

Modul Pembelajaran

Web Programming 1

Penulis : Hendri Mahmud Nawawi

CodeIgniter adalah salah satu framework PHP yang ringan dan kuat, yang dirancang untuk pengembangan aplikasi web yang cepat dan mudah. Dengan menggunakan arsitektur MVC, CodeIgniter membantu memisahkan logika aplikasi dari tampilan, sehingga memudahkan pengelolaan dan pengembangan kode.

A. Installasi Codigniter 3

Kunjungi Situs Resmi pada link CodeIgniter

<https://codeigniter.com/userguide3/installation/downloads.html> seperti gambar berikut:

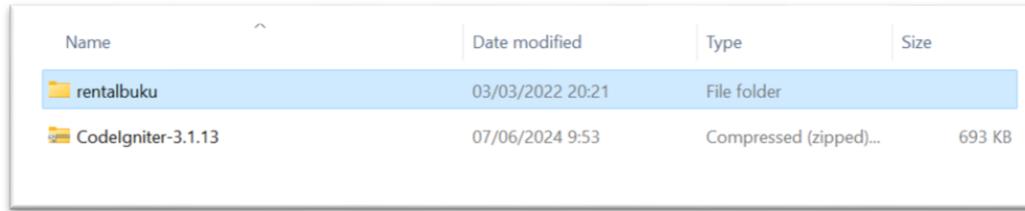
The screenshot shows a website with a dark header and sidebar. The sidebar on the left lists categories like 'Installation Instructions', 'CodeIgniter Overview', and 'Tutorial'. The main content area has a breadcrumb navigation: 'Docs > Installation Instructions > Downloading CodeIgniter'. Below this, a section titled 'Downloading CodeIgniter' lists several versions of the framework, with 'CodeIgniter v3.1.13 (Current version)' highlighted in red. A small note at the bottom right indicates 'Last updated: 2024-06-07'.

- [CodeIgniter v3.1.13 \(Current version\)](#)
- [CodeIgniter v3.1.12](#)
- [CodeIgniter v3.1.11](#)
- [CodeIgniter v3.1.10](#)
- [CodeIgniter v3.1.9](#)
- [CodeIgniter v3.1.8](#)
- [CodeIgniter v3.1.7](#)
- [CodeIgniter v3.1.6](#)
- [CodeIgniter v3.1.5](#)
- [CodeIgniter v3.1.4](#)
- [CodeIgniter v3.1.3](#)
- [CodeIgniter v3.1.2](#)
- [CodeIgniter v3.1.1](#)
- [CodeIgniter v3.1.0](#)

Download versi terbaru dari situ yang dikunjungi, selanjutnya download file [CodeIgniter v3.1.13 \(Current version\)](#). Selanjutnya setelah di download extrak file tersebut dan rename menjadi project yang kita inginkan.

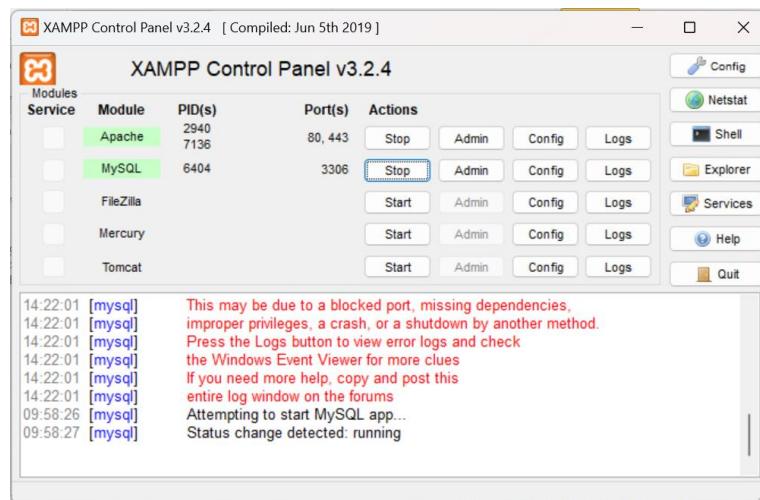
Name	Date modified	Type	Size
CodeIgniter-3.1.13	03/03/2022 20:21	File folder	
CodeIgniter-3.1.13	07/06/2024 9:53	Compressed (zipped)...	693 KB

