



MODUL WEB PROGRAMMING 1



**DISUSUN OLEH :
HENDRI MAHMUD NAWAWI**

**UNIVERSITAS NUSA MANDIRI
FAKULTAS TEKNOLOGI INFORMASI
PROGRAM STUDI INFORMATIKA**

Kata Pengantar

Puji syukur ke hadirat Tuhan Yang Maha Esa, atas segala rahmat dan karunia-Nya, sehingga Modul Pembelajaran Web Programming 1 ini dapat diselesaikan dengan baik. Modul ini disusun sebagai bahan ajar untuk mata kuliah Web Programming pada program studi terkait, dengan harapan dapat menjadi panduan yang efektif dalam memperkenalkan konsep dasar pemrograman web.

Materi dalam modul ini disusun secara sistematis, mulai dari pengenalan dasar pemrograman web seperti HTML, CSS, dan JavaScript, hingga implementasi kode program menggunakan framework CodeIgniter. Penyusunan modul ini telah disesuaikan dengan Rencana Pembelajaran Semester (RPS) yang berlaku, agar mahasiswa dapat mengikuti pembelajaran secara terstruktur dan sesuai dengan capaian pembelajaran.

Modul ini diharapkan dapat membantu mahasiswa dalam memahami dan mengaplikasikan pengetahuan pemrograman web secara bertahap. Setiap bab dilengkapi dengan contoh-contoh praktis dan latihan yang bertujuan untuk meningkatkan keterampilan teknis dalam membangun situs web yang dinamis dan interaktif.

Ucapan terima kasih saya sampaikan kepada semua pihak yang telah membantu dalam proses penyusunan modul ini. Semoga modul ini memberikan manfaat dan menjadi referensi yang berguna dalam pengembangan keahlian di bidang Web Programming.

Akhir kata, saya berharap modul ini tidak hanya menjadi pedoman belajar, tetapi juga inspirasi bagi mahasiswa untuk terus mengeksplorasi dan mengembangkan kemampuan di bidang teknologi web.

Jakarta, April 2024

Penyusun

DAFTAR ISI

Kata Pengantar.....	2
DAFTAR ISI	3
Bagian 1 – Pendahuluan	6
Bagian 2 – Pengenalan Tipe Data	22
Bagian 3 - Instalasi & Konfigurasi CodeIgniter 3	30
Bagian 4 - Struktur MVC di CodeIgniter	33
Bagian 5 - Membuat Template Sederhana	35
Bagian 6 - Membuat Database dan Tabel.....	42
Bagian 7 - Membuat Create, Read, Update dan Delete (CRUD)	44
Bagian 8 - Membuat Template dengan menggunakan bootstrap	54
TUGAS...!!!	64
DAFTAR PUSTAKA.....	65

Penjelasan Kontrak Kuliah, Penegasan Tugas Project dan Persiapan Aplikasi Pustaka Booking

Deskripsi:

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. Dalam hal ini kasus yang diangkat adalah perpustakaan.

Tujuan Pembelajaran:

Setelah melakukan bagian ini mahasiswa mampu:

1. Memahami proyek yang harus mereka buat dan presentasikan
2. Merancang, membuat, dan memodifikasi database
3. Melakukan Koneksi database.

A. Penjelasan Kontrak Kuliah dan Penegasan Tugas Project

1. Tugas Project (Kelompok)

Tugas project diadakan untuk memperoleh nilai UTS dan UAS, dengan kata lain tugas project ini sebagai pengganti UTS dan UAS. Tugas ini dikerjakan secara kelompok dengan maksimal 1 kelompok sebanyak 5 mahasiswa atau disesuaikan dengan jumlah mahasiswa dalam satu kelas.

2. Pembagian Kelompok

Untuk pembagian kelompok ditentukan oleh dosen pengampu mata kuliah, atau dapat disesuaikan dengan kelompok tugas pada mata kuliah yang lain.

3. Bentuk Tugas (Kelompok) Bentuk Tugas Project :

- a. Project merupakan program aplikasi jadi mulai dari *Back-End*, *Front-End* , dengan menambahkan laporan pada project aplikasi, tema project dapat disesuaikan dengan tema yang diambil pada mata kuliah lain.
- b. Pada pertemuan 14 dan 15, masing-masing kelompok wajib demo program yang telah dikerjakan (Presentasi).

