i class="glyphicon glyphicon-lock"></i> Ganti Password</a></li>
```



jadi kita akan membuat sebuah function baru di controller admin, yaitu function `ganti_password()`. Tambahkan function/syntax berikut pada controller admin.php.

*Application/controller/Admin.php*

```
function ganti_password(){  
    $this->load->view('admin/header');  
    $this->load->view('admin/ganti_password');  
    $this->load->view('admin/footer');  
}
```

Seperti yang terlihat pada function di atas, kita menampilkan view header.php, view ganti\_password.php dan view footer.php. pada view ganti\_password.php akan kita buat form untuk mengganti password.

Buat file view baru dengan nama ganti\_password.php. dalam folder admin dalam view.

*Application/view/admin/ganti\_password.php*

```
1 <div class="page-header">  
2   <h3>Ganti Password</h3>  
3 </div>  
4 <div class="row">  
5   <div class="col-md-6 col-md-offset-3">  
6     <?php  
7       if(isset($_GET['pesan'])){  
8         if($_GET['pesan'] == "berhasil"){  
9           echo "<div class='alert alert-success'>Password berhasil di ganti.</div>";  
10        }  
11      }  
12    <?>  
13    <form action="<?php echo base_url(). 'admin/ganti_password_act' ?>" method="post">  
14      <div class="form-group">  
15        <label>Password Baru</label>  
16        <input class="form-control" type="password" name="pass_baru">  
17        <?php echo form_error('pass_baru'); ?>  
18      </div>  
19  
20      <div class="form-group">  
21        <label>Ulangi Password Baru</label>  
22        <input class="form-control" type="password" name="ulang_pass">  
23        <?php echo form_error('ulang_pass'); ?>  
24      </div>  
25  
26      <div class="form-group">  
27        <input class="btn btn-primary btn-sm" type="submit" value="Simpan">  
28      </div>  
29    </form>  
30  </div>  
31 </div>
```

Aksi dari form ini akan di jalankan pada function *ganti\_password\_act* pada controller *admin*.

```
<form action="php echo base_url(). 'admin/ganti_password_act' ?" method="post">
```

Selanjutnya buat function *ganti\_password\_act()* pada controller *admin*.

#### *Application/controllers/Admin.php*

```
function ganti_password_act(){
    $pass_baru = $this->input->post('pass_baru');
    $ulang_pass = $this->input->post('ulang_pass');

    $this->form_validation->set_rules('pass_baru', 'Password Baru', 'required|matches[ulang_pass]');
    $this->form_validation->set_rules('ulang_pass', 'Ulangi Password Baru', 'required');
    if($this->form_validation->run() != false){
        $data = array('admin_password' => md5($pass_baru));
        $w = array('admin_id' => $this->session->userdata('id'));
        $this->m_rental->update_data($w,$data,'admin');
        redirect(base_url(). 'admin/ganti_password?pesan=berhasil');
    }else{
        $this->load->view('admin/header');
        $this->load->view('admin/ganti_password');
        $this->load->view('admin/footer');
    }
}
```

Syntax di atas adalah function yang kita buat untuk menangani form ganti password. Perhatikan pada validation yang kita buat, kita menggunakan validation required dan matches. Seperti yang pernah disinggung sebelumnya tentang form validation required, required berarti kita menetapkan form tersebut menjadi wajib di isi atau tidak boleh kosong. Dan matches[] adalah form validation untuk memvalidasi kesamaan value. Di sini kita mewajibkan password baru harus sama dengan form ulangi password.

```
$pass_baru = $this->input->post('pass_baru');
$ulang_pass = $this->input->post('ulang_pass');

$this->form_validation->set_rules('pass_baru', 'Password Baru', 'required|matches[ulang_pass]');
$this->form_validation->set_rules('ulang_pass', 'Ulangi Password Baru', 'required');
```

Kemudian kita cek validasinya, jika sesuai dan tidak ada masalah, maka kita update data password si admin.

```

if($this->form_validation->run() != false){
    $data = array('admin_password' => md5($pass_baru));
    $w = array('admin_id' => $this->session->userdata('id'));
    $this->m_rental->update_data($w,$data,'admin');
    redirect(base_url().'admin/ganti_password?pesan=berhasil');
}else{
    $this->load->view('admin/header');
    $this->load->view('admin/ganti_password');
    $this->load->view('admin/footer');
}

```

Syntax `$this->session->userdata('id')` merupakan perintah untuk menampilkan session 'id' admin yang sedang login. Setelah data password admin di update, maka kemudian halaman akan dialihkan lagi ke function *ganti\_password*. Sekarang kita akan mencoba mengganti password admin. Seperti yang dijelaskan sebelumnya, kita membuat validasi password dimana password baru yang diketikkan pada form password baru harus sama dengan yang di ketikkan pada form ulang password. Ini berguna untuk memvalidasi dan mencegah admin salah memasukkan password barunya (salah ketik). Disini kita akan mencoba mengganti password admin dengan “*administrator*”. Tapi kita coba dulu memasukkan password yang berbeda. Maka hasilnya akan muncul pesan notifikasi bahwa “password baru” dan “ulangi password baru” yang di input harus sama.

Ketika password baru dan ulangi password baru diinput dengan password yang sama, maka akan muncul pesan password berhasil duganti.

#### Ganti Password

Password berhasil di ganti.

Password Baru

Ulangi Password Baru

Simpan



**PERTEMUAN 7**  
**REVIEW MATERI**



**REVIEW MATERI**

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



## PERTEMUAN 9

### MEMBUAT CRUD DATA BUKU

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

Dalam kasus ini akan membuat CRUD untuk data-data buku, sesuai dengan gambaran perancangan database di pertemuan sebelumnya. Data-data buku yang akan disimpan adalah data id buku, judul buku, pengarang, penerbit, tahun terbit, isbn, jumlah buku, lokasi, gambar dan tanggal input.

#### 1. Menampilkan Data Buku

Langkah pertama akan dibuatkan file untuk menampilkan data buku. Data buku akan kita tampilkan pada function/method buku(). Sesuai dengan link yang sudah kita tentukan di menu “Data Buku” pada menu navigasi.