Ubah Folder CodeIgniter-3.1.13 menjadi **rentalbuku**

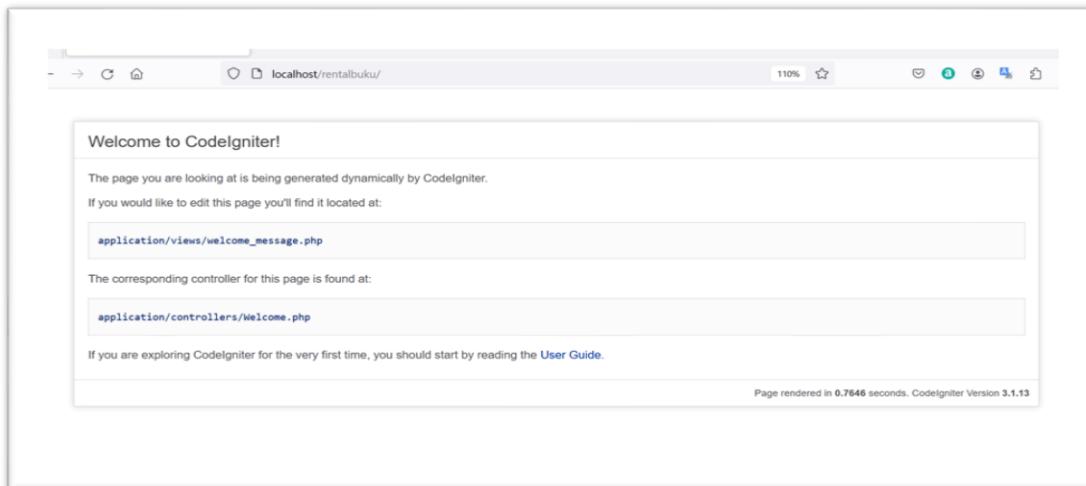


Copy file rentalbuku ke folder **C:xampp:htdocs**

Aktifkan xampp pada komputer/laptop untuk mengecek bahwa project rentalbuku yang dibuat sudah tersedia pada server lokal.

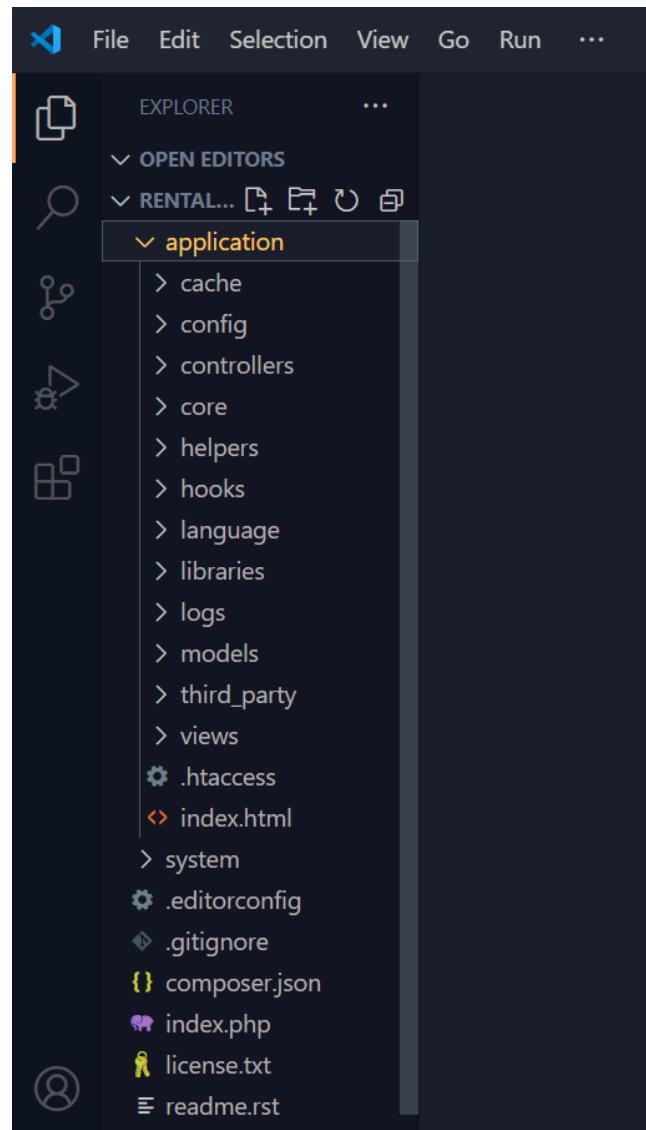


Setelah mengaktifkan **Apache** dan **MySQL** jalankan project kita pada browser pada url <http://localhost/rentalbuku/>. Jika pada tampilan browser sudah tampil seperti gambar dibawah ini maka project CodeIgniter 3 sudah ada di perangkat kita.



