



MODUL WEB PROGRAMMING II



Disusun Oleh:

UNIT PENGEMBANGAN AKADEMIK

**UNIVERSITAS BINA SARANA INFORMATIKA
PROGRAM STUDI SISTEM INFORMASI
JAKARTA
2020**

KATA PENGANTAR

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

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

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

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

Jakarta, Agustus 2018

Tim Penulis

DAFTAR ISI

HALAMAN SAMPUL	Error! Bookmark not defined.
KATA PENGANTAR	ii
DAFTAR ISI.....	iii
PERTEMUAN 1 PENJELASAN KONTRAK KULIAH, PENEGASAN TUGAS	1
1.1 Penjelasan Kontrak Kuliah dan Penegasan Tugas	1
PERTEMUAN 2 PENGENALAN CODE IGNITER	2
2.1 Framework Web	3
2.2 Codeigniter	4
2.3 Instalasi Software	4
2.4 Instalasi Codeigniter.....	5
2.5 Struktur Folder Codeigniter.....	6
PERTEMUAN 3 MODEL, VIEW, CONTROLLER.....	12
3.1 Pengertian MVC.....	12
3.2 Menentukan Kontroler Standar (Default).....	17
3.3 Menghilangkan index.php pada codeigniter	17
3.4 Aturan merancang Model, View, dan Controller	19
3.5 Parsing Data (mengirimkan nilai) dari controller Ke View	20
PERTEMUAN 4 MEMBUAT TEMPLATE (TEMPLATING) WEB SEDERHANA	24
4.1 Membuat Template Sederhana yang dinamis	25
4.2 Seting base_url () pada codeigniter	30
PERTEMUAN 5 MEMBUAT FORM VALIDASI PADA CODEIGNITER.....	35
5.1 Form Validasi	35
5.2 Mekanisme Validasi Data di dalam Form	35
5.3 Memuat Library Form_validation	36
PERTEMUAN 6 STUDI KASUS PUSTAKA-BOOKING	38
6.1 Persiapan Membuat Aplikasi pustaka-booking	39
6.2 Setting Base URL	41
6.3 Merancang & Membuat Database Aplikasi pustaka-booking dan Konfigurasi database di Codeigniter	42
6.4 Membuat Koneksi Database Codeigniter dengan MySQL	44
6.5 Membuat Model untuk CRUD	46
PERTEMUAN 7 REVIEW MATERI	49
PERTEMUAN 8 UJIAN TENGAH SEMESTER.....	51

PERTEMUAN 9 MEMBUAT FORM LOGIN DAN PESAN NOTIFIKASI	52
9.1 Membuat Halaman Login.....	52
9.2 Membuat Halaman Admin	62
PERTEMUAN 10 MEMBUAT FORM REGISTRASI USER	74
10.1 Membuat helper.....	75
10.2 Membuat Form Registrasi	81
10.3 Membuat Halaman My Profile dan Ubah Profile	86
PERTEMUAN 11 MEMBUAT KATEGORI BUKU	92
11.1 Membuat Tampil Data Kategori.....	93
11.2 Hapus data kategori	96
PERTEMUAN 12 MEMBUAT CRUD DATA BUKU	97
12.1 Menampilkan Data Buku dan tambah Buku	98
12.2 Update Data Buku	107
12.3 Menghapus Data Buku.	110
PERTEMUAN 13-15 PRESENTASI PROJEK	111
PERTEMUAN 16 UJIAN AKHIR SEMESTER	112
DAFTAR PUSTAKA	113

PERTEMUAN 1

PENJELASAN KONTRAK KULIAH, PENEGASAN TUGAS

Deskripsi Pembelajaran

Pada bab ini dijelaskan bagaimana proses pembelajaran berlangsung setiap pertemuannya yaitu mahasiswa diarahkan untuk mandiri dengan diberikan tugas individu di setiap pertemuan yang mana tugas tersebut berupa mengimplementasikan dan mengerjakan tiap materi dan latihan yang ada pada pertemuan berikutnya. Sehingga pada saatnya pertemuan tersebut berlangsung, mahasiswa sudah mengetahui materi yang ada dan bisa fokus berdiskusi pada bagian materi yang masih kurang paham.

Tujuan Pembelajaran

Pada bab ini diharapkan mahasiswa dapat memahami kontrak perkuliahan yang harus diikuti dan jenis tugas yang akan diberikan selama satu semester ke depan.

1.1 Penjelasan Kontrak Kuliah dan Penegasan Tugas

A. Tugas Projek (Kelompok)

1. Tugas project diadakan untuk memperoleh nilai dan pengganti UTS dan UAS. Tugas ini dikerjakan secara kelompok dengan 1 kelompok sebanyak 5 mahasiswa atau disesuaikan dengan jumlah mahasiswa dalam satu kelas.
2. Tugas dan Kelompok harus sesuai dan disinkronkan dengan tugas mata kuliah SIM, PSBO, dan WP2.
3. Bentuk tugas projek adalah mencari aplikasi web nativ (tidak menggunakan framework) yang sudah jadi kemudian di transformasikan menjadi aplikasi web framework CodeIgniter dan membuat analisa kebutuhan user dan sistem dari web tersebut.
4. Aplikasi web nativ harus ditunjukan ke dosen pengampu matakuliah, sebagai bukti bahwa aplikasi web nativ memang ada dan jalan.
5. Aplikasi web nativ diperbolehkan didapat dari berbagai sumber.

B. Tugas Mingguan (individu)

1. Tugas mingguan mulai dikerjakan di minggu pertama untuk disetorkan dan didemonstrasikan di minggu berikutnya. Begitu seterusnya di setiap minggu.

2. Bentuk tugas mingguan adalah mengimplementasikan materi tiap pertemuan yang ada di modul dimulai dari materi pertemuan ke 2 sampai materi pertemuan terakhir.

Tugas Pertemuan 1

- a. Mahasiswa mengerjakan dan materi yang ada di pertemuan 2 yaitu instalasi Codeigniter dan mengerjakan contoh 1 yang ada pada modul halaman 3-4 dan halaman 7.
- b. Demonstrasikan dan kumpulkan hasilnya di pertemuan selanjutnya.

PERTEMUAN 2

PENGENALAN CODE IGNITER

Deskripsi Pembelajaran

Pada bab ini akan dibahas tentang framework dan jenis-jenisnya, codeigniter, pemasangan software yang dibutuhkan, cara memasang dan menggunakan codeigniter, struktur dari codeigniter, file konfigurasi codeigniter, kontroler standar, dan latihan pengenalan.

Tujuan Pembelajaran

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.

2.1 Framework Web

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

Framework web untuk PHP:

- ✓ Codeigniter
- ✓ Yii
- ✓ Slim framework
- ✓ Zend framework
- ✓ Laravel
- ✓ Symfony
- ✓ CakePHP
- ✓ Phalcon
- ✓ Kohana

✓ FuelPHP, dll

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

2.2 Codeigniter

Codeigniter adalah framework web untuk bahasa pemrograman PHP, yang dibuat oleh Rick Ellis pada tahun 2006, penemu dan pendiri EllisLab (www.ellislab.com). 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.

2.3 Instalasi Software

Software yang dibutuhkan pada pembelajaran kali ini yaitu:

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

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

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

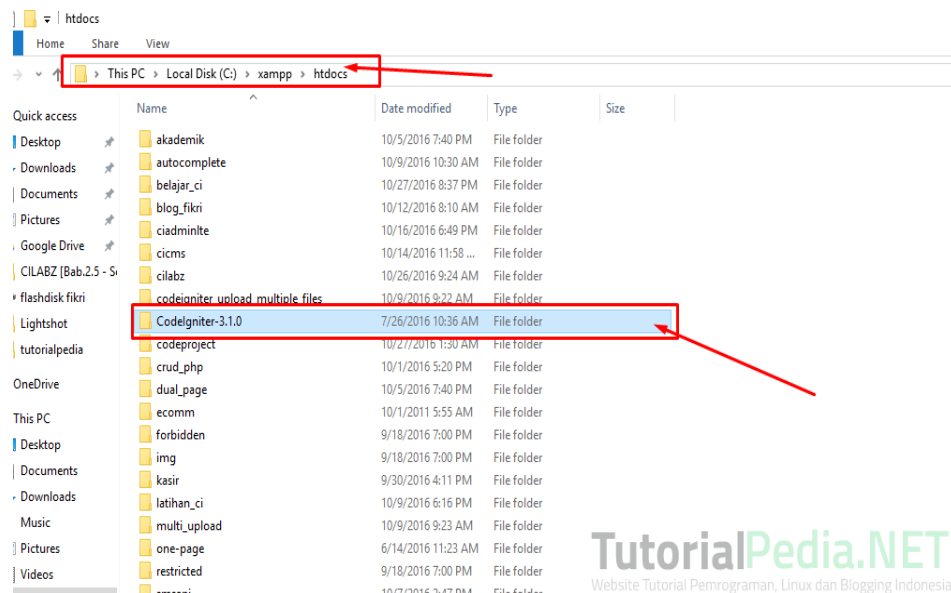
Semua software yang dibutuhkan diatas dapat anda download melalui link berikut:



https://drive.google.com/open?id=1RrGuGgMBiGRc_Qa8N905yA7w0LotrkNc

2.4 Instalasi Codeigniter

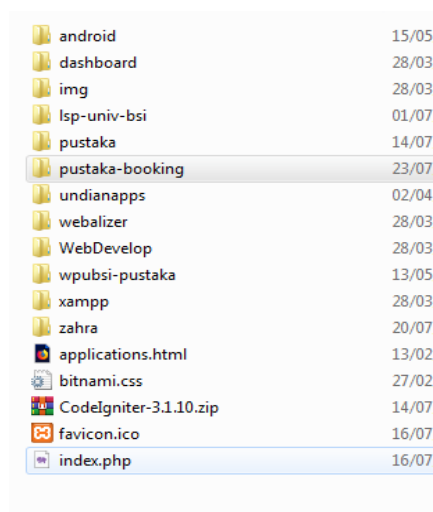
- a. Download Package codeigniter pada situ resminya yang sudah disebutkan di atas.
- b. Saat ini versi stabil dari codeigniter adalah 4.0.3