```
<li><a href="<?php echo base_url(). 'admin/buku'; ?>"><span class="glyphicon glyphicon-folder-open">
</span> Data Buku</a></li>
```

Tambahkan sebuah function dengan nama buku() pada controller admin.php.

*Application/controllers/Admin.php*

```
function buku(){
    $data['buku'] = $this->m_perpus->get_data('buku')->result();
    $this->load->view('admin/header');
    $this->load->view('admin/databuku',$data);
    $this->load->view('admin/footer');
}
```

Pertama, kita akan mengambil data buku dari database dengan menggunakan function *get\_data()* di model *m\_perpus*.

```
function edit_data($where,$table){
    return $this->db->get_where($table,$where);
}
```

Dari data yang didapatkan dari hasil query di function edit\_data() di atas kita return kembali untuk dimasukkan ke variabel array buku.

```
$data['buku'] = $this->m_perpus->get_data('buku')->result();  
$this->load->view('admin/header');  
$this->load->view('admin/databuku',$data);  
$this->load->view('admin/footer');
```

Lalu kita parsing/kirim datanya ke view buku.php seperti pada syntax di atas. jadi di sini pada view buku.php lah kita akan menampilkan data bukunya.

Buat view baru di dalam folder admin dengan nama buku.php

#### *Application/view/admin/buku.php*

```
1 <div class="page-header">  
2   <h3>Data Buku</h3>  
3 </div>  
4 <a href="<?php echo base_url(). 'admin/tambah_buku'; ?>" class="btn btn-primary btn-xs"><span class="glyphicon  
5   glyphicon-plus"></span> Buku Baru</a>  
6 <br/><br/>  
7 <div class="table-responsive">  
8   <table class="table table-bordered table-striped table-hover" id="table-datatable">  
9     <thead>  
10      <tr>  
11        <th>No</th>  
12        <th>Gambar</th>  
13        <th>Judul Buku</th>  
14        <th>Pengarang</th>  
15        <th>Penerbit</th>  
16        <th>Tahun Terbit</th>  
17        <th>ISBN</th>  
18        <th>Lokasi</th>  
19        <th>Status</th>  
20        <th>Pilihan</th>  
21      </tr>  
22    </thead>  
23    <tbody>  
24      <?php  
25        $no = 1;  
26        foreach($buku as $b){  
27          ?>  
28          <tr>
```

```

28 <td><?php echo $no++; ?></td>
29 <td></td>
30 <td><?php echo $b->judul_buku ?></td>
31 <td><?php echo $b->pengarang ?></td>
32 <td><?php echo $b->penerbit ?></td>
33 <td><?php echo $b->thn_terbit ?></td>
34 <td><?php echo $b->isbn ?></td>
35 <td><?php echo $b->lokasi ?></td>
36 <td>
37 <?php
38     if($b->status_buku == "1"){
39         echo "Tersedia";
40     }else if($b->status_buku == "0"){
41         echo "Sedang Di Pinjam";
42     }
43 ?>
44 </td>
45 <td nowrap="nowrap">
46     <a class="btn btn-primary btn-xs" href="<?php echo base_url(). 'admin/edit_buku/' . $b->id_buku; ?>">
47         <span class="glyphicon glyphicon-zoom-in"></span></a>
48     <a class="btn btn-warning btn-xs" href="<?php echo base_url(). 'admin/hapus_buku/' . $b->id_buku; ?>">
49         <span class="glyphicon glyphicon-remove"></span></a>
50 </td>
51 </tr>
52 <?php } ?>
53 </tbody>
</table>
</div>

```

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

[+ Tambah Buku](#)

10 records per page Search:

No	Judul Buku	Pengarang	Penerbit	Tahun Terbit	ISBN	Gambar	Status	Pilihan
1	Robotika Sederhana	Siswoyo Utomo	Wacana Ria	2013-01-01	34354422	---	Tersedia	<a href="#">🔍</a> <a href="#">✕</a>
2	Mahir Dengan PHP	Adri Kusuma	Pustaka Bangsa	2011-02-02	654123fe	---	Tersedia	<a href="#">🔍</a> <a href="#">✕</a>
3	Mahir Mewarnai	Akhmad Rahmat	CV.Indo Kreasi	2014-03-03	76234hww342	---	Tersedia	<a href="#">🔍</a> <a href="#">✕</a>
4	Hukum Fisika	Kurnia Sandi	Wacana Ria	2013-04-04	233214414	---	Sedang Di Pinjam	<a href="#">🔍</a> <a href="#">✕</a>
5	Mahir Bahasa Inggris	Aliuddin	CV.Indo Kreasi	2013-05-05	3553234454	---	Sedang Di Pinjam	<a href="#">🔍</a> <a href="#">✕</a>
6	Public Speaking	Pambudi Prasetyo	Aldi Pustaka	2015-06-06	843594759	---	Tersedia	<a href="#">🔍</a> <a href="#">✕</a>
7	Trik SQL	Ahdim Makaren	Wacana Ria	2014-07-07	54234762	---	Sedang Di Pinjam	<a href="#">🔍</a> <a href="#">✕</a>
8	Kemurnian Agama	Pambudi Prasetyo	Aldi Pustaka	2014-08-08	980958607	---	Tersedia	<a href="#">🔍</a> <a href="#">✕</a>
9	Mikrokontroler	Ahdim Makaren	Wacana Ria	2012-09-09	12121314	---	Sedang Di Pinjam	<a href="#">🔍</a> <a href="#">✕</a>
10	24 Jam Belajar FrameWork	Rudi Hartono	Unjung Pena	2017-03-02	12345345	---	Sedang Di Pinjam	<a href="#">🔍</a> <a href="#">✕</a>

Showing 1 to 10 of 12 entries

← Previous 1 2 Next →

## 2. Input Data Buku

Form input data buku akan kita buat pada method/function *tambah\_buku()* dalam controller *Admin.php*. seperti yang sudah kita arahkan pada tombol “Buku Baru”.