B. Konfigurasi CodeIgniter 3

Untuk menyesuaikan konfigurasi buka folder **rentalmobil** yang sudah dicopy dengan web editor seperti notepad++, sublime text, VSCode. Pada modul ini kita akan menggunakan VSCode. Seperti berikut adalah tampilan struktur codeigniter 3



- Konfigurasi awal adalah pada folder **config** dan memilih config.php, kemudian cari kalimat

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

Kemudian tambahkan project kita pada **base_url**

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

Simpan perubahan dengan menekan tombol **ctrl + s**

C. Membuat Halaman Web dengan Codeigniter

Dalam sesi ini, kita akan mempelajari langkah-langkah untuk membuat halaman web dasar dengan menggunakan CodeIgniter. Kita akan mulai dengan memahami konsep MVC, kemudian melanjutkan dengan pembuatan model, view, dan controller untuk membangun halaman web sederhana yang dapat menampilkan data dari database.

Langkah awal kita akan membuat **view** untuk menampilkan pesan **Hello World..!**

- Buatkan file baru dengan nama coba.php pada folder view

```
<?php
echo "Hello world";
echo "<br> Ini Adalah Project Pertama saya di CodeIgniter";
```

- Karena CodeIgniter menggunakan konsep MVC, maka langkah selanjutnya adalah membuat controller untuk memerintahkan tampilan view supaya bisa di Load di browser, maka langkah awal adalah membuat controller dengan nama Buku.php

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

class Buku extends CI_Controller
{
    public function index()
    {
        $this->load->view('coba');
    }
}
```

```
public function index()
{
    $this->load->view('coba');
}
```

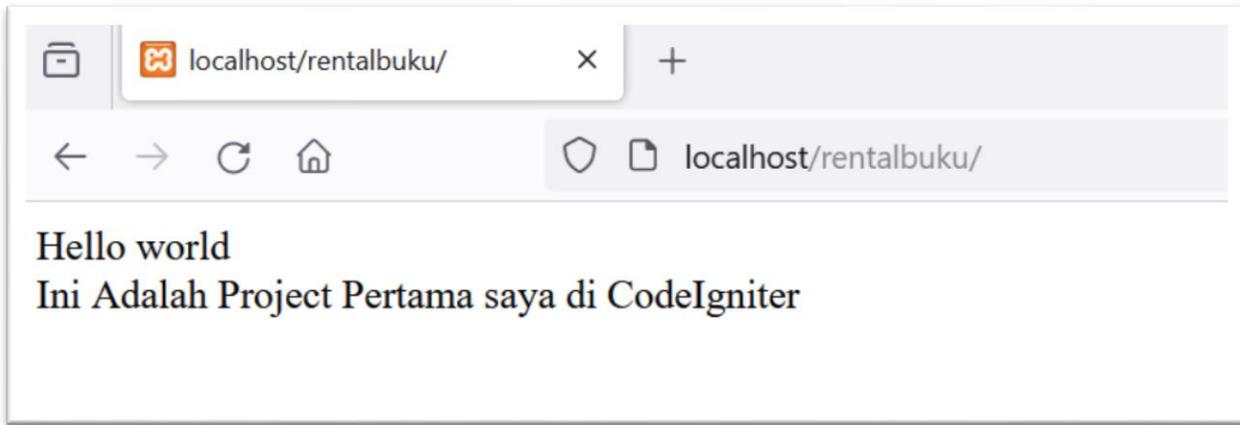
Fungsi ini merupakan perintah untuk meload file coba.php pada view. Selanjutnya mengubah settingan pada folder **config** pilih file **routes.php** dan ubah default_controller menjadi web Sebelum:

```
$route['default_controller'] = 'welcome';
$route['404_override'] = '';
$route['translate_uri_dashes'] = FALSE;
```

Sesudah:

```
$route['default_controller'] = 'buku';
$route['404_override'] = '';
$route['translate_uri_dashes'] = FALSE;
```

Maka Hasilnya setelah diload kembali pada browser adalah



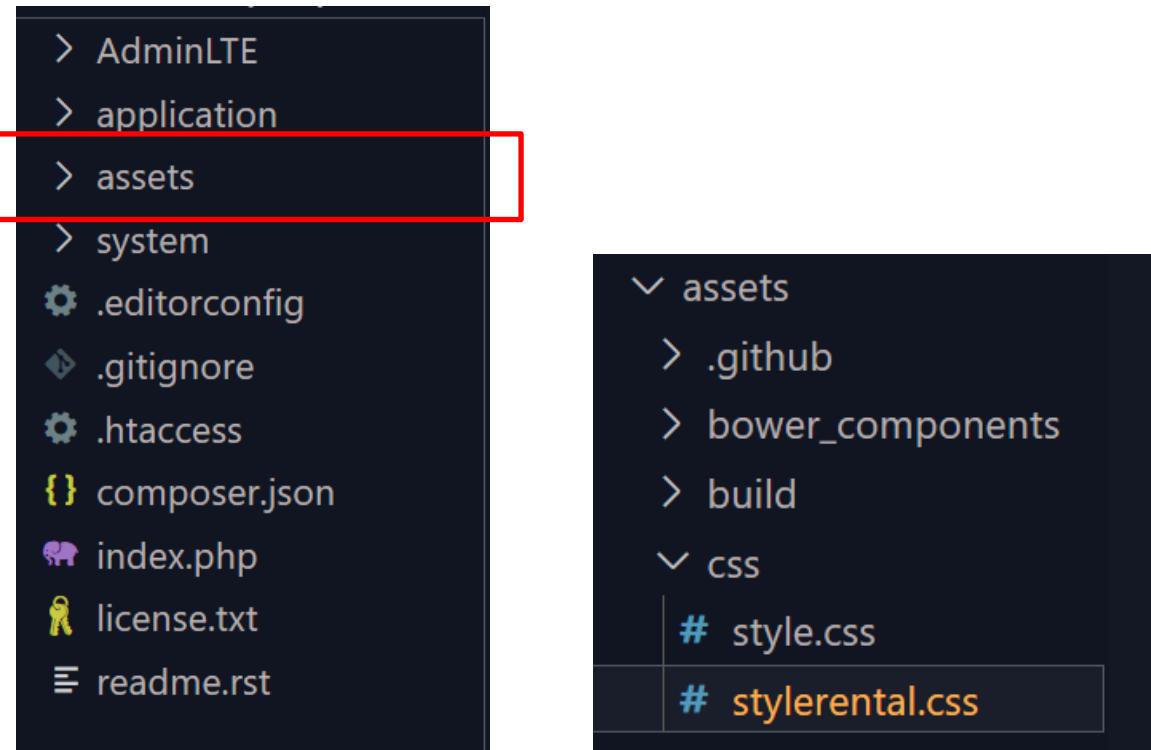
D. Membuat template sederhana

Dalam pengembangan aplikasi web, penggunaan template memegang peranan penting dalam memastikan konsistensi dan kemudahan pengelolaan tampilan situs. Pada bagian ini, kita akan mempelajari cara membuat template sederhana menggunakan framework CodeIgniter 3. Template ini akan mencakup header, footer, dan bagian konten utama yang dapat digunakan kembali di berbagai halaman. Dengan menggunakan template, kita dapat mempercepat proses

pengembangan dan memastikan tampilan yang seragam di seluruh aplikasi web. Mari kita mulai dengan langkah-langkah dasar untuk membuat dan mengimplementasikan template ini.

1. Membuat folder assets untuk menyimpan file .css

Buatlah folder baru pada project rental buku dengan nama **assets**, kemudian buat folder kembali dengan nama **css**, folder CSS ini akan menampung isi dari style yang digunakan dan akan di load ke project kita.



2. Buat file baru dengan nama style.css kemudian tulis kan settingan coding berikut untuk membuat tampilan 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;
}

```

3. Membuat Halaman View

Untuk membuat halaman web agar css yang dibuat bisa dilihat hasilnya maka terlebih dahulu membuat sebuah halaman view dengan nama index.php sebagai contoh, file **v_index.php**

```

<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Web Prog II | Merancang Template sederhana dengan
Codeigniter</title>
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/css/stylerental.css'); ?>">
</head>

<body>
    <div id="wrapper">
        <header>
            <hgroup>
                <h1>Rental Mobil</h1>