Sumber: www.tutorialpedia.net

Gambar 1. Penempatan Folder Codeigniter

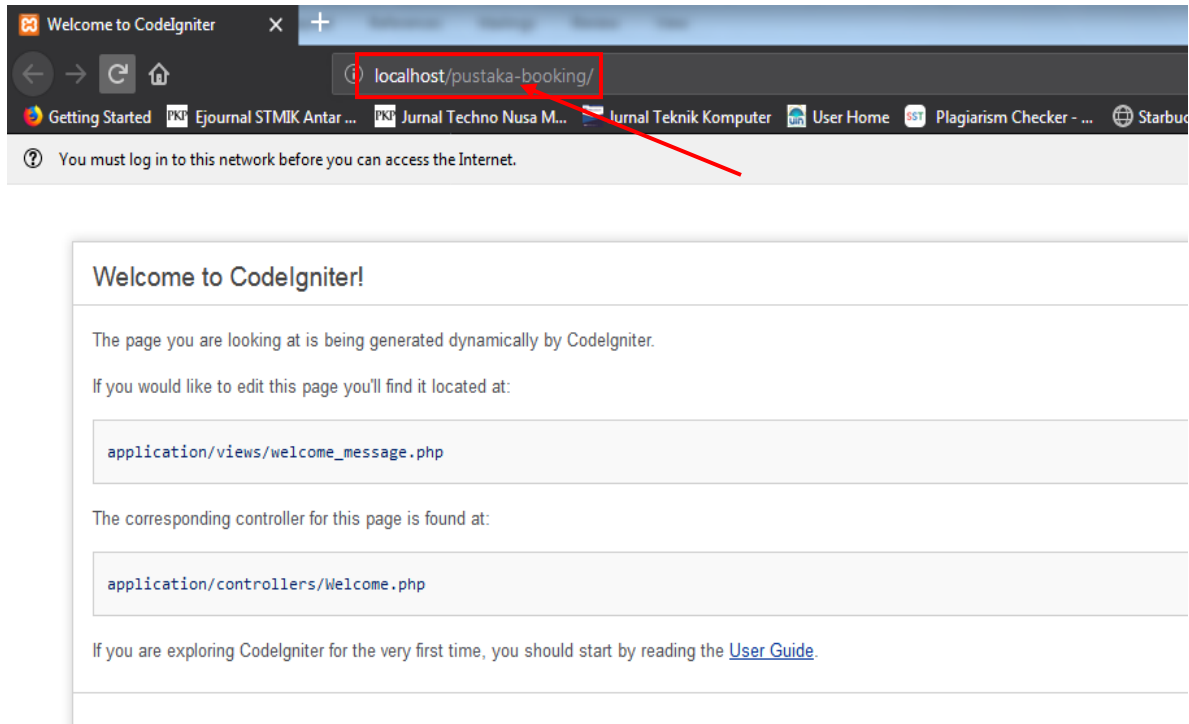
- c. Setelah didownload, ekstrak file codeigniter pada direktori **C:\xampp\htdocs**
- d. lalu ubah folder **Codeigniter** menjadi sesuai keinginan kita. disini saya akan ubah folder codeigniter tersebut menjadi **pustaka-bookng**.



Sumber: www.tutorialpedia.net

Gambar 2. Mengubah Nama Folder Root

- e. Setelah itu, sekarang coba akses folder tersebut melalui browser dengan mengetikkan url **http://localhost/pustaka-booking**. jangan lupa menyalakan **XAMPP** terlebih dahulu.



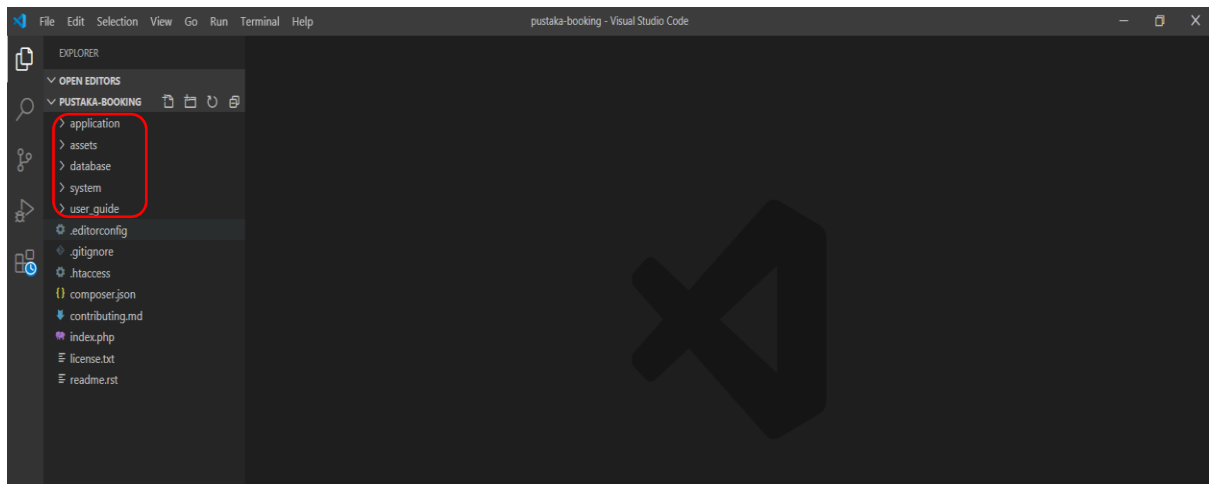
Sumber: www.tutorialpedia.net

Gambar 3. Tampilan Awal

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

2.5 Struktur Folder Codeigniter

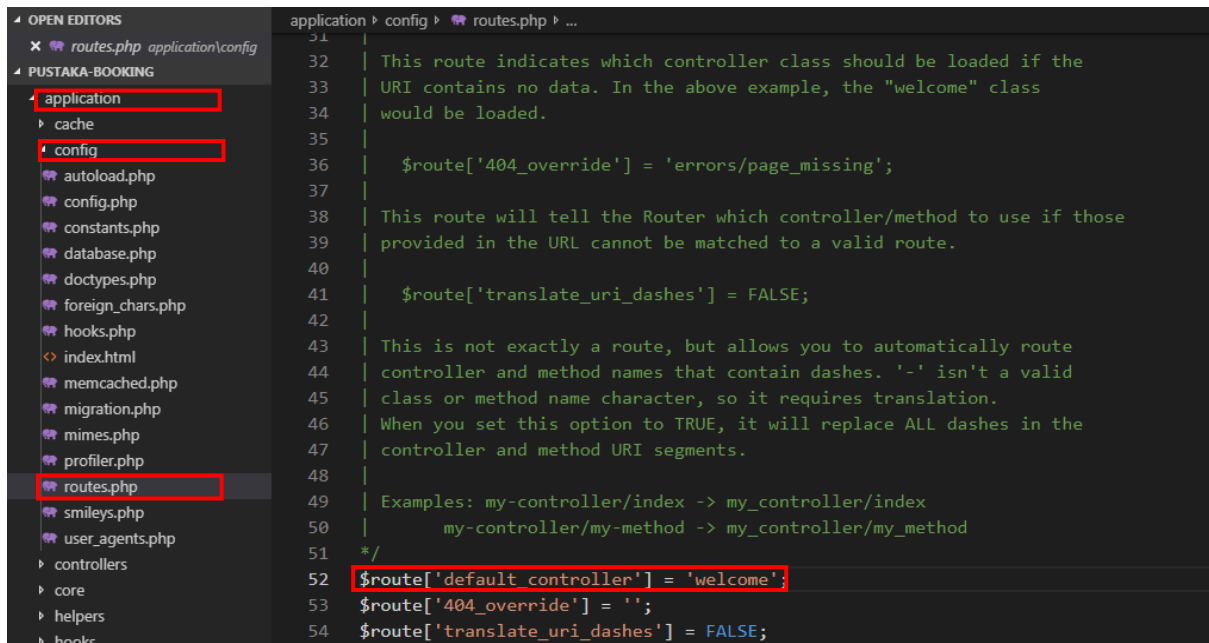
Di dalam folder codeigniter terdapat 3 folder utama yaitu **Application**, **System**, **User Guide**. Folder application berguna sebagai tempat untuk mengembangkan aplikasi web nanti, Folder **system** berisi library atau kumpulan fungsi-fungsi dasar Codeigniter (CI), sedangkan **User Guide** berisi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



Sumber: www.tutorialpedia.net

Gambar 4. Struktur Folder Codeigniter

Pertama kali kita fokus pada folder **application/config** di dalam folder **config** ini terdapat file-file konfigurasi utama dan pertama kali untuk memudahkan pembuatan dan pengembangan semua file yang akan dibuat 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 5).



Sumber: www.tutorialpedia.net

Gambar 5. Default Controller Codeigniter

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.

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

class Welcome extends CI_Controller {

    /**
     * Index Page for this controller.
     *
     * Maps to the following URL
     *      http://example.com/index.php/welcome
     *  - or -
     *      http://example.com/index.php/welcome/index
     *  - or -
     * Since this controller is set as the default controller in
     * config/routes.php, it's displayed at http://example.com/
     *
     * So any other public methods not prefixed with an underscore will
     * map to /index.php/welcome/<method_name>
     * @see https://codeigniter.com/user_guide/general/urls.html
     */
    public function index()
    {
        $this->load->view('welcome_message');
    }
}
```

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.

```
<?php
defined('BASEPATH') or exit('No direct script access allowed');
?>
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <title>Welcome to CodeIgniter</title>

    <style type="text/css">
        ::selection {
```

```

        background-color: #E13300;
        color: white;
    }

    ::-moz-selection {
        background-color: #E13300;
        color: white;
    }

    body {
        background-color: #fff;
        margin: 40px;
        font: 13px/20px normal Helvetica, Arial, sans-serif;
        color: #4F5155;
    }

    a {
        color: #003399;
        background-color: transparent;
        font-weight: normal;
    }

    h1 {
        color: #444;
        background-color: transparent;
        border-bottom: 1px solid #D0D0D0;
        font-size: 19px;
        font-weight: normal;
        margin: 0 0 14px 0;
        padding: 14px 15px 10px 15px;
    }

    code {
        font-family: Consolas, Monaco, Courier New, Courier, monospace;
        font-size: 12px;
        background-color: #f9f9f9;
        border: 1px solid #D0D0D0;
        color: #002166;
        display: block;
        margin: 14px 0 14px 0;
        padding: 12px 10px 12px 10px;
    }

    #body {
        margin: 0 15px 0 15px;
    }

    p.footer {

```

```

        text-align: right;
        font-size: 11px;
        border-top: 1px solid #D0D0D0;
        line-height: 32px;
        padding: 0 10px 0 10px;
        margin: 20px 0 0 0;
    }

    #container {
        margin: 10px;
        border: 1px solid #D0D0D0;
        box-shadow: 0 0 8px #D0D0D0;
    }
</style>
</head>

<body>

    <div id="container">
        <h1>Welcome to CodeIgniter!</h1>

        <div id="body">
            <p>The page you are looking at is being generated dynamically by C
odeIgniter.</p>

            <p>If you would like to edit this page you'll find it located at:<
/p>

            <code>application/views/welcome_message.php</code>

            <p>The corresponding controller for this page is found at:</p>
            <code>application/controllers/Welcome.php</code>

            <p>If you are exploring CodeIgniter for the very first time, you s
hould start by reading the <a href="user_guide/">User Guide</a>.</p>
        </div>

        <p class="footer">Page rendered in <strong>{elapsed_time}</strong> sec
onds. <?php echo (ENVIRONMENT === 'development') ? 'CodeIgniter Version <stro
ng>' . CI_VERSION . '</strong>' : '' ?></p>
    </div>

</body>

</html>

```

Contoh menampilkan Biodata

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