```

<a href="<?php echo base_url(). 'admin/tambah_buku'; ?>" class="btn btn-primary btn-xs">
    <span class="glyphicon glyphicon-plus"></span> Buku Baru</a>

```

Buat function *tambah\_buku()* baru pada controller *Admin.php*.

#### *Application/controller/Admin.php*

```
function tambah_buku(){
    //memuat data kategori untuk ditampilkan di select form
    $data['kategori'] = $this->m_perpus->get_data('kategori')->result();

    $this->load->view('admin/header');
    $this->load->view('admin/tambahbuku',$data);
    $this->load->view('admin/footer');
}
```

Pada function *tambah\_buku()* ini kita akan membuat form penginputan data buku baru. Yaitu akan kita buat pada view *tambahbuku.php*. seperti yang sudah kita tetapkan pada function *tambah\_buku()*

#### *Application/view/admin/tambahbuku.php*

```
1 <div class="page-header">
2   <h3>Buku Baru</h3>
3 </div>
4 <?= validation_errors('<p style="color:red;">','</p>'); ?>
5 <?php
6 if($this->session->flashdata())
7 {
8     echo "<div class='alert alert-danger alert-message'>";
9     echo $this->session->flashdata('alert');
10    echo "</div>";
11 }
12 >>
13 <form action="<?php echo base_url(). 'admin/tambah_buku_act' ?>" method="post" enctype="multipart/form-data">
14
15 <div class="form-group">
16   <label>Kategori</label>
17   <select name="id_kategori" class="form-control">
18     <option value="">-Pilih Kategori</option>
19     <?php foreach($kategori as $k){ ?>
20       <option value="<?php echo $k->id_kategori; ?>"><?php echo $k->nama_kategori; ?></option>
21     <?php } ?>
22   </select>
23   <?php echo form_error('id_kategori'); ?>
24 </div>
25
26 <div class="form-group">
27   <label>Judul Buku</label>
28   <input type="text" name="judul_buku" class="form-control">
29   <?php echo form_error('judul_buku'); ?>
30 </div>
31
```



```

31
32     <div class="form-group">
33         <label>Pengarang</label>
34         <input type="text" name="pengarang" class="form-control">
35     </div>
36
37     <div class="form-group">
38         <label>Penerbit</label>
39         <input type="text" name="penerbit" class="form-control">
40     </div>
41
42     <div class="form-group">
43         <label>Tahun Terbit</label>
44         <input type="date" name="thn_terbit" class="form-control">
45     </div>
46
47     <div class="form-group">
48         <label>ISBN</label>
49         <input type="text" name="isbn" class="form-control">
50     </div>
51
52     <div class="form-group">
53         <label>Jumlah Buku</label>
54         <input type="text" name="jumlah_buku" class="form-control">
55     </div>
56
57     <div class="form-group">
58         <label>Lokasi</label>
59         <input type="text" name="lokasi" class="form-control">
60     </div>
61
62     <div class="form-group">
63         <label>Status Buku</label>
64         <select name="status" class="form-control">
65             <option value="1">Tersedia</option>
66             <option value="0">Sedang Di Pinjam</option>
67         </select>
68         <?php echo form_error('status'); ?>
69     </div>
70
71     <div class="form-group">
72         <label>Gambar</label>
73         <input name="foto" type="file" class="form-control">
74     </div>
75
76     <div class="form-group">
77         <input type="submit" value="Simpan" class="btn btnprimary">
78     </div>
79 </div>
80 </form>

```

Form tambah buku ini akan di proses pada function/method *tambah\_buku\_act()* di controller *admin.php*.

Jadi kita buat lagi function baru di controller *admin.php*. yaitu function *tambah\_buku\_act()*.  
*Application/controller/Admin.php*

```

68
69     function tambah_buku_act(){
70         $tgl_input = date('Y-m-d');
71         $id_kategori = $this->input->post('id_kategori');
72         $judul = $this->input->post('judul_buku');
73         $pengarang = $this->input->post('pengarang');
74         $penerbit = $this->input->post('penerbit');
75         $thn_terbit = $this->input->post('thn_terbit');
76         $isbn = $this->input->post('isbn');
77         $jumlah_buku = $this->input->post('jumlah_buku');
78         $lokasi = $this->input->post('lokasi');
79         $status = $this->input->post('status');
80         $this->form_validation->set_rules('id_kategori', 'Kategori', 'required');
81         $this->form_validation->set_rules('judul_buku', 'Judul Buku', 'required');
82         $this->form_validation->set_rules('status', 'Status Buku', 'required');
83         if($this->form_validation->run() != false){
84             //konfigurasi upload gambar
85             $config['upload_path'] = './assets/upload/';
86             $config['allowed_types'] = 'jpg|png|jpeg';
87             $config['max_size'] = '2048';
88             $config['file_name'] = 'gambar'.time();
89
90             $this->load->library('upload', $config);
91
92             if($this->upload->do_upload('foto')){
93                 $image=$this->upload->data();
94
95                 $data = array(
96                     'id_kategori' => $id_kategori,
97                     'judul_buku' => $judul,
98                     'pengarang' => $pengarang,
99                     'penerbit' => $penerbit,
100                     'thn_terbit' => $thn_terbit,
101                     'isbn' => $isbn,
102                     'jumlah_buku' => $jumlah_buku,
103                     'lokasi' => $lokasi,
104                     'gambar' => $image['file_name'],
105                     'tgl_input' => $tgl_input,
106                     'status_buku' => $status
107                 );
108
109                 $this->m_perpus->insert_data($data, 'buku');
110                 redirect(base_url(). 'admin/buku');
111             }else{
112                 $this->session->set_flashdata('alert', 'Anda Belum Memilih Foto');
113             }
114         }else{
115             $this->load->view('admin/header');
116             $this->load->view('admin/tambahbuku');
117             $this->load->view('admin/footer');
118         }
119     }

```

Sekarang kita akan mencoba menginput data buku baru dengan klik pada tombol “buku baru”. Silahkan coba diisi semua, kemudian disimpan, ketika berhasil menyimpan maka akan kembali ke tampilan data buku.

### 3. Update Data Buku

Untuk megedit data buku, akan kita buat sebuah form baru lagi pada function *edit\_buku()* di controller *admin.php*. seperti yang telah kita arahkan pada tombol edit di data buku.