```

```

        <h3>Membuat Template Sederhana dengan CodeIgniter</h3>
    </hgroup>
    <nav>
        <ul>
            <li><a href="php echo base_url() . 'buku' ?&gt;"Home</a></li>
            <li><a href="php echo base_url() . 'buku/about' ?&gt;"About</a></li>
        </ul>
    </nav>
    <div class="clear"></div>
</header>

<section>
    <h1><?php echo $judul ?></h1>
    <p align='justify'>Pada pengertian codeigniter di atas  

        tadi di jelaskan bahwa codeigniter menggunakan metode MVC. Apa itu  

        MVC? Kita juga harus mengetahui apa itu MVC sebelum masuk dan  

        lebih  

        jauh dalam belajar codeigniter.</p>
    <p>MVC adalah teknik atau konsep yang memisahkan  

        komponen utama menjadi tiga komponen yaitu model, view dan  

        controller.</p>
    <ol type="a">
        <li>Model</li>
        <p align='justify'>Model adalah kelas yang merepresentasikan atau  

            memodelkan tipe data yang akan digunakan oleh aplikasi. Model  

            juga dapat didefinisikan sebagai bagian penanganan yang berhubungan dengan  

            pengolahan atau manipulasi database. Seperti misalnya  

            mengambil data  

            dari database, menginput dan pengolahan database lainnya. Semua  

            intruksi atau fungsi yang berhubung dengan pengolahan database  

            di letakkan di dalam model. Sebagai contoh, jika ingin membuat aplikasi  

            untuk menghitung luas dan keliling lingkaran, maka dapat  

            memodelkan objek lingkaran sebagai kelas model.</p>
        <p align='justify'>Sebagai catatan, Semua model harus disimpan di  

            dalam folder application\models</p>
        <li>View</li>
        <p align='justify'>View merupakan bagian yang menangani halaman  

            user  

            interface atau halaman yang muncul pada user(pada browser).  

            Tampilan dari user interface di kumpulkan pada view untuk memisahkannya  

            dengan controller dan model sehingga memudahkan web designer dalam  

            melakukan pengembangan tampilan halaman website.</p>
        <li>Controller</li>
    </ol>
</section>

```

```

        <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 tampilan aplikasi, karena back-end developer yang
menangani bagian controller dan modelnya. Jadi pembagian tugas
pun menjadi mudah dan pengembangan aplikasi dapat di lakukan dengan
cepat dan terstruktur.</p>
</section>
<footer>
    <a href="http://localhost/rental_mobil/web">Rental Mobil</a>
</footer>
</div>
</body>

</html>

```

4. Memanggil css kedalam view

Agar CSS yang dibuat bisa di load di view maka tambahan perintah

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

pada taggar <head> </head>

```

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Web Prog II | Merancang Template sederhana dengan
Codeigniter</title>
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/css/style.css'); ?>">

```

```
</head>
```

5. Melakukan pemanggilan di Controller

Buat Controller baru dengan nama Buku.php, kemudian panggil view yang sudah dibuat pada function index

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

class Buku extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->helper('url');
    }

    public function index()
    {
        $this->load->view('v_index');
    }
}
```

Ketika di load pada browser dengan controllers web dengan alamat <https://localhost/rentabukul/> maka hasilnya akan muncul seperti pada gambar berikut

The screenshot shows a web browser window with the URL `localhost/rental_buku`. The page has a dark header bar with the title "Rental Buku" and the subtitle "Membuat Template Sederhana dengan CodeIgniter". On the right side of the header are links for "Home", "About", and "Buku". The main content area has a heading "Halaman Depan". Below it, there is a paragraph about MVC, followed by three numbered sections (1. Model, 2. View, 3. Controller) each with a brief explanation. At the bottom of the content area is a footer bar with the text "Rental Buku".

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

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

1. 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
2. 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.
3. Controller
Controller merupakan kumpulan intruksi aksi yang menghubungkan model dan view, jadi user tidak akan berhubungan dengan model secara langsung, intinya data yang tersimpan di database (model) di ambil oleh controller dan kemudian controller pula yang menampilkan nya ke view. Jadi controller lah yang mengolah intruksi.
Dari penjelasan tentang model view dan controller di atas dapat di simpulkan bahwa controller sebagai penghubung view dan model. Misalnya pada aplikasi yang menampilkan data dengan menggunakan metode konsep mvc, controller memanggil intruksi pada model yang mengambil data pada database, kemudian controller yang meneruskannya pada view untuk di tampilkan. Jadi jelas sudah dan sangat mudah dalam pengembangan aplikasi dengan cara mvc ini karena web designer atau front-end developer tidak perlu lagi berhubungan dengan controller, dia hanya perlu berhubungan dengan view untuk mendesain tampilan 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.

E. Membuat Database dan Tabel

Untuk membuat database dan tabel pada CodeIgniter, langkah-langkah yang perlu dilakukan adalah sebagai berikut:

1. Membuat Database: Buka phpMyAdmin atau alat administrasi database lainnya dan buat database baru dengan nama **rentalbuku**.
2. Membuat Tabel: Buat tabel yang diperlukan dalam database rentalbuku. Berikut adalah contoh SQL untuk membuat tabel buku:

```
CREATE TABLE buku (
    id INT(11) AUTO_INCREMENT PRIMARY KEY,
    judul VARCHAR(255) NOT NULL,
    penulis VARCHAR(255) NOT NULL,
    tgl_publish DATE,
    kategori VARCHAR(100)
```

```
 );
```

3. Melakukan konfigurasi database

Buka file **database.php** yang ada di dalam folder **application/config** dan sesuaikan konfigurasi database dengan database yang telah dibuat. Setting menjadi sebagai berikut:

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

F. Membuat Create, Read, Update dan Delete (CRUD)

CRUD adalah operasi dasar yang dapat dilakukan terhadap data dalam sebuah aplikasi. Berikut adalah langkah-langkah untuk membuat operasi CRUD sederhana di CodeIgniter.

1. Membuat Model Buku, Buat model buku terlebih dahulu dengan nama **M_buku**

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

class M_buku extends CI_Model
{
    public function __construct()
    {
        parent::__construct();
```

```

    $this->load->database();
}

// Menampilkan Buku
public function tampil_data_buku()
{
    $query = $this->db->get('buku');
    return $query->result();
}

// Simpan Buku
public function simpan_buku($data)
{
    return $this->db->insert('buku', $data);
}

// Hapus Buku
public function hapus_buku($id)
{
    $this->db->where('id', $id);
    return $this->db->delete('buku');
}

// Edit Buku
public function update_buku($id)
{
    $data = array(
        'judul' => $this->input->post('judul'),
        'penulis' => $this->input->post('penulis'),
        'tgl_publish' => $this->input->post('tgl_publish'),
        'kategori' => $this->input->post('kategori')
    );

    $this->db->where('id', $id);
    return $this->db->update('buku', $data);
}
}

```

2. Menambahkan Fungsi pada Controllers Buku

Untuk menambahkan data buku, maka langkah pertama adalah membuat view form inputan untuk mengisi data-data pada tabel buku. Buatkan Link Baru pada taggar <head> dengan perintah berikut:

```
<li><a href="php echo base_url() . 'buku/tambah' ?&gt;"Buku</a></li>
```

3. Konfigurasi pada library config.php supaya form, url dan database dapat diload secara otomatis, maka dari itu buka kembali file autoload.php

Sebelum:

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

Sesudah:

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

4. Setting Pada Controller

Setelah ditambahkan pada autoload.php untuk ditambahkan, maka pada controllers panggil data libraris supaya berjalan dengan menambahkan perintah load libraries pada function construct

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

Hasilnya:

```
public function __construct()
{
    parent::__construct();
    $this->load->model('M_buku');
    $this->load->helper('url', 'form');
    $this->load->library('form_validation');
}
```

5. Buat View untuk input data buku

Supaya penulisannya rapih, maka view yang menampung data-data pada buku dibuatkan dalam satu folder di view yaitu **buku**, Pada folder buku buat file dengan nama input_buku.php

```
<section>
<table>
    <h3> Input Data Buku </h3>
    <form method="POST" action="= base_url('buku/tambah'); ?&gt;"&gt;
        &lt;tr&gt;
            &lt;td&gt;&lt;label for="title"&gt;Judul Buku&lt;/label&gt;&lt;/td&gt;
            &lt;td&gt;&lt;/td&gt;
            &lt;td&gt;&lt;input type="text" id="title" name="judul" required
size="35"&gt;&lt;/td&gt;</pre
```

```

        </tr>
        <tr>
            <td><label for="author">Penulis</label></td>
            <td>:</td>
            <td><input type="text" id="author" name="penulis" required size="35"></td>
        </tr>
        <tr>
            <td><label for="category">Kategori</label></td>
            <td>:</td>
            <td><select id="category" name="category" required>
                <option value="">Pilih Kategori</option>
                <option value="fiksi">Fiksi</option>
                <option value="non-fiksi">Non-Fiksi</option>
                <option value="sains">Sains</option>
                <option value="teknologi">Teknologi</option>
                <!-- Tambahkan opsi kategori lainnya sesuai kebutuhan -->
            </select></td>
        </tr>
        <tr>
            <td colspan="3">&ampnbsp</td>
        </tr>
        <tr>
            <td colspan="3"> <button type="submit" name="simpan">Simpan</button></td>
        </tr>
    </form>
</table>
</section>