```
<?php
class Contoh1 extends CI_Controller
{
    public function index()
    {
        echo "<h1>Perkenalkan</h1>";
        echo "Nama saya Imam Nawawi
            Saya tinggal di daerah Ciputat
            olah raga yang saya sukai adalah
            Bulutangkis";
    }
}
```

Untuk melihat hasilnya silahkan ketik URL berikut **localhost/pustaka-booking/index.php/contoh1**

Tugas Pertemuan 2

- a. Mahasiswa mengerjakan dan mengimplementasikan materi yang ada di pertemuan 3 yaitu mengerjakan contoh1, contoh2, contoh3, dan menghilangkan index.php yang ada di modul halaman 14-17.
- b. Demonstrasikan dan kumpulkan hasilnya di pertemuan selanjutnya.

PERTEMUAN 3

MODEL, VIEW, CONTROLLER

Deskripsi Pembelajaran

Pembelajaran pertemuan ini meliputi tentang pengertian, maksud dan konsep daripada MVC disertai contoh penggunaan mulai dari controller saja, kemudian controller dengan view, dan contoh menggunakan controller, view, dan model. Pada pertemuan ini juga dijelaskan tentang aturan dalam membuat MVC, menentukan controller standar default, menghilangkan index.php, pembuatan file **.htaccess** dan bagaimana mekanisme pengiriman nilai antara Model, view dan controller.

Tujuan Pembelajaran

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

3.1 Pengertian MVC

Pada pengertian codeigniter di atas tadi dijelaskan 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 kumpulkan 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 disimpulkan 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 ditampilkan. 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 dilakukan dengan cepat dan terstruktur.

Bentuk umum model seperti berikut:

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

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

```
Class Nama_model extends CI_model {  
    //konstruktor kelas model
```

```

Function __construct() {
    //memanggil konstruktor kelas CI_model
    $parent: : __construct();
    //...
}
// ...
}

```

Contoh1 penggunaan hanya controller

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

```

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

```

Untuk melihat hasilnya ketikkan url **localhost/pustaka-booking/latihan1**

Contoh2 menggunakan Controller dan Model

Model **Model_latihan1.php** simpan di **application\models**

```

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

    //method penjumlahan
    public function jumlah($n1 = null, $n2 = null)
    {

```

```

        $this->nilai1 = $nil1;
        $this->nilai2 = $nil2;
        $this->hasil = $this->nilai1 + $this->nilai2;
        return $this->hasil;
    }
}

```

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

```

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

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

```

Untuk melihat hasilnya ketikkan url

localhost/pustaka-booking/index.php/latihan1/penjumlahan/2/6

Contoh3 menggunakan Controller, View dan Model

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

```

<html>

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

<body>

```

```

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

</body>

</html>

```

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

```

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

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

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

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

```

Untuk melihat hasilnya ketikkan url

localhost/pustaka-booking/index.php/latihan1/penjumlahan/2/6

3.2 Menentukan Kontroler Standar (Default)

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

Contoh penulisan URI:

`http://localhost/pustaka-booking/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 pada gambar 3 pada pertemuan 2:

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

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

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

3.3 Menghilangkan index.php pada codeigniter

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

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

Kemudian hapus index.php pada pengaturan sehingga menjadi

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

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

File **pustaka-booking/.htaccess**

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

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

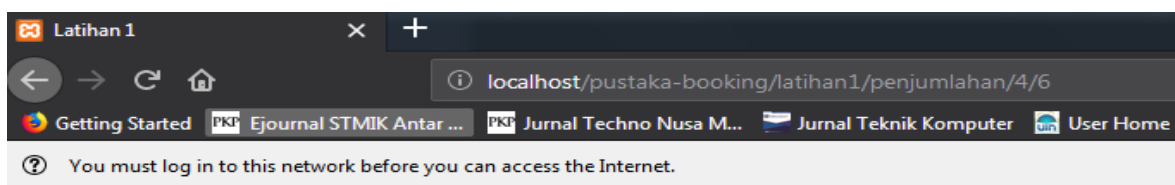
Jika biasanya untuk mengakses method dalam kontroler Latihan1 diatas harus dengan mengetikkan

<http://localhost/pustaka-booking/index.php/latihan1/penjumlahan/4/6>

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

<http://localhost/pustaka-booking/latihan1/penjumlahan/4/6>

berikut hasil dalam browser



3.4 Aturan merancang Model, View, dan Controller

a. Model

Secara umum perancangan model dapat dibuat menggunakan aturan berikut:

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

b. View

Berikut ini aturan yang dapat digunakan untuk merancang view.

- 1) View harus berisi kode kode yang bersifat presentasional, biasanya berupa kode HTML. View juga dapat berisi perintah perintah PHP sederhana yang masih berkaitan dengan proses pembuatan 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 model dan mengarahkan ke view (jika model bersangkutan menghasilkan data yang perlu ditampilkan ke user)
- 3) Controller seharusnya tidak berisi 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 HTML yang diperlukan untuk tujuan presentasi. Tugas ini sebaiknya dilakukan oleh view.

3.5 Parsing Data (mengirimkan nilai) dari controller Ke View

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

Perhatikan contoh cara pengiriman nilai ke view codeigniter berikut ini.

Contoh5 mengirimkan nilai dari controller ke view

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

```
<html>

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

<body>
    <center>
        <form    action="<?=    base_url('matakuliah/cetak');    ?>"
method="post">
            <table>
                <tr>
                    <th colspan="3">
                        Form Input Data Mata Kuliah
                    </th>
                </tr>
                <tr>
                    <td colspan="3">
```



```

        <hr>
    </td>
</tr>
<tr>
    <th>Kode MTK</th>
    <th>:</th>
    <td>
        <input type="text" name="kode" id="kode">
    </td>
</tr>
<tr>
    <th>Nama MTK</th>
    <td>:</td>
    <td>
        <input type="text" name="nama" id="nama">
    </td>
</tr>
<tr>
    <th>SKS</th>
    <td>:</td>
    <td>
        <select name="sks" id="sks">
            <option value="">Pilih SKS</option>
            <option value="2">2</option>
            <option value="3">3</option>
            <option value="4">4</option>
        </select>
    </td>
</tr>
<tr>
    <td colspan="3" align="center">
        <input type="submit" value="Submit">
    </td>
</tr>
</table>
</form>
</center>
</body>

</html>

```

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

```
<?php
class Matakuliah extends CI_Controller
{

    public function index()

    {

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

    }

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

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

    }

}
```

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

```
<html>

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

```

<body>
  <center>
    <table>
      <tr>
        <th colspan="3">
          Tampil Data Mata Kuliah
        </th>
      </tr>
      <tr>
        <td colspan="3">
          <hr>
        </td>
      </tr>
      <tr>
        <th>Kode MTK</th>
        <th>:</th>
        <td>
          <?= $kode; ?>
        </td>
      </tr>
      <tr>
        <td>Nama MTK</td>
        <td>:</td>
        <td>
          <?= $nama; ?>
        </td>
      </tr>
      <tr>
        <td>SKS</td>
        <td>:</td>
        <td>
          <?= $sks; ?>
        </td>
      </tr>
      <tr>
        <td colspan="3" align="center">
          <a href="<?= base_url('matakuliah');
?>">Kembali</a>
        </td>
      </tr>
    </table>
  </center>

```

```
</body>
```

```
</html>
```

Tugas pertemuan 3

- a. Mahasiswa mengerjakan dan mengimplementasikan materi yang ada di pertemuan 4 yaitu membuat template sederhana dan dinamis dan setting `base_url()` yang ada pada modul halaman 26 dan 30
- b. Demonstrasikan dan kumpulkan hasilnya di pertemuan selanjutnya

PERTEMUAN 4

MEMBUAT TEMPLATE (TEMPLATING) WEB SEDERHANA

Deskripsi Pembelajaran

Pertemuan 4 berisi tentang pembahasan dalam membuat template sederhana yang dinamis dengan membagi satu file view index.php menjadi beberapa file view. Kemudian ada pembahasan tentang pengaturan base_url dan konfigurasi yang berkaitan dengan url.

Tujuan Pembelajaran

Pada pertemuan ini, mahasiswa diharapkan mampu membuat template sederhana. Mahasiswa juga diharapkan mampu membuat template web dinamis, yaitu dengan cara membuat halaman yang dipecah-pecah agar dapat dengan mudah untuk me-load halaman-halaman tersebut sesuai dengan keinginan.

4.1 Membuat Template Sederhana yang dinamis

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

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

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

class Web extends CI_Controller{

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

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

```

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

```

Kemudian buatlah 3 buah file view beri nama **v_header.php**, **v_index.php**, dan **v_footer.php** lalu simpan dalam folder **application/views/** kemudian ketik script berikut:

v_header.php

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Web Prog II | Merancang Template sederhana dengan
codeigniter</title>
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url() ?>assets/css/stylebuku.css">
</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>
    </div>

```

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">

Model

<p align='justify'>Model adalah kelas yang merepresentasikan atau memodelkan tipe data yang akan digunakan oleh aplikasi. Model juga dapat didefinisikan sebagai bagian penanganan yang berhubungan dengan pengolahan atau manipulasi database. Seperti misalnya mengambil data dari database, menginput dan pengolahan database lainnya. Semua intruksi atau fungsi yang berhubungan dengan pengolahan database diletakkan di dalam model. Sebagai contoh, jika ingin membuat aplikasi untuk menghitung luas dan keliling lingkaran, maka dapat memodelkan objek lingkaran sebagai kelas model.</p>

<p align='justify'>Sebagai catatan, Semua model harus disimpan di dalam folder application\models</p>

View

<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>

Controller

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

<p align='justify'>Dari penjelasan tentang model view dan controller di atas dapat di simpulkan bahwa controller sebagai penghubung view dan model. Misalnya pada aplikasi yang menampilkan data dengan menggunakan metode konsep mvc, controller memanggil intruksi pada model yang mengambil data pada database, kemudian controller yang meneruskannya pada view untuk di tampilkan. Jadi jelas sudah dan sangat mudah dalam pengembangan aplikasi dengan cara mvc ini karena web designer atau front-end developer tidak perlu lagi berhubungan dengan controller, dia hanya perlu berhubungan dengan view untuk mendesign tampilann aplikasi, karena back-end developer yang

menangani bagian controller dan modelnya. Jadi pembagian tugas pun menjadi mudah dan pengembangan aplikasi dapat di lakukan dengan cepat dan terstruktur.</p>

</section>

V_footer.php

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

Kemudian buatlah **stylebuku.css** untuk membuat tampilan lebih bagus dan simpan di dalam folder root seperti berikut: **pustaka-boooking/assets/css/**

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

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

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

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

#wrapper header nav{
```



```

float: right;
margin-top: 50px;
}

#wrapper header nav ul{
padding: 0;
margin: 0;
}

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

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

.clear{
clear: both;
}

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

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

section{
padding: 20px;
}

```

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

4.2 Seting base_url () pada codeigniter

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

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

Kemudian diubah menjadi

```
$config['base_url'] = 'http://localhost/pustaka-booking/';
```

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

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

Hasil perintah di atas akan sama seperti ini

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

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

```
<?php  
defined('BASEPATH') or exit ('no direct script access allowed');  
  
class Web extends CI_Controller{  
  
    function __construct(){  
        parent::__construct();  
        $this->load->helper('url');  
    }  
  
    public function index(){
```