```
<a class="btn btn-primary btn-xs" href="{<?php echo base_url(). 'admin/edit_buku/' . $b->id_buku; ?>}"><span class="glyphicon glyphicon-zoom-in"></span></a>
```

Buat function baru lagi dengan nama *edit\_buku()* dalam controller *admin.php*.

*Application/controller/Admin.php*

```
function edit_buku($id){
    $where = array('id_buku' => $id);
    $data['buku'] = $this->db->query("select * from buku B, kategori K where B.id_kategori=K.id_kategori and B.id_buku='$id'")->result();
    $data['kategori'] = $this->m_perpus->get_data('kategori')->result();
    //$data['buku'] = $this->m_perpus->edit_data($where, 'buku')->result();
    $this->load->view('admin/header');
    $this->load->view('admin/editbuku', $data);
    $this->load->view('admin/footer');
}
```

Di sini kita menangkap data id buku yang ingin diedit di dalam parameter function *edit\_buku()*. Kemudian kita ambil data buku yang memiliki id tersebut, kemudian kita parsing/kirim ke view *editbuku.php*. karena pada view *editbuku.php* inilah akan kita buat form untuk mengedit data buku.

*Application/view/admin/editbuku.php*

```

1 <div class="page-header">
2   <h3>Edit Buku</h3>
3 </div>
4 <?php foreach($buku as $b){ ?>
5 <form action="<?php echo base_url(). 'admin/update_buku' ?>" method="post" enctype="multipart/form-data">
6   <div class="form-group">
7     <label>Kategori</label>
8     <select name="id_kategori" class="form-control">
9       <option value="<?php echo $b->id_kategori; ?>"><?php echo $b->nama_kategori; ?></option>
10      <?php foreach($kategori as $k){ ?>
11        <option value="<?php echo $k->id_kategori; ?>"><?php echo $k->nama_kategori; ?></option>
12      <?php } ?>
13    </select>
14    <?php echo form_error('id_kategori'); ?>
15  </div>
16
17  <div class="form-group">
18    <label>Judul Buku</label>
19    <input type="hidden" name="id" value="<?php echo $b->id_buku; ?>">
20    <input class="form-control" type="text" name="judul_buku" value="<?php echo $b->judul_buku; ?>">
21    <?php echo form_error('judul_buku'); ?>
22  </div>
23
24  <div class="form-group">
25    <label>Pengarang</label>
26    <input class="form-control" type="text" name="pengarang" value="<?php echo $b->pengarang; ?>">
27    <?php echo form_error('pengarang'); ?>
28  </div>
29
30  <div class="form-group">
31    <label>Penerbit</label>
32    <input class="form-control" type="text" name="penerbit" value="<?php echo $b->penerbit; ?>">
33    <?php echo form_error('penerbit'); ?>
34  </div>
35
36  <div class="form-group">
37    <label>Tahun Terbit</label>
38    <input class="form-control" type="text" name="thn_terbit" value="<?php echo $b->thn_terbit; ?>">
39    <?php echo form_error('thn_terbit'); ?>
40  </div>
41
42  <div class="form-group">
43    <label>ISBN</label>
44    <input class="form-control" type="text" name="isbn" value="<?php echo $b->isbn; ?>">
45    <?php echo form_error('isbn'); ?>
46  </div>
47
48  <div class="form-group">
49    <label>Jumlah Buku</label>
50    <input class="form-control" type="text" name="jumlah_buku" value="<?php echo $b->jumlah_buku; ?>">
51    <?php echo form_error('jumlah_buku'); ?>
52  </div>
53
54  <div class="form-group">
55    <label>Lokasi</label>
56    <input class="form-control" type="text" name="lokasi" value="<?php echo $b->lokasi; ?>">
57    <?php echo form_error('lokasi'); ?>
58  </div>
59
60  <div class="form-group">
61    <label>Status</label>
62    <select name="status" class="form-control">
63      <option <?php if($b->status_buku == "1"){echo "selected='selected'";} echo $b->status_buku; ?> value="1">
        Tersedia</option>
64      <option <?php if($b->status_buku == "0"){echo "selected='selected'";} echo $b->status_buku; ?> value="0">
        Sedang Di Pinjam</option>
65    </select>
66    <?php echo form_error('status'); ?>
67  </div>
68
69  <div class="form-group">
70    <label>Gambar</label>
71    <?php
72      if(isset($b->gambar)){
73        echo '<input type="hidden" name="old_pict" value="'. $b->gambar. ' ">';
74        echo '';
75      }
76    >
77    <input name="foto" type="file" class="form-control">
78  </div>
79
80  <div class="form-group">
81    <input type="submit" value="Update" class="btn btnprimary">
82  </div>
83 </form>
84 <?php } ?>

```

Form edit data buku ini akan di proses pada function *update\_buku()* dalam controller *admin.php* pada saat disubmit.

```
<form action="<?php echo base_url().'admin/update_buku' ?>" method="post">
```

Jadi buat lagi sebuah function baru dengan nama *update\_buku()* pada controller *Admin.php*.

### Application/controller/Admin.php

```
function update_buku(){
    $id = $this->input->post('id');
    $id_kategori = $this->input->post('id_kategori');
    $judul = $this->input->post('judul_buku');
    $pengarang = $this->input->post('pengarang');
    $penerbit = $this->input->post('penerbit');
    $thn_terbit = $this->input->post('thn_terbit');
    $isbn = $this->input->post('isbn');
    $jumlah_buku = $this->input->post('jumlah_buku');
    $lokasi = $this->input->post('lokasi');
    $status = $this->input->post('status');
    $this->form_validation->set_rules('id_kategori', 'ID Kategori', 'required');
    $this->form_validation->set_rules('judul_buku', 'Judul Buku', 'required|min_length[4]');
    $this->form_validation->set_rules('pengarang', 'Pengarang', 'required|min_length[4]');
    $this->form_validation->set_rules('penerbit', 'Penerbit', 'required|min_length[4]');
    $this->form_validation->set_rules('thn_terbit', 'Tahun Terbit', 'required|min_length[4]');
    $this->form_validation->set_rules('isbn', 'Nomer ISBN', 'required|numeric');
    $this->form_validation->set_rules('jumlah_buku', 'Jumlah Buku', 'required|numeric');
    $this->form_validation->set_rules('lokasi', 'Lokasi Buku', 'required|min_length[4]');
    $this->form_validation->set_rules('status', 'Status Buku', 'required');

    if($this->form_validation->run() != false){
        $config['upload_path'] = './assets/upload/';
        $config['allowed_types'] = 'jpg|png|jpeg';
        $config['max_size'] = '2048';
        $config['file_name'] = 'gambar'.time();