4. Tugas Mingguan

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

Bagian 1 – Pendahuluan

B. Dasar-Dasar Website

1. Internet

Internet merupakan “kependekan dari kata “*internetwork*”, yang berarti rangkaian komputer yang terhubung menjadi beberapa rangkaian jaringan”. Sistem komputer terhubung secara global dan menggunakan TCP/IP sebagai *protocol*. Secara umum *internet* dapat diartikan sebagai pertukaran informasi dan komunikasi. Semua informasi bisa didapatkan dengan mudah dan bebas di *internet* tanpa ada batasan.

Ada beberapa istilah yang sering digunakan apabila anda bekerja dengan *internet* diantaranya yaitu:

1. *World Wide Web (WWW)*

WWW merupakan kumpulan *web server* diseluruh dunia yang dapat menyediakan data dan informasi untuk dapat digunakan secara massal.

2. *Website*

Website atau situs *web* merupakan sebuah alamat tertentu di *WWW* yang menyediakan informasi tertentu. Untuk membuka sebuah situs *web*, anda dapat menggunakan *browser*.

3. *Web Pages (Halaman Web)*

Web pages atau halaman *web* merupakan bagian dari situs *web*, apabila situs *web* diumpamakan merupakan sebuah buku, maka halaman *web* merupakan lembaran-lembaran kertas penyusun buku tersebut.

4. *Home Page (Halaman Muka)*

Homepage merupakan halaman muka dari sebuah situs *web*, atau ibarat *cover* muka sebuah buku. *Homepage* biasanya berupa *outline* dari isi situs *web* yang bersangkutan.

5. *Browser*

Browser adalah aplikasi yang digunakan untuk berselancar didunia *internet*. *Browser* dapat memandu pengguna *internet* untuk berpindah antar situs *web* dengan mudah.

6. *URL (Universal Resource Locator)*

URL merupakan suatu alamat yang menunjukkan sebuah halaman tertentu *internet*. Contoh URL adalah: *http://www.google.com*

7. HTTP (*Hypertext Transfer Protocol*)
HTTP adalah bagian dari sebuah URL yang mengidentifikasikan lokasi *web*, dan digunakan dalam protokol HTML.
8. DNS (*Domain Name System*)
DNS merupakan sistem *database* terdistribusi yang tidak banyak dipengaruhi oleh bertambahnya *database*. DNS menjamin informasi *host* terbaru akan disebarkan ke jaringan bila diperlukan.
9. TCP/IP (*Transmission Control Protocol / Internet Protocol*)
TCP/IP (*Transmission Control Protocol/Internet Protocol*) merupakan metode-metode yang digunakan untuk menghubungi server. TCP/IP merupakan bahasa standarisasi untuk *internet*.
10. IP (*Internet Protocol*)
IP (*Internet Protocol*) merupakan protokol yang digunakan dalam *internet*, secara teknis bermakna suatu bentuk pengisian dan pengalamatan data-data dan informasi yang akan dikirim melalui *internet*.
11. *Hyperlink*
Hyperlink atau disebut *link* saja merupakan sebuah fasilitas yang sangat berperan mempopulerkan pengguna *internet*, karena mampu mereferensikan sebuah teks atau gambar ke alamat lain di *internet*.
12. *Web Browser*
Menggunakan *web browser* mudah, yang diperlukan hanyalah Anda harus memiliki alamat *web* yang akan dibuka. Alamat ini biasa disebut dengan *Uniform Resource Locator* (URL). Di dalam sistem operasi *Windows* Anda juga terdapat program *web browser* sertaan, yaitu *Internet Explorer*. Namun demikian diluar terdapat banyak program *alternative web browser* yang sebagian besar bersifat gratis, seperti *Netscape*, *Firefox*, *Opera*, *Avant Browser*, dan seterusnya.