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

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

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

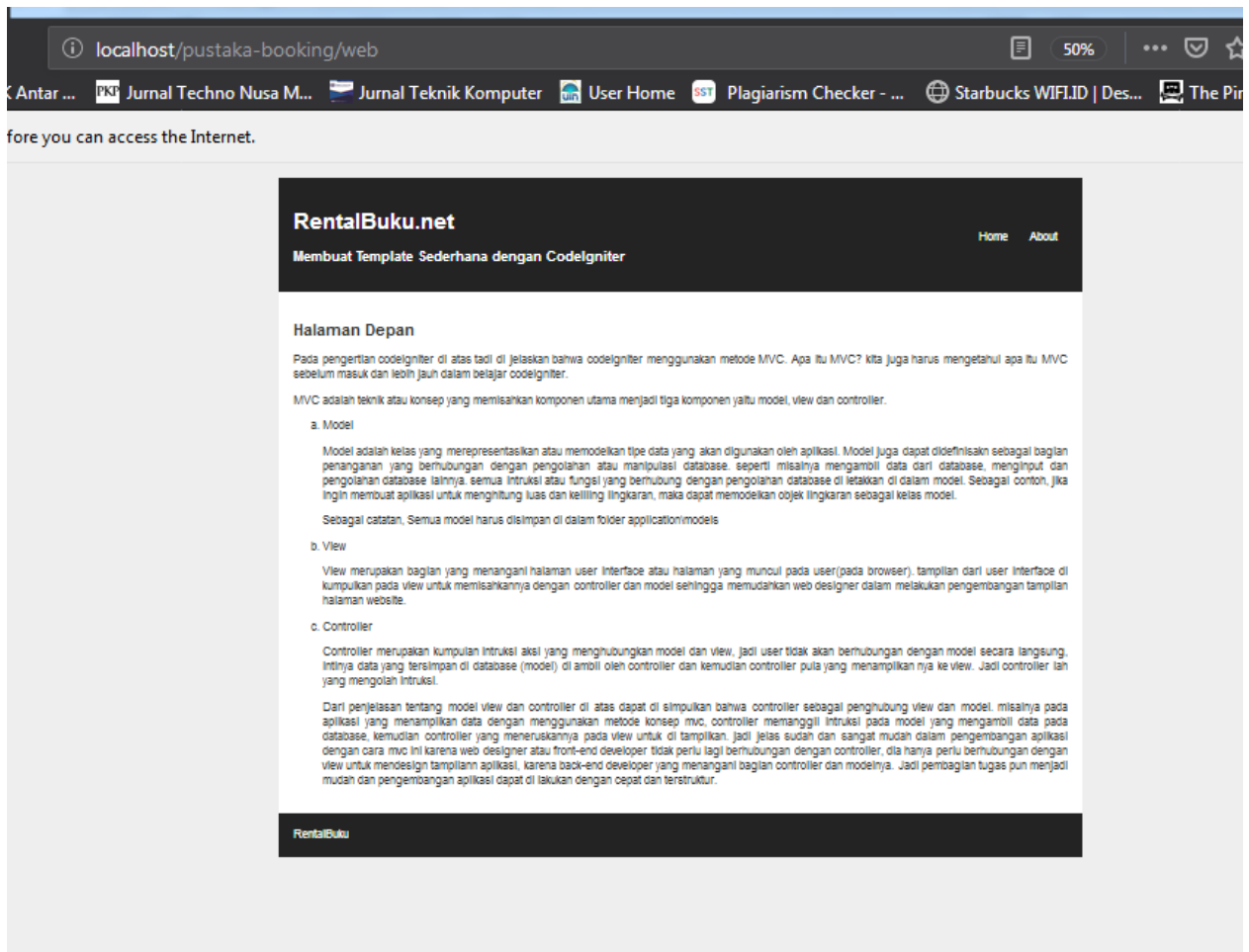
Kemudian diubah dengan ditambahkan kata 'url' menjadi seperti di bawah

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

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

```
http://localhost/pustaka-booking/web
```

hasilnya seperti gambar di bawah.



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

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

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

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

```
<section>
<h1><?php echo $judul ?></h1>
```

```

<h4>Nama</h4>
<ul type="disc">
    <li>Nama Depan : Imam</li>
    <li>Nama Belakang : Nawawi</li>
</ul>
<br>
<h4>Alamat</h4>
<ul type="none">
    <li> Jalan Ciledug Raya No. 168 Pesanggrahan</li>
</ul>

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

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

```

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

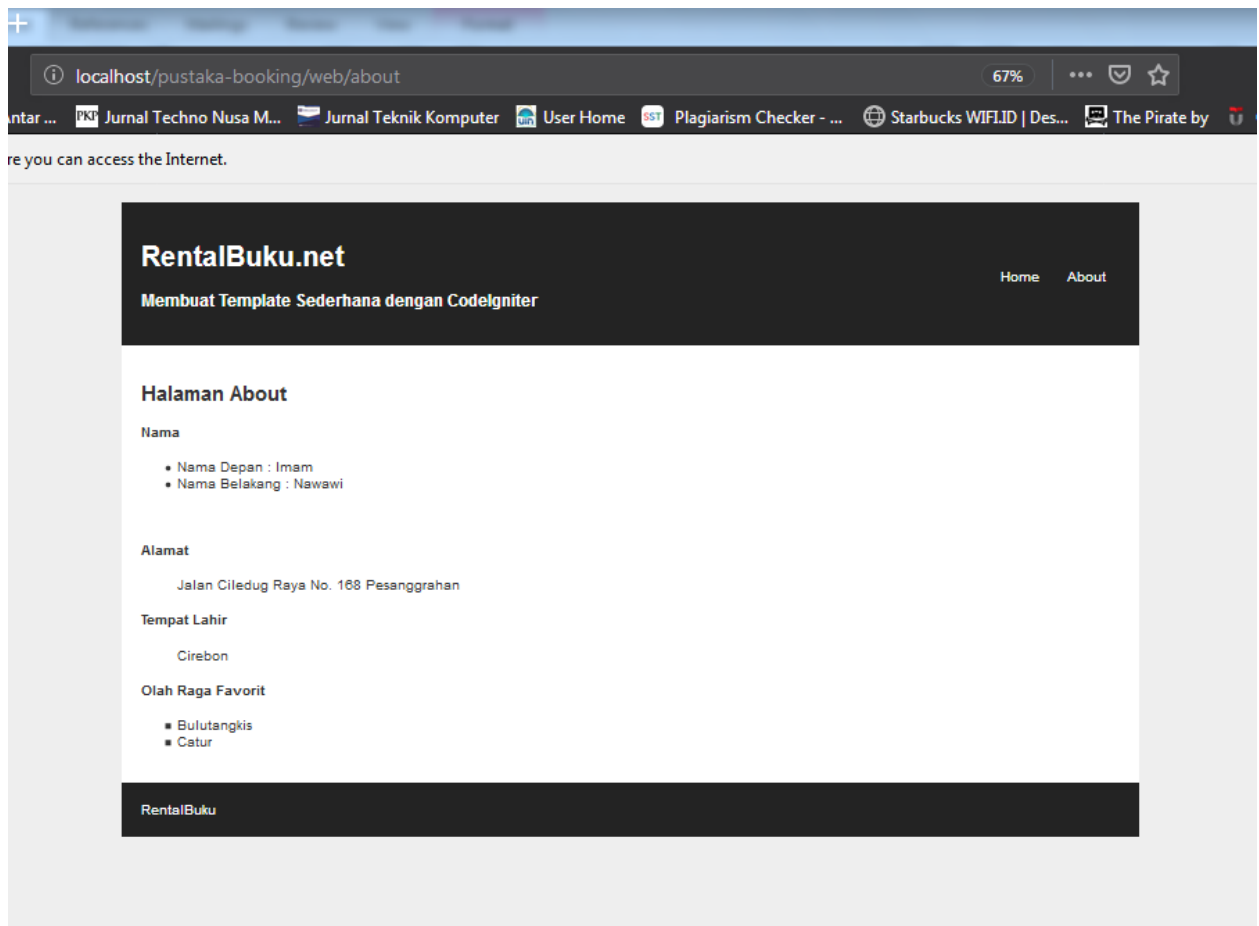
```

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

```

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

http://localhost/pustaka-booking/web/about



Tugas pertemuan 4

- c. Mahasiswa mengerjakan dan mengimplementasikan materi yang ada di pertemuan 5 yaitu menambahkan validasi data dengan form validasi pada controller matakuliah. Modul halaman 35
- d. Demonstrasikan dan kumpulkan di pertemuan selanjutnya

PERTEMUAN 5

MEMBUAT FORM VALIDASI PADA CODEIGNITER

Deskripsi Pembelajaran

Pembahasan yang ada pada pertemuan ini adalah tentang pembuatan validasi untuk memasukkan data ke dalam database dengan menggunakan library `form_validation`.

Tujuan Pembelajaran

Pada pertemuan ini, mahasiswa diharapkan mampu memahami dan membuat form untuk input data dengan disertai validasi. Agar dalam memasukkan data ke database menjadi lebih akurat dan tidak terjadi kesalahan. Mahasiswa juga diharapkan memahami penggunaan library untuk form validasi.

5.1 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

5.2 Mekanisme Validasi Data di dalam Form

- a. Form ditampilkan ke halaman web
- b. User mengisi data ke dalam form tersebut
- c. Jika terdapat satu atau beberapa data yang salah, maka form akan ditampilkan ulang, biasanya disertai dengan pesan kesalahan.

- d. Proses ini akan diulang sampai semua data yang diminta di dalam form diisi dengan benar.

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

5.3 Memuat Library Form_validation

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

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

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

Buka file tersebut lalu cari baris sintak seperti berikut:

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

Kemudian diubah dengan ditambahkan kata form_validation seperti berikut:

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

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

```
<?php
class Matakuliah extends CI_Controller
{

    public function index()
    {

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

    public function cetak()
    {
```



```

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

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

        if ($this->form_validation->run() != true) {
            $this->load->view('view-form-matakuliah');
        } else {
            $data = [
                'kode' => $this->input->post('kode'),
                'nama' => $this->input->post('nama'),
                'sks' => $this->input->post('sks')
            ];

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

```

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

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

http://localhost/malasngoding/user_guide/libraries/form_validation.html#the-form

Tugas pertemuan 5

- a. Mahasiswa mengerjakan materi yang ada di pertemuan 6 yaitu persiapan dalam membuat aplikasi pustaka booking yang berisi setting base_url, membuat database, mengatur koneksi database, membuat model user dan model buku. Ada di modul halaman 41,42,44 dan 45
- b. Demonstrasikan dan kumpulkan di pertemuan selanjutnya

PERTEMUAN 6

STUDI KASUS PUSTAKA-BOOKING

Deskripsi Pembelajaran

Pertemuan ini membahas bagaimana persiapan untuk membuat sebuah aplikasi web. Dari mulai Analisa kebutuhan pengguna dan sistem, desain dan pembuatan database, dan pembuatan model yang diperlukan untuk dapat menambah, mengubah, membaca, dan menghapus data pada database.

Tujuan Pembelajaran

Pada pertemuan ini mahasiswa diharapkan dapat memahami persiapan dasar untuk memulai sebuah projek kasus. Mahasiswa juga diharapkan mampu memahami desain dan pembuatan database, koneksi database dan membuat model untuk Create Read Update Delete (CRUD) data untuk membuat aplikasi pustaka-booking.

6.1 Persiapan Membuat Aplikasi pustaka-booking

Pustaka-booking merupakan aplikasi reservasi atau booking peminjaman buku pada sebuah perpustakaan yang dilakukan secara online dengan melibatkan anggota, admin dan aplikasi pustaka-booking. Aplikasi pustaka-booking ini dibagi menjadi 2 jenis tampilan yaitu tampilan bagian backend dan tampilan frontend. Untuk pembahasan Web Programming 2 ini hanya membahas tampilan backend saja, sedangkan untuk frontend nya akan di bahas di mata kuliah web programming 3 di semester mendatang.