```

6. Tambahkan dan Konfigurasi Controllers agar dapat memanggil view input_buku.php dengan menambahkan function **tambah()**

```

public function tambah()
{
    // Cek apakah ada perintah simpan
    if ($this->input->post('simpan')) {
        $data['title'] = 'Simpan Data Buku';

        $this->form_validation->set_rules('judul', 'Judul', 'required');
        $this->form_validation->set_rules('penulis', 'Penulis', 'required');
    }
}

```

```

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

        if ($this->form_validation->run() === FALSE) {
            $this->load->view('v_header', $data);
            $this->load->view('buku/tambah');
            $this->load->view('v_footer');
        } else {
            $data = array(
                'judul' => $this->input->post('judul'),
                'penulis' => $this->input->post('penulis'),
                'kategori' => $this->input->post('kategori'),
                'tgl_publish' => date('Y-m-d')
            );

            $insert = $this->M_buku->simpan_buku($data);

            if ($insert) {
                $this->session->set_flashdata('success', 'Data Buku
berhasil disimpan.');
            } else {
                $this->session->set_flashdata('error', 'Data Buku
gagal disimpan.');
            }
            redirect('buku/tambah');
        }
    } else {
        $data = 'Tambah Buku';
        $this->load->view('v_header', $data);
        $this->load->view('buku/input_buku', $data);
        $this->load->view('v_footer');
    }
}

```

Sehingga ketika di load di browser pada tab Buku menampilkan halaman berikut

Rental Buku

Membuat Template Sederhana dengan Codeigniter

Home About Buku

Input Data Buku

Judul Buku :

Penulis :

Kategori :

Rental Buku

Lakukan pengisian data untuk menguji apakah data yang sudah diinput tersimpan ke database atau tidak. Untuk melihat apakah sudah tersimpan atau belum kita dapat melihatnya pada halaman phpmyadmin, namun karena ini dalam bentuk project maka kita harus menampilkannya pada halaman web.

7. Menampilkan data tabel

Pada langkah ini buatkan view terlebih dahulu untuk menampilkan tabel dengan nama tampil_buku.php

```
<section class="container mt-5">
    <h2 class="mb-4"><?= $judul; ?></h2>
    <table border="1" class="table table-bordered table-striped">
        <thead class="thead-dark">
            <tr>
                <th>ID</th>
                <th>Judul Buku</th>
                <th>Penulis</th>
                <th>Kategori</th>
                <th>Aksi</th>
            </tr>
        </thead>
        <tbody>
            <?php
                $no = 1;
                foreach ($buku as $book) : ?>
```

```

<tr>
    <td><?= $no++; ?></td>
    <td><?= $book->judul; ?></td>
    <td><?= $book->penulis; ?></td>
    <td><?= $book->kategori; ?></td>
    <td>
        <a href=<?= base_url('buku/view/' . $book->id); ?>" class="btn btn-info btn-sm">Lihat</a>
        <a href=<?= base_url('buku/edit/' . $book->id); ?>" class="btn btn-warning btn-sm">Edit</a>
        <a href=<?= base_url('buku/delete/' . $book->id); ?>" class="btn btn-danger btn-sm" onclick="return confirm('Apakah Anda yakin ingin menghapus buku ini?');">Hapus</a>
    </td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
<br>
<a href=<?= base_url('buku/tambah'); ?>">Tambah Buku</a>
</section>

```

Hasil Load pada browser

ID	Judul Buku	Penulis	Kategori	Aksi
1	Web Programming 1	Hendri	teknologi	Lihat Edit Hapus
2	Pelatihan Web	Hafidz Nurdin	sains	Lihat Edit Hapus
3	Ada Apa dengan Cinta	Melly Lee And Kawan-kawan	non-fiksi	Lihat Edit Hapus
4	Naruto	Komik 23	fiksi	Lihat Edit Hapus

8. Membuat Halaman Edit

Untuk membuat halaman edit maka terlebih dahulu mengarahkan fungsi button/link untuk memanggil halaman edit pada kolom aksi dari data tabel, pada tahap ke 7 bagian aksi edit mengarah pada halaman link sebagai berikut:

```
<a href="<?= base_url('buku/edit/' . $book->id); ?>" class="btn btn-warning btn-sm">Edit</a>
```