2. Perangkat Lunak Web Server

Web Server adalah sebuah perangkat lunak server yang berfungsi menerima

permintaan HTTP atau HTTPS dari Client yang dikenal dengan web browser dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML. Server web yang terkenal diantaranya adalah:

- a. Apache, web server antar platform
 - 1) XAMPP
 - 2) PHPTriad; *discontinued*
 - 3) Apache2Triad
 - 4) Laragon (bisa di download [disini](#))
- b. Internet Information Service (IIS), hanya dapat berjalan di sistem operasi MS Windows
- c. Pengertian HTML (Hypertext Markup Language)

Hypertext Markup Language (HTML) adalah sebuah bahasa untuk menampilkan konten di web. HTML sendiri adalah bahasa pemrograman yang bebas, artinya tidak dimiliki oleh siapapun, pengembangannya dilakukan oleh banyak orang di banyak Negara dan bias dikatakan sebagai sebuah bahasa yang dikembangkan bersama-sama secara global.

Sebuah dokumen HTML sendiri adalah dokumen teks yang dapat diedit oleh editor teks apapun. Dokumen HTML punya beberapa elemen yang dikelilingi oleh tag-teks yang dimulai dengan symbol < dan berakhir dengan sebuah symbol >.

Editor teks yang digunakan oleh penyusun adalah menggunakan Visual Studio Code (VS Code) dan bahasa pemrograman PHP Versi 8

C. Struktur Dasar HTML

Elemen HTML dimulai dengan tag awal, yang diikuti dengan isi elemen dan tag akhir. Tag berakhir termasuk simbol / diikuti oleh tipe elemen, misalnya </HEAD>. Sebuah elemen HTML dapat bersarang di dalam elemen lainnya. Sebuah dokumen

```
1 <html>
2   <head>
3     <title>Judul Halaman Web</title>
4   </head>
5   <body>
6
7 </body>
8 </html>
```

HTML standar terlihat seperti ini :

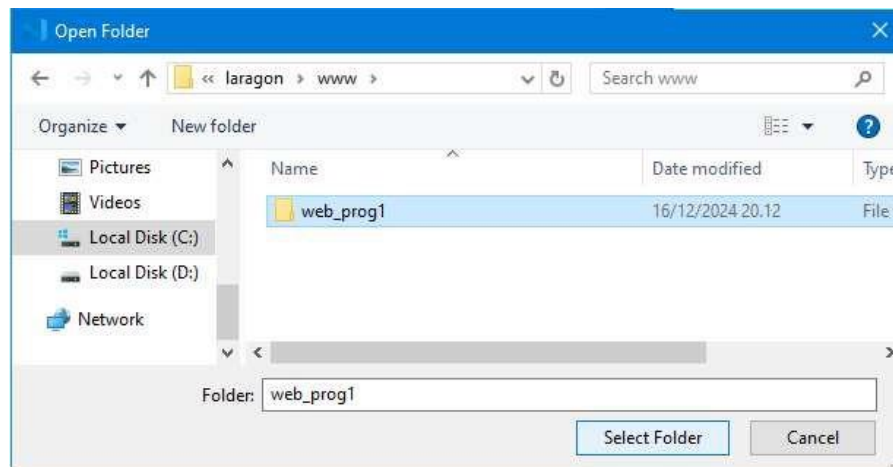
Keterangan :

1. Tag HTML secara default dimulai dari `<HTML>` dan diakhiri dengan `</HTML>`.
2. Tag `<HEAD> ... </HEAD>` merupakan tag kepala sebelum badan. Tag kepala ini akan terlebih dulu dieksekusi sebelum tag badan. Di dalam tag ini berisi tag `<META>` dan `<TITLE>`. Tag `<META>` merupakan informasi atau header suatu dokumen HTML. Atribut yang dimiliki oleh tag ini antara lain:
 - a. `HTTP_EQUIV`, atribut ini berfungsi untuk menampilkan dokumen HTML secara otomatis dalam jangka waktu tertentu.
 - b. `CONTENT`, atribut ini berisi informasi tentang isi document HTML yang akan dipanggil.
 - c. `NAME`, atribut ini merupakan identifikasi dari meta itu sendiri. Tag `<META>` dalam suatu document HTML boleh ada maupun tidak.
3. Tag `<TITLE> ... </TITLE>` adalah tag judul. Sebaiknya setiap halaman web memiliki judul, dan judul tersebut dituliskan di dalam `<TITLE> ... </TITLE>`. Judul ini akan muncul dalam titlebar dari browser.
4. Tag `<BODY> ... </BODY>` adalah tag berisi content dari suatu halaman web.