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

Alur logika sistem booking pustaka ini yaitu seseorang yang ingin melakukan booking diharuskan mendaftar menjadi anggota terlebih dahulu, selanjutnya ketika sudah menjadi

anggota, baru dapat melakukan booking terhadap buku yang akan dipinjam. Setelah melakukan booking, anggota diharuskan mengambil buku yang telah diboeking dengan cara datang langsung ke perpustakaan dalam waktu 1x24 jam. Kemudian konfirmasi ke petugas atau admin untuk diambilkan buku yang telah diboeking berdasarkan bukti booking melalui aplikasi.

Pustaka-booking secara utuh memiliki kebutuhan sebagai berikut:

Kebutuhan user:

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

Kebutuhan sistem:

- a. Sistem akan melakukan validasi username dan password ketika ada seorang user yang melakukan login
- b. Sistem juga melakukan validasi data buku dan anggota yang akan diinput ke dalam database

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

Persiapan untuk memulai membangun aplikasi pustaka-booking 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 pustaka-booking. Semua pengaturan atau penyesuaian yang dilakukan di sini sesuai dengan kebutuhan untuk membuat aplikasi pustaka-booking dengan codeigniter. diantaranya adalah:

6.2 Setting Base URL

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

pustaka-booking/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 pustaka-booking ini. Untuk menjalankan atau membuka library dan helper secara otomatis di codeiginter, kita bisa mengaturnya dalam file autoload.php dalam folder *application/config/autoload.php*.

Pustaka-booking/application/config/autoload.php

Library yang akan kita load secara otomatis adalah *database*, *session* dan *form_validation*.

Cari sintak

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

Kemudian tambahkan database, session dan form_validation. Sehingga menjadi seperti berikut:

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

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

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

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

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

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

6.3 Merancang & Membuat Database Aplikasi pustaka-booking dan Konfigurasi database di Codeigniter

Berikut adalah gambaran struktur database yang akan kita buat untuk aplikasi pustaka-booking ini.

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

a. Tabel User (**user**)

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

b. Tabel Role (**role**)

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

Isi tabel role

id	role
1	administrator
2	member

c. Tabel Kategori (**kategori**)

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

Isi tabel kategori

id	kategori
1	Komputer
2	Bahasa
3	Sains
4	Hobby
5	Komunikasi
6	Hukum
7	Agama
8	Populer
9	Komik

d. Tabel Buku (**buku**)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	judul_buku	varchar(128)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 3	id_kategori	int(11)			No	None			Change Drop More
<input type="checkbox"/> 4	pengarang	varchar(64)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 5	penerbit	varchar(64)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 6	tahun_terbit	year(4)			No	None			Change Drop More
<input type="checkbox"/> 7	isbn	varchar(64)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 8	stok	int(11)			No	None			Change Drop More
<input type="checkbox"/> 9	dipinjam	int(11)			No	None			Change Drop More
<input type="checkbox"/> 10	dibooking	int(11)			No	None			Change Drop More
<input type="checkbox"/> 11	image	varchar(256)	latin1_swedish_ci		Yes	book-default-cover.jpg			Change Drop More

Isi tabel buku

id	judul_buku	id_kategori	pengarang	penerbit	tahun_terbit	isbn	stok	dipinjam	dibooking	image
1	Statistika dengan Program Komputer	1	Ahmad Kholiqul Amin	Deep Publish	2014	9786022809432	6	1	1	img1557402455.jpg
2	Mudah Belajar Komputer untuk Anak	1	Bambang Agus Setiawan	Huta Media	2014	9786025118500	5	3	1	img1557402397.jpg
5	PHP Komplet	1	Jubilee	Elex Media Komputindo	2017	8346753547	5	1	1	img1555522493.jpg
10	Detektif Conan Ep 200	9	Okigawa sasuke	Cultura	2016	874387583987	5	0	0	img1557401465.jpg
14	Bahasa Indonesia	2	Umri Nur'aini dan Indriyani	Pusat Perbukuan	2015	757254724884	3	0	0	img1557402703.jpg
15	Komunikasi Lintas Budaya	5	Dr. Dedy Kurnia	Published	2015	878674646488	5	0	0	img1557403156.jpg
16	Kolaborasi Codeigniter dan Ajax dalam Perancangan ...	1	Anton Subagia	Elex Media komputindo	2017	43345356577	5	0	0	img1557403502.jpg
17	From Hobby to Money	4	Deasylawati	Elex Media Komputindo	2015	87968686787879	5	0	0	img1557403658.jpg
18	Buku Saku Pramuka	8	Rudi Himawan	Pusat Perbukuan	2016	97868687978796	6	0	0	img1557404613.jpg
19	Rahasia Keajaiban Bumi	3	Nurul Ihsan	Luxima	2014	565756565768868	5	0	0	img1557404689.jpg

6.4 Membuat Koneksi Database Codeigniter dengan MySQL

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

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

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

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



```

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

```

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

```

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

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

```

6.5 Membuat Model untuk CRUD

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

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

ModelUser.php

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

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

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

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

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

    public function getUserLimit()
    {
        $this->db->select('*');
        $this->db->from('user');
```

```

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

```

ModelBuku.php

```

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

class ModelBuku extends CI_Model
{
    //manajemen buku
    public function getBuku()
    {
        return $this->db->get('buku');
    }

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

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

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

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

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

```

```

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

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

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

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

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

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

```

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

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

Application/config/autoload.php

Cari baris sintak berikut:

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

Kemudian ubah menjadi

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

Tugas pertemuan 6

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

PERTEMUAN 7

REVIEW MATERI

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

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

Tugas pertemuan 7

- a. Mahasiswa mengerjakan dan mengimplementasikan materi yang ada di pertemuan 9 yaitu membuat halaman login dan membuat halaman beranda admin
- b. Demonstrasikan dan Kumpulkan pada pertemuan selanjutnya

PERTEMUAN 8
UJIAN TENGAH SEMESTER

PERTEMUAN 9

MEMBUAT FORM LOGIN DAN PESAN NOTIFIKASI

Deskripsi Pembelajaran

Pertemuan 9 ini berisi pembahasan tentang bagaimana membuat halaman login dan halaman beranda admin, membuat sebuah pesan notifikasi dan implementasi penerapan framework css dari bootstrap.

Tujuan Pembelajaran

Pada pertemuan ini mahasiswa diharapkan mampu membuat form Login dengan enkripsi password MD5 dan memahami pembuatan notifikasi kesalahan dalam inputan form. Pembaca juga diharapkan mampu memahami dan membuat halaman index untuk admin dengan menerapkan framework css dari bootstrap.

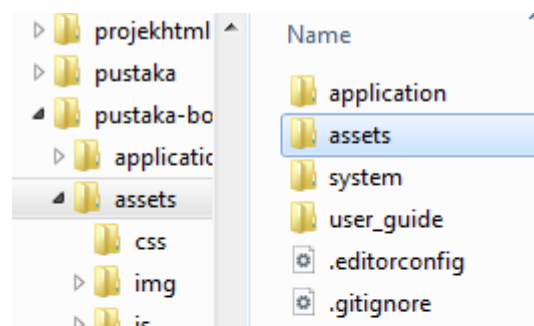
9.1 Membuat Halaman Login

Halaman login akan kita buat di halaman utama pada saat aplikasi pustaka-booking bagian admin diakses.

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

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

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



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

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

<head>

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

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

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

```

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

</head>

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

<div class="container">

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

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

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

```



```

<script src="vendor/jquery/jquery.min.js"></script>
<script
src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<!-- Core plugin JavaScript-->
<script src="vendor/jquery-easing/jquery.easing.min.js"></script>

<!-- Custom scripts for all pages-->
<script src="js/sb-admin-2.min.js"></script>

</body>

</html>

```

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

aute_header.php

```

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

<head>

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

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

    <!-- Custom fonts for this template-->
    <link href="<?= base_url('assets/'); ?>vendor/fontawesome-
free/css/all.min.css" rel="stylesheet" type="text/css">

```

```

    <link
href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400i,600,600i,700,700i,800,800i,900,900i" rel="stylesheet">

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

</head>

<body class="bg-gradient-primary">
login.php

```

```

<div class="container">

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

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

            <div class="card o-hidden border-0 shadow-lg my-5">
                <div class="card-body p-0">
                    <!-- Nested Row within Card Body -->
                    <div class="row">
                        <div class="col-lg">
                            <div class="p-5">
                                <div class="text-center">
                                    <h1 class="h4 text-gray-900 mb-4">Halaman Login!!</h1>

                                </div>
                                <?= $this->session->flashdata('pesan'); ?>

                                <form class="user" method="post"
action="{?= base_url('autentifikasi'); ?>">
                                    <div class="form-group">
                                        <input type="text"
class="form-control form-control-user" value="{?=
set_value('email'); ?>" id="email" placeholder="Masukkan Alamat
Email" name="email">

                                        <?= form_error('email',
'<small class="text-danger pl-3">', '</small>'); ?>
                                    </div>

```

```
<div class="form-group">
    <input type="password"
class="form-control form-control-user" id="password"
placeholder="Password" name="password">
        <?= form_error('password',
'<small class="text-danger pl-3">', '</small>'); ?>
    </div>
    <button type="submit" class="btn
btn-primary btn-user btn-block">
        Masuk
    </button>
</form>
<hr>
<div class="text-center">
    <a class="small" href="<?=
base_url('autentifikasi/lupaPassword'); ?>">Lupa Password?</a>
</div>
<div class="text-center">
    <a class="small" href="<?=
base_url('autentifikasi/registrasi'); ?>">Daftar Member!</a>
</div>
</div>
</div>
</div>
</div>
```

aute_footer.php

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

```

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

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

</body>

</html>

```

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

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

```

<?php

class Autentifikasi extends CI_Controller
{

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

        $this->form_validation->set_rules('email', 'Alamat Email',
        'required|trim|valid_email', [
            'required' => 'Email Harus diisi!!',
            'valid_email' => 'Email Tidak Benar!!'
        ]);
        $this->form_validation->set_rules('password', 'Password',
        'required|trim', [

```

```

        'required' => 'Password Harus diisi'
    });
    if ($this->form_validation->run() == false) {
        $data['judul'] = 'Login';
        $data['user'] = '';
        //kata 'login' merupakan nilai dari variabel judul dalam
        array $data dikirimkan ke view aute_header
        $this->load->view('templates/aute_header', $data);
        $this->load->view('autentifikasi/login');
        $this->load->view('templates/aute_footer');
    } else {
        $this->_login();
    }
}

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

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

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

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

                if ($user['role_id'] == 1) {
                    redirect('admin');
                } else {
                    if ($user['image'] == 'default.jpg') {

```