Maka dari itu langkah yang harus dilakukan adalah membuat view edit

- Pada folder view tambahkan file baru dengan nama **edit_buku.php**

```
<section class="container mt-5">
    <h2 class="mb-4"><?= $judul; ?></h2>
    <form method="POST" action="<?= base_url('buku/edit/' . $buku->id); ?>" enctype="multipart/form-data">
        <div class="form-group">
            <label for="judul">Judul Buku:</label>
            <input type="text" class="form-control" id="judul" name="judul" value="<?= $buku->judul; ?>" required>
        </div>
        <div class="form-group">
            <label for="penulis">Penulis:</label>
            <input type="text" class="form-control" id="penulis" name="penulis" value="<?= $buku->penulis; ?>" required>
        </div>
        <div class="form-group">
            <label for="kategori">Kategori:</label>
            <select class="form-control" id="kategori" name="kategori" required>
                <option value="">Pilih Kategori</option>
                <option value="fiksi" <?= ($buku->kategori == 'fiksi') ? 'selected' : ''; ?>>Fiksi</option>
                <option value="non-fiksi" <?= ($buku->kategori == 'non-fiksi') ? 'selected' : ''; ?>>Non-Fiksi</option>
                <option value="sains" <?= ($buku->kategori == 'sains') ? 'selected' : ''; ?>>Sains</option>
                <option value="teknologi" <?= ($buku->kategori == 'teknologi') ? 'selected' : ''; ?>>Teknologi</option>
                <!-- Tambahkan opsi kategori lainnya sesuai kebutuhan -->
            </select>
        </div>
        <input type="submit" class="btn btn-primary btn-md" value="Simpan Perubahan" name="edit">
    </form>
</section>
```

- Tambahkan fungsi edit di controllers

```
public function edit($id)
{
```

```

$data['buku'] = $this->M_buku->tampil_buku_perId($id);

if (empty($data['buku'])) {
    show_404();
}

$data['judul'] = 'Edit Buku';

$this->form_validation->set_rules('judul', 'Judul', 'required');
$this->form_validation->set_rules('penulis', 'Penulis',
'required');
$this->form_validation->set_rules('kategori', 'Kategori',
'required');

if ($this->form_validation->run() === FALSE) {
    $this->load->view('v_header', $data);
    $this->load->view('buku/edit_buku', $data);
    $this->load->view('v_footer');
} else {
    // $this->M_buku->update_buku($id);

    $data = array(
        'judul' => $this->input->post('judul'),
        'penulis' => $this->input->post('penulis'),
        'tgl_publish' => date('Y-m-d'),
        'kategori' => $this->input->post('kategori')
    );
    $this->M_buku->update_buku($id, $data);
    redirect('buku/data_buku');
}
}

```

- Pada Saat di Load tampilan Halaman akan menjadi seperti gambar dibawah ini pastikan id yang dipanggil adalah id yang dipilih pada data di tabel.

Rental Buku

Membuat Template Sederhana dengan CodeIgniter

Home About Buku

Edit Buku

Judul Buku:

Web Programming 1

Penulis:

Hendri

Kategori:

Teknologi

Rental Buku

- Supaya perintah edit dapat berjalan sesuai yang diharapkan, maka di model perlu ditambahkan fungsi untuk menjalankan perintah perubahan, sehingga perlu ditambahkan fungsi **update_buku** sebagai berikut

```
// Edit Buku
public function update_buku($id, $data)
{
    $this->db->where('id', $id);
    return $this->db->update('buku', $data);
}
```

- Jalankan dengan salah satu contoh pengeditan

9. Membuat Halaman Hapus

Pada Halaman hapus tidak perlu membuat view untuk melakukan perintah hapus cukup dengan membuat fungsi di controller dan fungsi di model untuk menjalankan logika penghapusannya. Pada Link yang di tampilkan pada data tabel adalah sebagai berikut

```
<a href="= base_url('buku/delete/' . $book-&gt;id); ?" class="btn btn-danger btn-sm" onclick="return confirm('Apakah Anda yakin ingin menghapus buku ini?');">Hapus</a>
```

- Pesan **onclick="return confirm('Apakah Anda yakin ingin menghapus buku ini?')** digunakan untuk meneruskan proses hapus apakah yakin atau tidak jika sudah yakin maka perintah hapus akan dilaksanakan oleh sistem.
- Membuat fungsi hapus pada controllers

```
public function delete($id)
{
    $this->M_buku->hapus_buku($id);
    redirect('buku/data_buku');
}
```

- Membuat Logika Hapus pada model

```
// Hapus Buku
public function hapus_buku($id)
{
    $this->db->where('id', $id);
    return $this->db->delete('buku');
}
```

- Jalankan fungsi hapus pada data anda dan pastikan data terhapus sesuai dengan harapan