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

```

159
160 $where = array('id_buku' => $id);
161 ▼ $data = array(
162     'id_kategori' => $id_kategori,
163     'judul_buku' => $judul,
164     'pengarang' => $pengarang,
165     'penerbit' => $penerbit,
166     'thn_terbit' => $thn_terbit,
167     'isbn' => $isbn,
168     'jumlah_buku' => $jumlah_buku,
169     'lokasi' => $lokasi,
170     'gambar' => $image['file_name'],
171     'status_buku' => $status
172 );
173
174 ▼ if($this->upload->do_upload('foto')){
175     //proses upload gambar
176     $image = $this->upload->data();
177     unlink('assets/upload/'.$this->input->post('old_pict', TRUE));
178     $data['gambar'] = $image['file_name'];
179
180     $this->m_perpus->update_data($where, $data, 'buku');
181 }else {
182     $this->m_perpus->update_data($where, $data, 'buku');
183 }
184
185 $this->m_perpus->update_data($where, $data, 'buku');
186 redirect(base_url().'admin/buku');
187 }else{
188     $where = array('id_buku' => $id);
189     $data['buku'] = $this->db->query("select * from buku B, kategori K where B.id_kategori=K.id_kategori and
190         B.id_buku='$id'")->result();
191     $data['kategori'] = $this->m_perpus->get_data('kategori')->result();
192     // $data['buku'] = $this->m_perpus->edit_data($where, 'buku')->result();
193     $this->load->view('admin/header');
194     $this->load->view('admin/editbuku', $data);
195     $this->load->view('admin/footer');
196 }
197 }

```

Sekarang kita akan mencoba mengubah/mengedit salah satu data buku.

Klik pada tombol edit. Lalu ubah yang dibutuhkan kemudian simpan/update

#### 4. Menghapus Data Buku.

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

```

<a class="btn btn-warning btn-xs" href="php echo base_url().'admin/hapus_buku/'.$b-&gt;id_buku; ?">
<span class="glyphicon glyphicon-remove"></span></a>

```

Untuk aksi dari penghapusan data buku kita akan membuatnya di function/method hapus\_buku() dalam controller admin. Buat sebuah method baru dengan nama hapus\_buku() di controller admin.php.



*Application/controllers/admin.php*

```
function hapus_buku($id){  
    $where = array('id_buku' => $id);  
    $this->m_perpus->delete_data($where,'buku');  
    redirect(base_url().'admin/buku');  
}
```

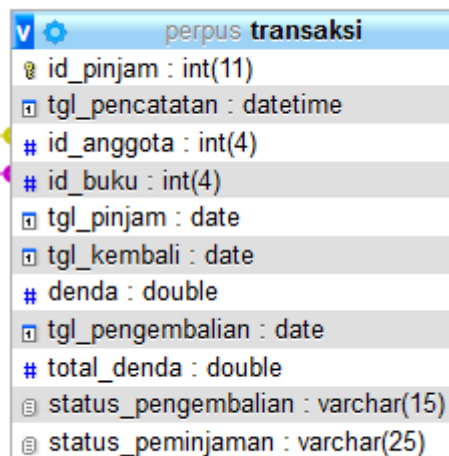
Sekarang untuk mencobanya, bisa klik tombol hapus pada buku yang ingin di hapus. Maka data bukupun terhapus.



## PERTEMUAN 10

### MEMBUAT PROSES TRANSASKI PEMINJAMAN

Setelah selesai membuat CRUD data master. Selanjutnya kita akan lanjut membuat proses transaksi peminjaman buku. Silahkan perhatikan kembali perancangan table transaksi yang telah kita buat.



	perpus transaksi
id_pinjam	int(11)
tgl_pencatatan	datetime
# id_anggota	int(4)
# id_buku	int(4)
tgl_pinjam	date
tgl_kembali	date
# denda	double
tgl_pengembalian	date
# total_denda	double
status_pengembalian	varchar(15)
status_peminjaman	varchar(25)

Seperti yang sudah disinggung sebelumnya. Pada table transaksi ini kita akan menyimpan data id anggota yang ingin meminjam buku di kolom anggota. Selengkapnya silahkan perhatikan pada table berikut.

#### 1. Menampilkan Data Transaksi Peminjaman

Pertama kita akan menampilkan dulu data peminjaman. Kita akan menampilkan data peminjaman di method peminjaman(). Dalam controller Admin.php. Buat sebuah method baru pada controller Admin.php. yaitu method peminjaman().

##### Application/controllers/Admin.php

```
function peminjaman(){  
    $data['peminjaman'] = $this->db->query("SELECT * FROM transaksi T, buku B, anggota A WHERE T.id_buku=B.id_buku  
    and T.id_anggota=A.id_anggota")->result();  
    $this->load->view('admin/header');  
    $this->load->view('admin/peminjaman',$data);  
    $this->load->view('admin/footer');  
}
```

Query mysql yang kita jalankan di sini adalah “select \* from peminjaman, detail\_pinjam where peminjaman.id\_pinjam=detail\_pinjam.id\_pinjam”. yaitu untuk menampilkan atau mengambil data dari table peminjaman, tabel detail pinjam yang id\_pinjam sma dengan detail\_pinjam. Setelahh kita mendapatkan data kemudian kita parsing ke view peminjama.php untuk ditampilkan.

Buat view baru dalam folder admin dengan nama peminjama.php.

### *Application/view/admin/peminjaman.php*

```

1 <div class="page-header">
2 <h3>Data Transaksi</h3>
3 </div>
4 <a href="{<?php echo base_url(). 'admin/transaksi_add'; ?>}" class="btn btn-primary btn-sm"><span class="glyphicon
5 <br/><br/>
6 <div class="table-responsive">
7 <table class="table table-bordered table-striped table-hover" id="table-datatable">
8 <thead>
9 <tr>
10 <th>No</th>
11 <th>Anggota</th>
12 <th>Buku</th>
13 <th>Tgl. Pinjam</th>
14 <th>Tgl. Kembali</th>
15 <th>Denda / Hari</th>
16 <th>Tgl. Dikembalikan</th>
17 <th>Total Denda</th>
18 <th>Status Buku</th>
19 <th>Status Pinjam</th>
20 </tr>
21 </thead>
22 <tbody>
23 <?php
24 $no = 1;
25 foreach($peminjaman as $p){
26 >
27 <tr>
28 <td><?php echo $no++; ?></td>
29 <td><?php echo $p->id_anggota; ?></td>
30 <td><?php echo $p->id_buku; ?></td>
31 <td><?php echo date('d/m/Y',strtotime($p->tgl_pinjam)); ?></td>
32 <td><?php echo date('d/m/Y',strtotime($p->tgl_kembali)); ?></td>
33 <td><?php echo "Rp. ".number_format($p->denda); ?></td>
34 <td>
35 <?php
36 if($p->tgl_pengembalian == "0000-00-00"){
37 echo "-";
38 }else{
39 echo date('d/m/Y',strtotime($p->tgl_pengembalian));
40 }
41 >
42 </td>
43 <td><?php echo "Rp. ". number_format($p->totaldenda). " , -"; ?></td>
44 <td>
45 <?php
46 if($p->status_buku == "1"){
47 echo "Selesai";
48 }else{
49 echo "-";
50 }
51 >
52 </td>