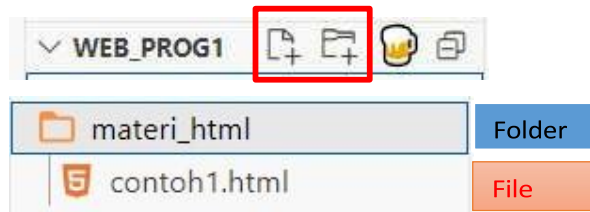
Contoh penggunaan script HTML

Buat lembar baru pada VS Code, cara pembuatan file di VS Code:

1. Buka Vs Code, kemudian pilih File – Open Folder
2. Pilih folder yang sudah dibuat sebelumnya



3. Buat file baru atau folder baru (optional) beri nama contoh1.html



kemudian ketikkan perintah di bawah ini. Simpan dengan nama Contoh1.php

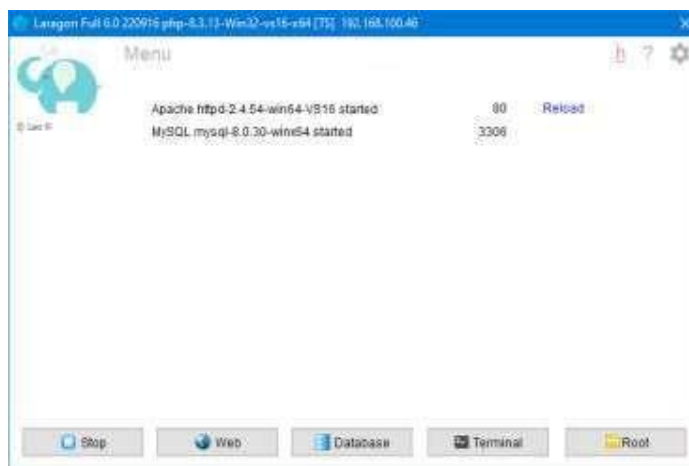
```
web_prog1 > materi_html_1 > contoh1.html > ...  
1  <html>  
2      <head>  
3          <title>Contoh 1</title>      → Judul Web  
4      </head>  
5  <body>  
6      Hello World...<br>  
7      Selamat Belajar Web Programming  
8  </body>  
9  </html>
```

Kemudian simpan file di atas di dalam folder c:\Laragon\www\ buat folder baru untuk menyimpan file di dalam folder www (dalam hal ini modul menggunakan folder web_prog1 dan didalamnya ada folder materi_html). Simpan file dengan nama contoh1.html. Pembuatan nama file pada saat penyimpanan harus diakhiri dengan extension “.html”

Sebelum menjalankan script tersebut, jalankan Laragon yang telah diinstall.

Kemudian klik Start All 

Lihat gambar di bawah ini :



Untuk melihat hasil dari file di atas dapat menggunakan browser Mozilla, google chrome, internet explorer atau jenis browser lain. Ketikkan pada address bar “Localhost\Nama Folder Penyimpanan\”, kemudian pilih file contoh1.html



Hasilnya :



Kode-kode dalam HTML biasanya disebut TAG. Tag adalah sesuatu yang digunakan untuk menandai elemen-elemen dalam suatu dokumen HTML. Tag dalam HTML terdiri dari tanda lebih kecil (<), tanda lebih besar (>), dan garismiring (/). Biasanya Tag dituliskan secara berpasangan, misanya <h1> dan </h1>. Tag yang tidak menggunakan garis miring (/) adalah Tag pembuka atau awal elemen. Sedangkan yang Tag yang mengandung garis miring (/) adalah penutup elemen atau akhir elemen. Namun, ada juga Tag yang dalam pemakaiannya tidak berpasangan, diantaranya adalah :

- Tag untuk ganti paragraph yaitu <p>
- Tag untuk ganti baris atau line break yaitu

- Tag untuk garis datar yaitu <hr>
- Tag list item yaitu

Untuk tag yang tidak berpasangan diatas, sebaiknya tetap ditulis menggunakan pasangannya. Hal ini dilakukan untuk mengantisipasi standar rekomendasi HTML kedepannya. Penulisan untuk semua Tag bebas, maksudnya kita bisa menggunakan

huruf besar, huruf kecil, bahkan dicampur (tidak case sensitive). Tapi untuk mengantisipasi standar penulisan Tag, sebaiknya kita menggunakan huruf kecil semua.