```

        $this->session->set_flashdata('pesan',
        '<div class="alert alert-info alert-message" role="alert">Silahkan
        Ubah Profile Anda untuk Ubah Photo Profil</div>');
    }
    redirect('user');
}
} else {
    $this->session->set_flashdata('pesan', '<div
    class="alert alert-danger alert-message" role="alert">Password
    salah!!</div>');
    redirect('autentifikasi');
}
} else {
    $this->session->set_flashdata('pesan', '<div
    class="alert alert-danger alert-message" role="alert">User belum
    diaktifasi!!</div>');
    redirect('autentifikasi');
}
} else {
    $this->session->set_flashdata('pesan', '<div
    class="alert alert-danger alert-message" role="alert">Email tidak
    terdaftar!!</div>');
    redirect('autentifikasi');
}
}
}
}

```

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

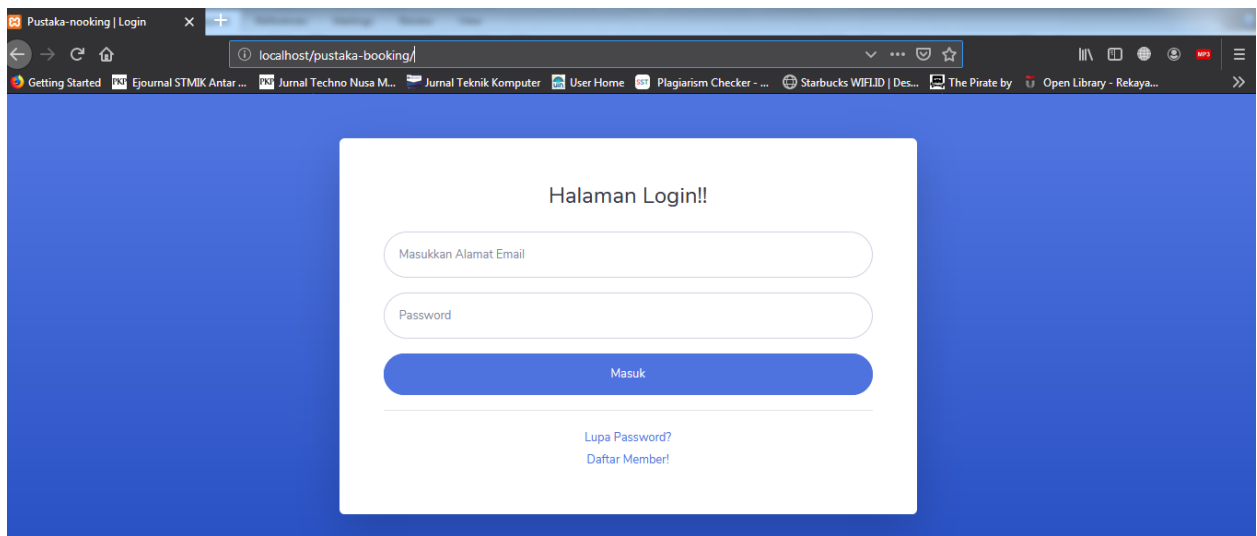
```

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

```

Sebelum melihat hasilnya, ubah pengaturan controller default terlebih dahulu yang sebelum controller welcome ubah mejadi controller autentifikasi.

Untuk melihat hasilnya silahkan ketik URL seperti ini **localhost/pustaka-booking/**, maka hasilnya seperti dibawah ini:



9.2 Membuat Halaman Admin

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

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

header.php

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

<head>

    <meta charset="utf-8">
```

```

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

<title>pustaka-booking | <?= $judul; ?></title>

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

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

</head>

<body id="page-top">

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

```

index.php

```

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

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

```

```

        <div class="card border-left-danger shadow h-100 py-2 bg-
primary">
            <div class="card-body">
                <div class="row no-gutters align-items-center">
                    <div class="col mr-2">
                        <div class="text-md font-weight-bold text-white text-
uppercase mb-1">Jumlah Anggota</div>
                        <div class="h1 mb-0 font-weight-bold text-white"><?=
$this->ModelUser->getUserWhere(['role_id' => 1])->num_rows();
?></div>
                    </div>
                    <div class="col-auto">
                        <a href="<?= base_url('user/anggota'); ?>"><i
class="fas fa-users fa-3x text-warning"></i></a>
                    </div>
                </div>
            </div>
        </div>

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

```

```

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

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

                            echo $totaldipinjam;
                        ?>
                    </div>
                </div>
                <div class="col-auto">
                    <a href="<? = base_url('user'); ?>"><i class="fas fa-
user-tag fa-3x text-success"></i></a>
                </div>
            </div>
        </div>
    </div>
</div>

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

```

```

        $totaldibooking = $this->ModelBuku-
>total('dibooking', $where);
        echo $totaldibooking;
        ?>
    </div>
</div>
<div class="col-auto">
    <a href="<?php echo base_url('user'); ?>"><i class="fas fa-
shopping-cart fa-3x text-danger"></i></a>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- end row ux-->

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

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

```

```

<tbody>
  <?php
  $i = 1;
  foreach ($anggota as $a) { ?>
    <tr>
      <td><?= $i++; ?></td>
      <td><?= $a['nama']; ?></td>
      <td><?= $a['email']; ?></td>
      <td><?= $a['role_id']; ?></td>
      <td><?= $a['is_active']; ?></td>
      <td><?= date('Y', $a['tanggal_input']); ?></td>
    </tr>
  <?php } ?>
</tbody>
</table>
</div>

<div class="table-responsive table-bordered col-sm-5 ml-auto mr-
auto mt-2">
  <div class="page-header">
    <span class="fas fa-book text-warning mt-2"> Data
    Buku</span>
    <a href="<?= base_url('buku'); ?>"><i class="fas fa-search
    text-primary mt-2 float-right"> Tampilkan</i></a>
  </div>
  <div class="table-responsive">
    <table class="table mt-3" id="table-datatable">
      <thead>
        <tr>
          <th>#</th>
          <th>Judul Buku</th>
          <th>Pengarang</th>
          <th>Penerbit</th>
          <th>Tahun Terbit</th>
          <th>ISBN</th>
          <th>Stok</th>
        </tr>
      </thead>
      <tbody>
        <?php
        $i = 1;

```

```

        foreach ($buku as $b) { ?>
            <tr>
                <td><?= $i++; ?></td>
                <td><?= $b['judul_buku']; ?></td>
                <td><?= $b['pengarang']; ?></td>
                <td><?= $b['penerbit']; ?></td>
                <td><?= $b['tahun_terbit']; ?></td>
                <td><?= $b['isbn']; ?></td>
                <td><?= $b['stok']; ?></td>
            </tr>
            <?php } ?>
        </tbody>
    </table>
</div>
</div>

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

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

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

```

footer.php

```

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

</div>

```



```

<!-- End of Content Wrapper -->

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

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

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

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

<!-- Core plugin JavaScript-->

```

```

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

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

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

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

</body>

</html>

```

sidebar.php

```

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

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

    <!-- Divider -->

```

```

<hr class="sidebar-divider">

<!-- Looping Menu-->

    <!-- Heading -->
    <div class="sidebar-heading">
        Master Data
    </div>

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

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

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

```

topbar.php

```

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

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

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

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

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

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

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

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

```

Profile Saya

```

        </a>
        <div class="dropdown-divider"></div>
        <a class="dropdown-item" href="<?=
base_url('autentifikasi/logout'); ?>" data-toggle="modal" data-
target="#logoutModal">
            <i class="fas fa-sign-out-alt fa-sm fa-
fw mr-2 text-gray-400"></i>
            Logout
        </a>
    </div>
</li>

</ul>

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

```

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

```

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

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

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

```

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

Tugas pertemuan 8

- a. Mahasiswa mengerjakan dan mengimplementasikan latihan yang ada di materi pertemuan 10 yaitu membuat helper dan membuat halaman registrasi member, membuat halaman my profile dan ubah profile. Lihat modul halaman 75, 81, 86
- b. Demonstrasikan dan Kumpulkan hasil pengerjaan pada pertemuan selanjutnya

PERTEMUAN 10

MEMBUAT FORM REGISTRASI USER

Deskripsi Pembelajaran

Berisi pembahasan tentang pembuatan helper, jalannya registrasi anggota, halaman my profile dan mengubah profile. Dipertemuan ini juga dibahas aturan penamaan dan penggunaan helper.

Dalam kasus ini, sementara form registrasi bisa diakses melalui halaman login dengan link registrasi user. Dikarenakan pada materi web programming 2 ini pembahasannya belum sampai membuat halaman front-end. Yaitu halaman member dan pengunjung. Pembahasannya akan ada pada materi web programming 3 di semester selanjutnya. Jika sudah sampai halaman front-end, barulah link untuk mengakses form registrasi akan diletakkan pada bagian frontend.

Tujuan Pembelajaran

Pada pertemuan ini mahasiswa diharapkan dapat memahami konsep registrasi dan membuat form input registrasi. Mahasiswa juga diharapkan memahami bagaimana membuat helper sendiri dan bagaimana cara penggunaannya.

10.1 Membuat helper

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

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

```
<?php
```

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

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

Dari tampilan script di atas, terlihat ada duan function yaitu **cek_login()** untuk memastikan sudah login atau belum.

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

blok.php

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

<head>

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

    <title>Akses diblock </title>

    <!-- Custom fonts for this template-->
```



```

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

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

</head>

<body id="page-top">

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

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

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

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

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

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

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

```

```

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

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

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

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

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

```

```

    <script src="<?=
base_url('assets/');?>vendor/bootstrap/js/bootstrap.bundle.min.js"><
/script>

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

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

</body>

</html>

```

gagal.php

```

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

<head>

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

    <title>Gagal Aktifasi Akun </title>

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

    <!-- Custom styles for this template-->

```

```

    <link href="{?= base_url('assets/'); ?>css/sb-admin-2.min.css"
rel="stylesheet">

</head>

<body id="page-top">

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

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

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

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

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

                    </div>

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

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

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

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

```

```

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

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

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

</body>

</html>

```

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

```

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

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

```

10.2 Membuat Form Registrasi

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

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

```

<div class="container">

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

        <div class="card-body p-0">
            <!-- Nested Row within Card Body -->
            <div class="row">
                <div class="col-lg">
                    <div class="p-5">
                        <div class="text-center">
                            <h1 class="h4 text-gray-900 mb-
4">Daftar Menjadi Member!</h1>
                        </div>
                        <form class="user" method="post"
action="<%= base_url('autentifikasi/registrasi'); ?>">
                            <div class="form-group">
                                <input type="text" class="form-
control form-control-user" id="nama" name="nama" placeholder="Nama
Lengkap" value="<%= set_value('nama'); ?>">
                                <%= form_error('nama', '<small
class="text-danger pl-3">', '</small>'); ?>
                            </div>
                            <div class="form-group">
                                <input type="text" class="form-
control form-control-user" id="email" name="email"
placeholder="Alamat Email" value="<%= set_value('email'); ?>">
                                <%= form_error('email', '<small
class="text-danger pl-3">', '</small>'); ?>
                            </div>
                            <div class="form-group row">
                                <div class="col-sm-6 mb-3 mb-sm-
0">
                                    <input type="password"
class="form-control form-control-user" id="password1"
name="password1" placeholder="Password">
                                    <%= form_error('password1',
'<small class="text-danger pl-3">', '</small>'); ?>
                                </div>
                                <div class="col-sm-6">
                                    <input type="password"
class="form-control form-control-user" id="password2"
name="password2" placeholder="Ulangi Password">

```

```

                                <?= form_error('password2',
'<small class="text-danger pl-3">', '</small>'); ?>
                                </div>
                            </div>
                            <button type="submit" class="btn btn-
primary btn-user btn-block">
                                Daftar Menjadi Member
                            </button>
                        </form>
                        <hr>
                        <div class="text-center">
                            <a class="small" href="<?=
base_url('autentifikasi/lupaPassword'); ?>">Lupa Password?</a>
                        </div>
                        <div class="text-center">
                            Sudah Menjadi Member?<a class="small"
href="<?= base_url('autentifikasi'); ?>"> Login!</a>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