```

```

53         <td>
54             <?php
55                 if($p->status == "1"){
56                     echo "-";
57                 }else{
58                     ?>
59                     <a class="btn btn-sm btn-success" href="<?php echo base_url(). 'admin/transaksi_selesai/' . $t->
60                         transaksi_id; ?>"><span class="glyphicon glyphicon-ok"></span> Transaksi Selesai</a>
61                     <br/>
62                     <a class="btn btn-sm btn-danger" href="<?php echo base_url(). 'admin/transaksi_hapus/' . $t->
63                         transaksi_id; ?>"><span class="glyphicon glyphicon-remove"></span> Batalkan Transaksi</a>
64                     <?php } ?>
65                 </td>
66             </tr>
67             <?php } ?>
68         </tbody>
69     </table>
70 </div>

```

Dan hasilnya adalah

#### Data Transaksi

+ Transaksi Baru

10 records per page
Search:

No	Anggota	Buku	Tgl. Pinjam	Tgl. Kembali	Denda / Hari	Tgl. Dikembalikan	Total Denda	Status Buku	Status Pinjam
No data available in table									

Showing 0 to 0 of 0 entries

← Previous
Next →

Pada table data peminjaman di atas, kita telah berhasil menampilkan data peminjaman. Namun masih kosong karena belum ada data peminjaman yang diinput. Oh ya, kita juga membuat kondisi, jika status transaksi nantinya sama dengan 1. Maka kita tampilkan “selesai” pada kolom status di tabel transaksi di atas. Dan pada kolom terakhir kita membuat kondisi juga, jika status transaksi belum selesai, maka kita tampilkan tombol “batalkan transaksi” dan tombol “transaksi selesai”.

## 2. Membuat Proses Transaksi

membuat penginputan transaksi peminjaman buku. Penginputan transaksi akan kita buat pada method *tambah\_peminjaman()*. Langsung saja kita buat method *tambah\_peminjaman()* di controller **Admin.php**.

### Application/controllers/Admin.php

```
function tambah_peminjaman(){
    $w = array('status_buku'=>'1');
    $data['buku'] = $this->m_perpus->edit_data($w,'buku')->result();
    $data['anggota'] = $this->m_perpus->get_data('anggota')->result();
    $data['peminjaman'] = $this->m_perpus->get_data('transaksi')->result();

    $this->load->view('admin/header');
    $this->load->view('admin/tambah_peminjaman',$data);
    $this->load->view('admin/footer');
}
```

Seperti yang terlihat pada method di atas. di sini kita akan membuat form input transaksi nya di view *tambah\_peminjaman.php*. dan juga mengirimkan (parsing) 2 data ke view *tambah\_peminjaman.php*. yaitu data buku dan data anggota.

```
$data['buku'] = $this->m_perpus->edit_data($w,'buku')->result();
$data['anggota'] = $this->m_perpus->get_data('anggota')->result();
```

Kemudian buat view nya dengan nama *tambah\_peminjaman.php*. seperti yang telah kita tetapkan pada method *tambah\_peminjaman()*.

*Application/views/admin/tambah\_peminjaman.php*

```
Admin.php x tambah_peminjaman.php x tambahbuku.php x peminjaman.php x
1 <div class="page-header">
2   <h3>Transaksi Baru</h3>
3 </div>
4 <form action="<?php echo base_url(). 'admin/tambah_peminjaman_act' ?>" method="post">
5   <div class="form-group">
6     <label>Anggota</label>
7     <select name="anggota" class="form-control">
8       <option value="">-Pilih Anggota-</option>
9       <?php foreach($anggota as $a){ ?>
10        <option value="<?php echo $a->id_anggota; ?>"><?php echo $a->nama_anggota; ?></option>
11        <?php } ?>
12      </select>
13      <?php echo form_error('anggota'); ?>
14    </div>
15
16    <div class="form-group">
17      <label>Buku</label>
18      <select name="buku" class="form-control">
19        <option value="">-Pilih Buku-</option>
20        <?php foreach($buku as $b){ ?>
21          <option value="<?php echo $b->id_buku; ?>"><?php echo $b->judul_buku; ?></option>
22          <?php } ?>
23        </select>
24        <?php echo form_error('buku'); ?>
25      </div>
26
27    <div class="form-group">
28      <label>Tanggal Pinjam</label>
29      <input type="date" name="tgl_pinjam" class="form-control">
30      <?php echo form_error('tgl_pinjam'); ?>
31    </div>
32  </form>
```

```

32
33     <div class="form-group">
34         <label>Tanggal Kembali</label>
35         <input type="date" name="tgl_kembali" class="form-control">
36         <?php echo form_error('tgl_kembali'); ?>
37     </div>
38
39     <div class="form-group">
40         <label>Harga Denda / Hari</label>
41         <input type="text" name="denda" class="form-control">
42         <?php echo form_error('denda'); ?>
43     </div>
44
45     <div class="form-group">
46         <input type="submit" value="Simpan" class="btn btnprimary btn-sm">
47     </div>
48 </form>

```

Tadi pada method `tambah_peminjaman`, kita mengirimkan data buku dan data anggota ke view `tambah_peminjaman.php`. Kemudian pada view `tmabh_peminjaman.php` kita tampilkan data buku pada form buku. Dan tampilkan data anggota pada form anggota.