Jenis – jenis tag dalam HTML :

*** Tag dasar**

Tag awal	Kegunaan
<html>	Untuk mendefinisikan sebuah dokumen HTML.
<body>	Untuk mendefinisikan body atau isi sebuah dokumen.
<h1> sampai <h6>	Untuk mendefinisikan heading 1 sampai 6
<p>	Untuk mendefinisikan sebuah paragraf
 	Untuk membuat baris baru
<hr>	Untuk memisah antar bagian atau paragraf (horizontal rule)
<!-- diakhiri dengan -->	Untuk mendefinisikan komentar dalam source code

*** Tag Pemformatan**

Tag awal	Kegunaan
	Definisi teks yang ditebalkan
<big>	Definisi teks yang besar ukurannya
	Definisi teks yang ditekan
<i>	Definisi teks yang dicetak miring (italic)
<small>	Definisi teks kecil ukurannya
<u>	Definisi teks yang bergaris bawah
<sub>	Definisi teks yang jadi subscript
<sup>	Definisi teks yang jadi supercript
<ins>	Definisi teks yang disisipkan
	Definisi teks yang dihapus

Tag Computer Output

Tag awal	Kegunaan
<code>	Definisi teks computer code
<kbd>	Definisi teks keyboard
<samp>	Definisi contoh computer code
<tt>	Definisi teks teletype
<var>	Definisi suatu variabel
<pre>	Definisi teks preformatted

*** Tag Citation, Quotation, Definition**

Tag awal	Kegunaan
<abbr>	Definisi suatu singkatan
<acronym>	Definisi suatu akronim
<address>	Definisi penulisan alamat
<bdo>	Definisi arah penulisan
<blockquote>	Definisi quotation panjang
<q>	Definisi quotation pendek
<cite>	Definisi suatu citation
<dfn>	Definisi istilah (term)

*** Tag Link**

Tag awal	Kegunaan
<a>	Mendefinisikan suatu link

*** Tag Image**

Tag awal	Kegunaan
	Definisi sebuah image dalam dokumen
<map>	Definisi sebuah image map
<area>	Definisi suatu area dalam image map

Tag untuk list

Start Tag	Kegunaan
	Mendefinisikan sebuah list ordered
	Mendefinisikan sebuah list unordered
	Mendefinisikan sebuah item dalam list
<dl>	Mendefinisikan sebuah list definisi
<dt>	Mendefinisikan sebuah istilah list definisi

Contoh script penggunaan Tag HTML

Buat lembar baru pada VS Code, kemudian ketikkan perintah di bawah ini. Simpan dengan nama Contoh2.html

```
contoh2.html x
materi_html > contoh2.html > ...
1 <html>
2 <head>
3 <title>Contoh 2</title>
4 </head>
5 <body bgcolor="#00CCFF" text="red">
6 <p> Ini adalah contoh penggunaan formatting TAG dalam HTML <br />
7   masing-masing TAG memiliki atribut masing-masing<br />
8 </p>
9 <font color="black" face="verdana">Ini juga termasuk contoh penggunaan formatting TAG<br /></font>
10 <p>Ini adalah contoh Bullet</p>
11 <ol type="A">
12   <li>Contoh 1</li>
13   <li>Contoh 2</li>
14 </ol>
15 <h1><marquee bgcolor="#000099">
16   Ini juga salah satu penggunaan Tag</marquee></h1>
17 </body>
18 </html>
```

Simpan dengan nama contoh2.html

Hasil Tampilan



D. Pembuatan Tabel Menggunakan HTML

Tabel penting peranannya dalam halaman Web, selain untuk menampilkan teks atau gambar dalam format lajur dan kolom bias juga menggunakan tabel untuk membantu me-layout tampilan halaman.

Tabel merupakan sebuah kotak yang terdiri atas baris/row dan kolom.column. Untuk membuat tabel, anda menggunakan tag <table> dan menutupnya dengan tag

</table>. Anda bisa juga menambahkan atribut lain di tag <table> pembuka. Misalnya menentukan warna, border, dan sebagainya.