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

```

public function registrasi()
{
    if ($this->session->userdata('email')) {
        redirect('user');
    }
    //membuat rule untuk inputan nama agar tidak boleh kosong
    dengan membuat pesan error dengan
    //bahasa sendiri yaitu 'Nama Belum diisi'
    $this->form_validation->set_rules('nama', 'Nama Lengkap',
    'required', [
        'required' => 'Nama Belum diis!!'
    ]);
}

```

```

        //membuat rule untuk inputan email agar tidak boleh kosong,
        tidak ada spasi, format email harus valid
        //dan email belum pernah dipakai sama user lain dengan
        membuat pesan error dengan bahasa sendiri
        //yaitu jika format email tidak benar maka pesannya 'Email
        Tidak Benar!!'. jika email belum diisi,
        //maka pesannya adalah 'Email Belum diisi', dan jika email
        yang diinput sudah dipakai user lain,
        //maka pesannya 'Email Sudah dipakai'
        $this->form_validation->set_rules('email', 'Alamat Email',
        'required|trim|valid_email|is_unique[user.email]', [
            'valid_email' => 'Email Tidak Benar!!',
            'required' => 'Email Belum diisi!!',
            'is_unique' => 'Email Sudah Terdaftar!'
        ]);
        //membuat rule untuk inputan password agar tidak boleh
        kosong, tidak ada spasi, tidak boleh kurang dari
        //dari 3 digit, dan password harus sama dengan repeat
        password dengan membuat pesan error dengan
        //bahasa sendiri yaitu jika password dan repeat password
        tidak diinput sama, maka pesannya
        //'Password Tidak Sama'. jika password diisi kurang dari 3
        digit, maka pesannya adalah
        //'Password Terlalu Pendek'.
        $this->form_validation->set_rules('password1', 'Password',
        'required|trim|min_length[3]|matches[password2]', [
            'matches' => 'Password Tidak Sama!!',
            'min_length' => 'Password Terlalu Pendek'
        ]);
        $this->form_validation->set_rules('password2', 'Repeat
        Password', 'required|trim|matches[password1]');
        //jika jika disubmit kemudian validasi form diatas tidak
        berjalan, maka akan tetap berada di
        //tampilan registrasi. tapi jika disubmit kemudian validasi
        form diatas berjalan, maka data yang
        //diinput akan disimpan ke dalam tabel user
        if ($this->form_validation->run() == false) {
            $data['judul'] = 'Registrasi Member';
            $this->load->view('templates/aut_header', $data);
            $this->load->view('autentifikasi/registrasi');
            $this->load->view('templates/aut_footer');
        } else {

```



```

        $email = $this->input->post('email', true);
        $data = [
            'nama' => htmlspecialchars($this->input-
>post('nama', true)),
            'email' => htmlspecialchars($email),
            'image' => 'default.jpg',
            'password' => password_hash($this->input-
>post('password1'), PASSWORD_DEFAULT),
            'role_id' => 2,
            'is_active' => 0,
            'tanggal_input' => time()
        ];

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

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

```

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

```

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

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

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

    public function getUserWhere($where = null)

```

```

{
    return $this->db->get_where('user', $where);
}

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

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

```

10.3 Membuat Halaman My Profile dan Ubah Profile

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

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

```

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

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

```

```

    }

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

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

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

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

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

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

```

```

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

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

    if ($upload_image) {
        $config['upload_path'] = './assets/img/profile/';
        $config['allowed_types'] = 'gif|jpg|png';
        $config['max_size'] = '3000';
        $config['max_width'] = '1024';
        $config['max_height'] = '1000';
        $config['file_name'] = 'pro' . time();

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

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

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

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

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

```

```

        redirect('user');
    }
}
}

```

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

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

Index.php

```

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

    <div class="row">
        <div class="col-lg-6 justify-content-x">
            <?= $this->session->flashdata('pesan'); ?>
        </div>
    </div>

    <div class="card mb-3" style="max-width: 540px;">
        <div class="row no-gutters">
            <div class="col-md-4">
                
            </div>
            <div class="col-md-8">
                <div class="card-body">
                    <h5 class="card-title"><?= $user['nama'];
?></h5>

                    <p class="card-text"><?= $user['email']; ?></p>
                    <p class="card-text"><small class="text-
muted">Jadi member sejak: <br><b><?= date('d F Y',
$user['tanggal_input']); ?></b></small></p>
                </div>
                <div class="btn btn-info ml-3 my-3">
                    <a href="<?= base_url('user/ubahprofil'); ?>"
class="text text-white"><i class="fas fa-user-edit"></i> Ubah
Profil</a>
                </div>
            </div>
        </div>
    </div>

```

```

        </div>
    </div>

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

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

```

ubah-profile.php

```

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

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

```

```

                
            </div>
            <div class="col-sm-9">
                <div class="custom-file">
                    <input type="file" class="custom-
file-input" id="image" name="image">
                    <label class="custom-file-label"
for="image">Pilih file</label>
                </div>
            </div>
        </div>
        <div class="form-group row justify-content-end">
            <div class="col-sm-10">
                <button type="submit" class="btn btn-
primary">Ubah</button>
                <button class="btn btn-dark"
onclick="window.history.go(-1)"> Kembali</button>
            </div>
        </div>
    </form>
</div>
</div>
<!-- /.container-fluid -->

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

```

Tugas pertemuan 10

- a. Mahasiswa mengerjakan dan mengimplementasikan latihan yang ada di materi pertemuan 11 yaitu membuat CRUD kategori buku
- b. Demonstrasikan dan Kumpulkan hasil pengerjaan pada pertemuan selanjutnya

PERTEMUAN 11

MEMBUAT KATEGORI BUKU

Deskripsi Pembelajaran

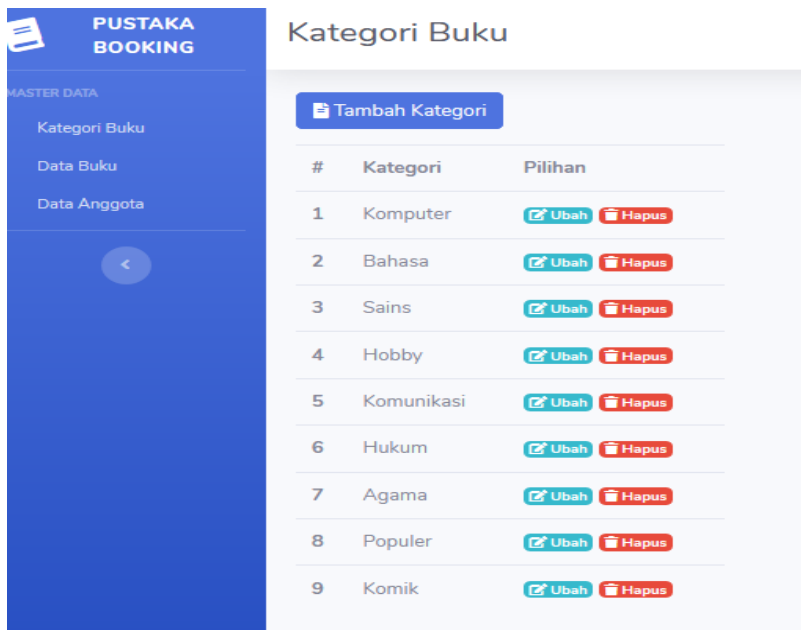
Berisi pembahasan tentang pembuatan menambah data, menampilkan data, mengubah data, dan menghapus data kategori buku.

Tujuan Pembelajaran

Pada pertemuan ini mahasiswa diharapkan dapat memahami konsep create, read, update, dan delete. Untuk memanipulasi sebuah data pada database

11.1 Membuat Tampil Data Kategori

Kategori buku akan ditampilkan seperti berikut:



#	Kategori	Pilihan
1	Komputer	Ubah Hapus
2	Bahasa	Ubah Hapus
3	Sains	Ubah Hapus
4	Hobby	Ubah Hapus
5	Komunikasi	Ubah Hapus
6	Hukum	Ubah Hapus
7	Agama	Ubah Hapus
8	Populer	Ubah Hapus
9	Komik	Ubah Hapus

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

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

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

```

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

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

```

```

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

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

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

<!-- Modal Tambah kategori baru-->
<div class="modal fade" id="kategoriBaruModal" tabindex="-1"
role="dialog" aria-labelledby="kategoriBaruModalLabel" aria-
hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title"
id="kategoriBaruModalLabel">Tambah Kategori</h5>
                <button type="button" class="close" data-
dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <form action="<?= base_url('buku/kategori'); ?>"
method="post">
                <div class="modal-body">
                    <div class="form-group">
                        <select name="kategori" class="form-control
form-control-user">
                            <option value="">Pilih Kategori</option>
                            <?php
                                $k =
['Sains', 'Hobby', 'Komputer', 'Komunikasi', 'Hukum', 'Agama', 'Populer', '
Bahasa', 'Komik'];
                                for ($i=0;$i<9;$i++) { ?>
                                    <option value="<?= $k[$i];?>"><?=
$k[$i];?></option>
                                <?php } ?>
                            </select>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        <div class="modal-footer">
            <button type="button" class="btn btn-secondary"
data-dismiss="modal"><i class="fas fa-ban"></i> Close</button>
            <button type="submit" class="btn btn-primary"><i
class="fas fa-plus-circle"></i> Tambah</button>
        </div>
    </form>
</div>
</div>
</div>
<!-- End of Modal Tambah Mneu -->

```

11.2 Hapus data kategori

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

```

public function kategori()
{
    $data['judul'] = 'Kategori Buku';
    $data['user'] = $this->ModelUser->cekData(['email' => $this->session->userdata('email')])->row_array();
    $data['kategori'] = $this->ModelBuku->getKategori()->result_array();

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

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

```

```

];

$this->ModelBuku->simpanKategori($data);
redirect('buku/kategori');
}
}

public function hapusKategori()
{
    $where = ['id' => $this->uri->segment(3)];
    $this->ModelBuku->hapusKategori($where);
    redirect('buku/kategori');
}

```

Tugas 10

- a. Mahasiswa mengerjakan dan mengimplementasikan latihan yang ada di materi pertemuan 12 yaitu membuat form tambah buku dan tampilan data buku, membuat modelbuku.php, mengubah dan update data buku, dan menghapus data buku. Semuanya ada di modul halaman 98, 101, 108, dan 110
- b. Demonstrasikan dan Kumpulkan hasil pengerjaan pada pertemuan selanjutnya

PERTEMUAN 12

MEMBUAT CRUD DATA BUKU

Deskripsi Pembelajaran

Pertemuan ini membahas tentang pembuatan form tambah data buku dan tampilan data buku yang sudah masuk dalam database, kemudian membahas tentang pembuatan modelbuku.php untuk pemetaan data dari database, dan membuat form ubah data serta hapus data buku.

Dalam kasus ini pembuatan CRUD data-data buku, disesuaikan 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.