Selanjutnya kita akan membuat aksi dari form input transaksi ini. Aksi dari form ini akan kita buat di method `tambah_peminjaman_act` di controller `Admin.php`. seperti yang sudah kita buat pada form input transaksi ini.

Buat method baru dengan nama `transaksi_add_act` di controller `admin.php`.

***Application/controllers/admin.php***



```

314 ▼ function tambah_peminjaman_act(){
315
316     $tanggal_pencatatan = date('Y-m-d H:i:s');
317     $anggota = $this->input->post('anggota');
318     $buku = $this->input->post('buku');
319     $tgl_pinjam = $this->input->post('tgl_pinjam');
320     $tgl_kembali = $this->input->post('tgl_kembali');
321     $denda = $this->input->post('denda');
322     $this->form_validation->set_rules('anggota', 'Anggota', 'required');
323     $this->form_validation->set_rules('buku', 'Buku', 'required');
324     $this->form_validation->set_rules('tgl_pinjam', 'Tanggal Pinjam', 'required');
325     $this->form_validation->set_rules('tgl_kembali', 'Tanggal Kembali', 'required');
326     $this->form_validation->set_rules('denda', 'Denda', 'required');
327
328 ▼ if($this->form_validation->run() != false){
329     $data = array(
330         'tgl_pencatatan' => $tanggal_pencatatan,
331         'id_anggota' => $anggota,
332         'id_buku' => $buku,
333         'tgl_pinjam' => $tgl_pinjam,
334         'tgl_kembali' => $tgl_kembali,
335         'denda' => $denda,
336         'tgl_pengembalian' => '0000-00-00',
337         'total_denda' => '0',
338         'status_pengembalian' => '0',
339         'status_peminjaman' => '0'
340     );
341
342     $this->m_perpus->insert_data($data, 'transaksi');
343
344     // update status buku yg di pinjam
345     $d = array('status_buku' => '0', 'tgl_input' => substr($tanggal_pencatatan, 0, 10));
346     $w = array('id_buku' => $buku);
347     $this->m_perpus->update_data($w, $d, 'buku');
348
349     redirect(base_url(). 'admin/peminjaman');
350 }else{
351     $w = array('status_buku' => '1');
352     $data['buku'] = $this->m_perpus->edit_data($w, 'buku')->result();
353     $data['anggota'] = $this->m_perpus->get_data('anggota')->result();
354
355     $this->load->view('admin/header');
356     $this->load->view('admin/tambah_peminjaman', $data);
357     $this->load->view('admin/footer');
358 }
359 }
360

```

Perhatikan pada action/aksi dari form peminjaman di atas. pertama kita menangkap data-data yang di kirim dari form ke dalam variabel masing-masing.

Kemudian kita setting form validation nya. Karena kita ingin form anggota, buku, tanggal pinjam, tanggal kembali dan denda wajib di isi(*required*). Dan jika validasi sesuai, maka kita input datanya ke table transaksi. Dengan menyesuaikan data yang di input sebagai berikut.

Setelah menginput data transaksi, kita juga akan mengubah status ketersediaan buku menjadi “0”. Seperti yang kita rancang sebelumnya. Jika status nya 1, berarti

buku tersedia. dan buku yang berstatus 0 berarti sedang di sewa atau sedang dipinjam. Jadi di sini kita mengubah status buku yang ingin di pinjam menjadi '0'.

Di sini saya menginput peminjaman baru, dengan mengisi anggota yang ingin melakukan peminjaman buku adalah siapa, buku yang ingin di sewa adalah membongkar rahasia web master, tanggal peminjamannya di mulai tanggal 29/10/2017 sampai dengan tanggal 31/10/2017 (selama 3 hari). Dengan biaya atau harga sebesar 500.000, dan denda perhari jika terlambat sebesar 100.000. dan klik simpan.

### 3. Membuat Pembatalan Peminjaman

Pada table transaksi di atas, kita telah membuat tombol/link yang mengarah ke method transaksi\_hapus.

```
<a class="btn btn-sm btn-danger" href="php echo base_url(). 'admin/transaksi_hapus/'. $t-&gt;transaksi_id; ?"><span class="glyphicon glyphicon-remove"></span> Batalkan Transaksi</a>
```

Pada method transaksi\_hapus ini kita akan membuat aksi untuk menghapus transaksi. Buat method baru lagi dengan nama *hapus\_peminjaman()* pada controller Admin.php.

**Application/controllers/Admin.php**

```
function hapus_peminjaman($id){
    $w = array('id_pinjam' => $id);
    $data = $this->m_perpus->edit_data($w, 'transaksi')->row();
    //$data = $this->m_perpus->edit_data($w, 'peminjaman')->row();
    $ww = array('id_buku' => $data->id_buku);
    $data2 = array('status_buku' => '1');
    $this->m_perpus->update_data($ww, $data2, 'buku');
    $this->m_perpus->delete_data($w, 'transaksi');
    redirect(base_url(). 'admin/peminjaman');
}
```

Pada method hapus\_peminjaman(), kita membuat aksi untuk menghapus mengubah kembali status buku ke "1". Dan menghapus data transaksi dari table peminjaman.

### 4. Transaksi Selesai

Pada saat tombol "transaksi\_selesai" di klik, maka halaman akan dialihkan ke method *transaksi\_selesai()*. Pada method transaksi\_selesai() ini, kita akan menampilkan sebuah form

baru, yaitu form untuk konfirmasi transaksi selesai. Buat method baru dalam controllers Admin.php. yaitu method *transaksi\_selesai()*.

```
function transaksi_selesai($id){
    $data['buku'] = $this->m_perpus->get_data('buku')->result();
    $data['anggota'] = $this->m_perpus->get_data('anggota')->result();
    $data['peminjaman'] = $this->db->query("select * from transaksi t, anggota a, buku b where t.id_buku =
        b.id_buku and t.id_anggota=a.id_anggota and t.id_pinjam='$id'")->result();

    $this->load->view('admin/header');
    $this->load->view('admin/transaksi_selesai',$data);
    $this->load->view('admin/footer');
}
```

Di sini kita memarsing data buku, data kostumer ke view transaksi\_selesai.php. kita juga memarsing data transaksi yang ingin di proses. Dan query untuk mendapatkan data transaksi yang ingin di proses/di konfirmasi adalah seperti berikut.