Di dalam tag <table> ada beberapa tag lain yang perlu dipahami, yaitu :

a. Tag <tr>

Artinya tag untuk menuliskan baris biasa di tabel. TR singkatan dari Table Row.

b. Tag <td>

Artinya tag untuk menuliskan kotak di dalam baris, makanya tag <td> ada di dalam tag <tr>. TD singkatan dari Table Data.

c. Tag <th>

Artinya tag untuk menuliskan kotak biasa seperti <td>, namun untuk header tabel. TH singkatan dari Table Header.

Menggabungkan sel

Sel-sel tabel secara normal memiliki lebar dan tinggi yang sama. Jika kita ingin membuat sebuah sel memiliki lebar atau tinggi yang berbeda dari sel-sel lainnya, maka satu- satunya cara yang bisa kita lakukan adalah dengan menggabungkan beberapa sel menjadi satu. Cara ini disebut merge atau penggabungan sel.

Untuk menggabungkan sel-sel tabel ini diperlukan atribut rowspan atau colspan. Atribut rowspan digunakan untuk menggabungkan sel-sel tabel pada kolom yang sama. Atribut colspan untuk menggabungkan sel-sel tabel pada baris yang sama. Berikut contoh penggabungan kedua jenis :

1. Secara Vertikal (Rowspan)

Tabel dengan kode HTML dibawah ini sel-sel kolom pertama akan digabung:

```
<table>
  <tr>
    <td>...</td>
    <td>...</td>
  </tr>
  <tr>
    <td>...</td>
    <td>...</td>
  </tr>
  <tr>
    <td>...</td>
    <td>...</td>
  </tr>
</table>
```


Setelah digabung maka kondisi kode HTML menjadi seperti berikut:

```

<table>
  <tr>
    <td rowspan="3">...</td>
    <td>...</td>
  </tr>
  <tr>
    <td>...</td>
  </tr>
  <tr>
    <td>...</td>
  </tr>
</table>

```


2. Secara Horizontal (Colspan)

Tabel dengan kode HTML dibawah ini sel-sel baris pertama akan digabung:

```

<table>
  <tr>
    <td>...</td>
    <td>...</td>
  </tr>
  <tr>
    <td>...</td>
    <td>...</td>
  </tr>
  <tr>
    <td>...</td>
    <td>...</td>
  </tr>
</table>

```


Setelah digabung maka kondisi kode HTML menjadi seperti berikut:

```

<table>
  <tr>
    <td colspan="2">...</td>
  </tr>
  <tr>
    <td>...</td>
    <td>...</td>
  </tr>
  <tr>
    <td>...</td>
    <td>...</td>
  </tr>
</table>

```


Contoh script pembuatan tabel

Buat lembar baru pada Notepad, kemudian ketikkan perintah di bawah ini. Simpan dengan nama Contoh3.html

```
contoh3.html X
materi_html > contoh3.html > ...
1  <html>
2  <head>
3  <title> Contoh 3 - Penggunaan Tabel</title>
4  </head>
5  <body>
6  <h2>Tabel Data Mahasiswa Universitas Nusa Mandiri</h2>
7  <table border="1" bgcolor="pink">
8  <tr>
9      <th>NIM</th>
10     <th>Nama</th>
11     <th>Alamat</th>
12     <th>Tempat, Tanggal Lahir</th>
13     <th>Jurusan</th>
14 </tr>
15 <tr>
16     <td>11230001</td>
17     <td>Maruloh</td>
18     <td>Cengkareng</td>
19     <td>Jakarta, 28 Desember 1988</td>
20     <td>Sistem Informasi</td>
21 </tr>
22 <tr>
23     <td>11230002</td>
24     <td>Hafidz</td>
25     <td>Bogor</td>
26     <td>Jakarta, 01 Januari 1994</td>
27     <td>Sistem Informasi</td>
28 </tr>
29 <tr>
30     <td>12230003</td>
31     <td>Hikmatullah</td>
32     <td>Depok</td>
33     <td>Jakarta, 28 Februari 1997</td>
34     <td>Informatika</td>
35 </tr>
36 </table>
37 </body>
38 </html>
```