Tujuan Pembelajaran

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

12.1 Menampilkan Data Buku dan tambah Buku

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

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

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

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

```

//manajemen Buku
public function index()
{
    $data['judul'] = 'Data Buku';
    $data['user'] = $this->ModelUser->cekData(['email' => $this->session->userdata('email')])>row_array();
    $data['buku'] = $this->ModelBuku->tampil()->result_array();
    $data['kategori'] = $this->ModelBuku->getKategori()->result_array();

    $this->form_validation->set_rules('judul_buku', 'Judul Buku', 'required|min_length[3]', [
        'required' => 'Judul Buku harus diisi',
        'min_length' => 'Judul buku terlalu pendek'
    ]);
    $this->form_validation->set_rules('id_kategori', 'Kategori', 'required', [
        'required' => 'Nama pengarang harus diisi',
    ]);
    $this->form_validation->set_rules('pengarang', 'Nama Pengarang', 'required|min_length[3]', [
        'required' => 'Nama pengarang harus diisi',
        'min_length' => 'Nama pengarang terlalu pendek'
    ]);
    $this->form_validation->set_rules('penerbit', 'Nama Penerbit', 'required|min_length[3]', [
        'required' => 'Nama penerbit harus diisi',
        'min_length' => 'Nama penerbit terlalu pendek'
    ]);
    $this->form_validation->set_rules('tahun', 'Tahun Terbit', 'required|min_length[3]|max_length[4]|numeric', [
        'required' => 'Tahun terbit harus diisi',
        'min_length' => 'Tahun terbit terlalu pendek',
        'max_length' => 'Tahun terbit terlalu panjang',
        'numeric' => 'Hanya boleh diisi angka'
    ]);
    $this->form_validation->set_rules('isbn', 'Nomor ISBN', 'required|min_length[3]|numeric', [
        'required' => 'Nama ISBN harus diisi',
        'min_length' => 'Nama ISBN terlalu pendek',
        'numeric' => 'Yang anda masukan bukan angka'
    ]);
}

```

```

    ]);
    $this->form_validation->set_rules('stok', 'Stok',
'required|numeric', [
        'required' => 'Stok harus diisi',
        'numeric' => 'Yang anda masukan bukan angka'
    ]);

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

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

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

        $data = [
            'judul_buku' => $this->input->post('judul_buku',
true),
            'id_kategori' => $this->input->post('id_kategori',
true),
            'pengarang' => $this->input->post('pengarang',
true),
            'penerbit' => $this->input->post('penerbit', true),
            'tahun_terbit' => $this->input->post('tahun', true),
            'isbn' => $this->input->post('isbn', true),
            'stok' => $this->input->post('stok', true),

```



```

        'dipinjam' => 0,
        'dibooking' => 0,
        'image' => $gambar
    ];

    $this->ModelBuku->simpanBuku($data);
    redirect('buku');
}
}
}

```

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

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

```

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

class ModelBuku extends CI_Model
{
    //manajemen buku
    public function getBuku()
    {
        return $this->db->get('buku');
    }

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

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

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

    public function hapusBuku($where = null)

```

```

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

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

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

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

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

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

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

//join
public function joinKategoriBuku($where)

```

```

{
    $this->db->select('buku.id_kategori,kategori.kategori');
    $this->db->from('buku');
    $this->db->join('kategori','kategori.id =
buku.id_kategori');
    $this->db->where($where);
    return $this->db->get();
}
}

```

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

```

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

    <?= $this->session->flashdata('pesan'); ?>
    <div class="row">
        <div class="col-lg-12">
            <?php if(validation_errors()){?>
                <div class="alert alert-danger" role="alert">
                    <?= validation_errors();?>
                </div>
            <?php }?>
            <?= $this->session->flashdata('pesan'); ?>
            <a href="" class="btn btn-primary mb-3" data-
toggle="modal" data-target="#bukuBaruModal"><i class="fas fa-file-
alt"></i> Buku Baru</a>
            <table class="table table-hover">
                <thead>
                    <tr>
                        <th scope="col">#</th>
                        <th scope="col">Judul</th>
                        <th scope="col">Pengarang</th>
                        <th scope="col">Penerbit</th>
                        <th scope="col">Tahun Terbit</th>
                        <th scope="col">ISBN</th>
                        <th scope="col">Stok</th>
                        <th scope="col">DiPinjam</th>
                        <th scope="col">DiBooking</th>
                        <th scope="col">Gambar</th>

```

```

        <th scope="col">Pilihan</th>
    </tr>
</thead>
<tbody>

    <?php
        $a = 1;
        foreach ($buku as $b) { ?>
    <tr>
        <th scope="row"><?= $a++; ?></th>
        <td><?= $b['judul_buku']; ?></td>
        <td><?= $b['pengarang']; ?></td>
        <td><?= $b['penerbit']; ?></td>
        <td><?= $b['tahun_terbit']; ?></td>
        <td><?= $b['isbn']; ?></td>
        <td><?= $b['stok']; ?></td>
        <td><?= $b['dipinjam']; ?></td>
        <td><?= $b['dibooking']; ?></td>
        <td>
            <picture>
                <source srcset=""
type="image/svg+xml">
                
            </picture></td>
        <td>
            <a href="<?=
base_url('buku/ubahBuku/').$b['id'];?>" class="badge badge-info"><i
class="fas fa-edit"></i> Ubah</a>
            <a href="<?=
base_url('buku/hapusbuku/').$b['id'];?>" onclick="return
confirm('Kamu yakin akan menghapus <?= $judul.'
'.$b['judul_buku'];?> ?');" class="badge badge-danger"><i class="fas
fa-trash"></i> Hapus</a>
        </td>
    </tr>
    <?php } ?>
</tbody>
</table>
</div>
</div>

```

```

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

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

<!-- Modal Tambah buku baru-->
<div class="modal fade" id="bukuBaruModal" tabindex="-1"
role="dialog" aria-labelledby="bukuBaruModallabel" aria-
hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title"
id="bukuBaruModallabel">Tambah Buku</h5>
                <button type="button" class="close" data-
dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <form action="<?= base_url('buku'); ?>" method="post"
enctype="multipart/form-data">
                <div class="modal-body">
                    <div class="form-group">
                        <input type="text" class="form-control form-
control-user" id="judul_buku" name="judul_buku"
placeholder="Masukkan Judul Buku">
                    </div>
                    <div class="form-group">
                        <select name="id_kategori" class="form-
control form-control-user">
                            <option value="">Pilih Kategori</option>
                            <?php
                                foreach ($kategori as $k) { ?>
                                    <option value="<?= $k['id']; ?>"><?=
$k['kategori']; ?></option>
                                <?php } ?>
                            </select>
                        </div>
                        <div class="form-group">

```

```

        <input type="text" class="form-control form-
control-user" id="pengarang" name="pengarang" placeholder="Masukkan
nama pengarang">
    </div>
    <div class="form-group">
        <input type="text" class="form-control form-
control-user" id="penerbit" name="penerbit" placeholder="Masukkan
nama penerbit">
    </div>
    <div class="form-group">
        <select name="tahun" class="form-control
form-control-user">
            <option value="">Pilih Tahun</option>
            <?php
            for ($i=date('Y'); $i > 1000 ; $i--) {
?>
                <option value="<?= $i;?>"><?=
$i;?></option>
            <?php } ?>
        </select>
    </div>
    <div class="form-group">
        <input type="text" class="form-control form-
control-user" id="isbn" name="isbn" placeholder="Masukkan ISBN">
    </div>
    <div class="form-group">
        <input type="text" class="form-control form-
control-user" id="stok" name="stok" placeholder="Masukkan nominal
stok">
    </div>
    <div class="form-group">
        <input type="file" class="form-control form-
control-user" id="image" name="image">
    </div>
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-secondary"
data-dismiss="modal"><i class="fas fa-ban"></i> Close</button>
    <button type="submit" class="btn btn-primary"><i
class="fas fa-plus-circle"></i> Tambah</button>
</div>
</form>

```

```

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



```

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

Data Buku

Imam Khotib

Buku Baru

#	Judul	Pengarang	Penerbit	Tahun Terbit	ISBN	Stok	DiPinjam	DiBooking	Gambar	Pilihan
1	Statistika dengan Program Komputer	Ahmad Kholiqul Amin	Deep Publish	2014	9786022809432	6	1	1		Ubah Hapus
2	Mudah Belajar Komputer untuk Anak	Bambang Agus Setiawan	Huta Media	2014	9786025118500	5	3	1		Ubah Hapus

12.2 Update Data Buku

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

```

public function ubahBuku()
{
    $data['judul'] = 'Ubah Data Buku';
    $data['user'] = $this->ModelUser->cekData(['email' => $this->session->userdata('email')])->row_array();
    $data['buku'] = $this->ModelBuku->bukuWhere(['id' => $this->uri->segment(3)])->result_array();
    $kategori = $this->ModelBuku->joinKategoriBuku(['buku.id' => $this->uri->segment(3)])->result_array();
    foreach ($kategori as $k) {

```

```

        $data['id'] = $k['id_kategori'];
        $data['k'] = $k['kategori'];
    }
    $data['kategori'] = $this->ModelBuku->getKategori()-
>result_array();

    $this->form_validation->set_rules('judul_buku', 'Judul
Buku', 'required|min_length[3]', [
        'required' => 'Judul Buku harus diisi',
        'min_length' => 'Judul buku terlalu pendek'
    ]);
    $this->form_validation->set_rules('id_kategori', 'Kategori',
'required', [
        'required' => 'Nama pengarang harus diisi',
    ]);
    $this->form_validation->set_rules('pengarang', 'Nama
Pengarang', 'required|min_length[3]', [
        'required' => 'Nama pengarang harus diisi',
        'min_length' => 'Nama pengarang terlalu pendek'
    ]);
    $this->form_validation->set_rules('penerbit', 'Nama
Penerbit', 'required|min_length[3]', [
        'required' => 'Nama penerbit harus diisi',
        'min_length' => 'Nama penerbit terlalu pendek'
    ]);
    $this->form_validation->set_rules('tahun', 'Tahun Terbit',
'required|min_length[3]|max_length[4]|numeric', [
        'required' => 'Tahun terbit harus diisi',
        'min_length' => 'Tahun terbit terlalu pendek',
        'max_length' => 'Tahun terbit terlalu panjang',
        'numeric' => 'Hanya boleh diisi angka'
    ]);
    $this->form_validation->set_rules('isbn', 'Nomor ISBN',
'required|min_length[3]|numeric', [
        'required' => 'Nama ISBN harus diisi',
        'min_length' => 'Nama ISBN terlalu pendek',
        'numeric' => 'Yang anda masukan bukan angka'
    ]);
    $this->form_validation->set_rules('stok', 'Stok',
'required|numeric', [
        'required' => 'Stok harus diisi',
        'numeric' => 'Yang anda masukan bukan angka'
    ]);

```



```

]);

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

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

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

    $data = [
        'judul_buku' => $this->input->post('judul_buku',
true),
        'id_kategori' => $this->input->post('id_kategori',
true),
        'pengarang' => $this->input->post('pengarang',
true),
        'penerbit' => $this->input->post('penerbit', true),
        'tahun_terbit' => $this->input->post('tahun', true),
        'isbn' => $this->input->post('isbn', true),
        'stok' => $this->input->post('stok', true),
        'image' => $gambar
    ];

```

```

        $this->ModelBuku->updateBuku($data, ['id' => $this->input->post('id')]);
        redirect('buku');
    }
}

```

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

12.3 Menghapus Data Buku.

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

```

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

```

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

```

public function hapusBuku()
{
    $where = ['id' => $this->uri->segment(3)];
    $this->ModelBuku->hapusBuku($where);
    redirect('buku');
}

```

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

PERTEMUAN 13-15
PRESENTASI PROJEK

PERTEMUAN 16
UJIAN AKHIR SEMESTER

DAFTAR PUSTAKA

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

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

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