Ketiga data yang disebutkan di atas tadi kita parsing ke view transaksi\_selesai.php. Buat view baru dalam folder admin dengan nama *transaksi\_selesai.php*.

#### Application/views/admin/transaksi\_selesai.php

```
1 <div class="page-header">
2 <h3>Transaksi Selesai</h3>
3 </div>
4 <?php foreach($peminjaman as $p){ ?>
5 <form action="<?php echo base_url(). 'admin/transaksi_selesai_act' ?>" method="post">
6 <input type="hidden" name="id" value="<?php echo $p->id_pinjam ?>">
7 <input type="hidden" name="buku" value="<?php echo $p->id_buku ?>">
8 <input type="hidden" name="tgl_kembali" value="<?php echo $p->tgl_kembali ?>">
9 <input type="hidden" name="denda" value="<?php echo $p->denda ?>">
10 <div class="form-group">
11 <label>Anggota</label>
12 <select class="form-control" name="anggota" disabled>
13 <option value="">-Pilih Anggota-</option>
14 <?php foreach($anggota as $k){ ?>
15 <option <?php if($p->id_anggota == $k->id_anggota){echo "selected='selected'";} ?> value="<?php echo $k
    ->id_anggota; ?>"><?php echo $k->nama_anggota; ?></option>
16 <?php } ?>
17 </select>
18 </div>
19
20 <div class="form-group">
21 <label>Buku</label>
22 <select class="form-control" name="buku" disabled>
23 <option value="">-Pilih Buku-</option>
24 <?php foreach($buku as $m){ ?>
25 <option <?php if($p->id_buku == $m->id_buku){echo "selected='selected'";} ?> value="<?php echo $m->id_buku;
    ?>"><?php echo $m->judul_buku; ?></option>
26 <?php } ?>
27 </select>
28 </div>
29
```

```

29
30 <div class="form-group">
31 <label>Tanggal Pinjam</label>
32 <input class="form-control" type="date" name="tgl_pinjam" value="{<?php echo $p->tgl_pinjam ?>"
33 disabled>
34 </div>
35
36 <div class="form-group">
37 <label>Tanggal Kembali</label>
38 <input class="form-control" type="date" name="tgl_kembali" value="{<?php echo $p->tgl_kembali ?>" disabled>
39 </div>
40
41 <div class="form-group">
42 <label>Harga Denda / Hari</label>
43 <input class="form-control" type="text" name="denda" value="{<?php echo $p->denda ?>" disabled>
44 </div>
45
46 <div class="form-group">
47 <label>Tanggal Dikembalikan Oleh Anggota</label>
48 <input class="form-control" type="date" name="tgl_dikembalikan">
49 <?php echo form_error('tgl_dikembalikan'); ?>
50 </div>
51 <div class="form-group">
52 <input type="submit" value="Simpan" class="btn btnprimary btn-sm">
53 </div>
54 </form>
55 <?php } ?>

```

Buat method baru dengan nama *transaksi\_selesai\_act()* dalam controller *Admin.php*.  
**Application/controllers/Admin.php**

```

function transaksi_selesai_act(){
    $id = $this->input->post('id');
    $tgl_dikembalikan = $this->input->post('tgl_dikembalikan');
    $tgl_kembali = $this->input->post('tgl_kembali');
    $buku = $this->input->post('buku');
    $denda = $this->input->post('denda');
    $this->form_validation->set_rules('tgl_dikembalikan','Tanggal Di Kembali','required');
    if($this->form_validation->run() != false){
        // menghitung selisih hari
        $batas_kembali = strtotime($tgl_kembali);
        $dikembalikan = strtotime($tgl_dikembalikan);
        $selisih = abs(($batas_kembali - $dikembalikan)/(60*60*24));
        $total_denda = $denda*$selisih;
        // update status peminjaman
        $data = array('status_peminjaman' => '1', 'total_denda' => $total_denda,'tgl_pengembalian' => $
            tgl_dikembalikan,'status_pengembalian' => '1');
        // $data3 = array();
        $w = array('id_pinjam' => $id);
        $this->m_perpus->update_data($w,$data,'transaksi');
        // $this->m_perpus->update_data($w,$data3,'detail_pinjam');
        // update status buku
        $data2 = array('status_buku' => '1');
        $w2 = array('id_buku' => $buku);
        $this->m_perpus->update_data($w2,$data2,'buku');

        redirect(base_url(). 'admin/peminjaman');
    }else{
        $data['buku'] = $this->m_perpus->get_data('buku')->result();
        $data['anggota'] = $this->m_perpus->get_data('anggota')->result();
        $data['peminjaman'] = $this->db->query("select * from peminjaman p, anggota a, detail_pinjam d, buku b
            where p.id_anggota = a.id_anggota and p.id_pinjam = d.id_pinjam and d.id_buku = b.id_buku and
            p.id_pinjam='$id'")->result();

        $this->load->view('admin/header');
        $this->load->view('admin/transaksi_selesai',$data);
        $this->load->view('admin/footer');
    }
}
}

```

Selanjutnya kita mengubah format tanggal kembali dan tanggal buku di kembalikan. Tanggal kembali kita masukkan ke variabel \$batas\_kembali. Dan tanggal buku di kembalikan kita masukkan ke variabel \$dikembalikan. Selanjutnya kita menghitung selisih hari dari batas pengembelian buku dan tanggal buku di kembalikan oleh anggota. Selanjutnya kita menggunakan fungsi abs() berguna untuk membulatkan bilangan. Baru terakhir kita kalikan jumlah denda perhari dengan selisih hari pengembalian buku. Hasil dari perkalian ini adalah total denda yang harus di bayar oleh anggota (total denda = denda perhari X berapa hari terlambat). Dan selanjutnya kita update datanya yang ada di table “peminjaman”.

Terakhir kita juga mengupdate atau mengubah status buku yang di kembalikan menjadi kembali tersedia (1). Tentu teman-teman masih ingat bukan, kalau kolom status\_buku yang kita buat pada table buku untuk membuat status buku. Yang mana kita telah menentukan jika nilai nya 1 = tersedia dan 2 = sedang dipinjam.

**PERTEMUAN 11**  
**REVIEW MATERI**



**REVIEW MATERI**



**PERTEMUAN 12-14**  
**PRESENTASI PROJEK**



**PRESENTASI PROJEK**

## 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