Jika dilihat di browser, maka terlihat sebagai berikut :

Tabel Data Mahasiswa Universitas Nusa Mandiri

NIM	Nama	Alamat	Tempat, Tanggal Lahir	Jurusan
11230001	Maruloh	Cengkareng	Jakarta, 28 Desember 1988	Sistem Informasi
11230002	Hafidz	Bogor	Jakarta, 01 Januari 1994	Sistem Informasi
12230003	Hikmatullah	Depok	Jakarta, 28 Februari 1997	Informatika

Penggunaan Cellpadding dan cellspacing

Buat lembar baru pada VS Code, kemudian ketikkan perintah di bawah ini. Simpan dengan nama tabelcell.html

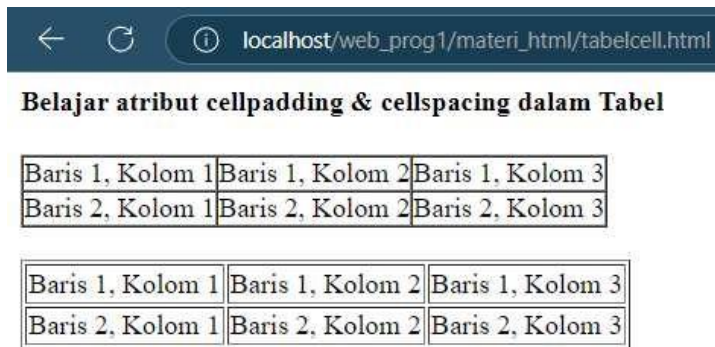
```
tabelcell.html X
materi_html > tabelcell.html > html > body
1  <html>
2  <head>
3  <title>Penggunaan atribut Cellpadding dan cellspacing dalam Tabel</title>
4  </head>
5  <body>
6  <h4>Belajar atribut cellpadding & cellspacing dalam Tabel</h4>
7  <table border="1" cellspacing="0" cellpadding="0">
8      <tr>
9          <td>Baris 1, Kolom 1</td>
10         <td>Baris 1, Kolom 2</td>
11         <td>Baris 1, Kolom 3</td>
12     </tr>
13     <tr>
14         <td>Baris 2, Kolom 1</td>
15         <td>Baris 2, Kolom 2</td>
16         <td>Baris 2, Kolom 3</td>
17     </tr>
18 </table>
19 <br />
```

```

20 <table border="1" cellspacing="2" cellpadding="1">
21   <tr>
22     <td>Baris 1, Kolom 1</td>
23     <td>Baris 1, Kolom 2</td>
24     <td>Baris 1, Kolom 3</td>
25   </tr>
26   <tr>
27     <td>Baris 2, Kolom 1</td>
28     <td>Baris 2, Kolom 2</td>
29     <td>Baris 2, Kolom 3</td>
30   </tr>
31 </table>
32 </body>
33 </html>

```

Hasil di browser :



Penggunaan Rowspan dan colspan

Buat lembar baru pada Notepad, kemudian ketikkan perintah di bawah ini. Simpan dengan nama tabelspan.html

```

tabelspan.html X
materi_html > tabelspan.html > ...
1  <html>
2  <head>
3  <title>Contoh Penggunaan Atribut Colspan dan Rowspan Tag Tabel</title>
4  </head>
5  <body>
6  <h2>Contoh atribut colspan dan rowspan </h2>
7  <table border="1" cellspacing="0" cellpadding="4">
8      <tr>
9          <td>Baris 1, Kolom 1</td>
10         <td>Baris 1, Kolom 2</td>
11         <td>Baris 1, Kolom 3</td>
12     </tr>
13     <tr>
14         <td>Baris 2, Kolom 1</td>
15         <td colspan="2">Baris 2, Kolom 2 & 3</td>
16     </tr>
17     <tr>
18         <td rowspan="2">Baris 3 & 4, Kolom 1</td>
19         <td>Baris 3, Kolom 2</td>
20         <td>Baris 3, Kolom 3</td>
21     </tr>
22     <tr>
23         <td>Baris 4, Kolom 2</td>
24         <td>Baris 4, Kolom 3</td>
25     </tr>
26 </table>
27 </body>
28 </html>

```

Hasil di browser :



Contoh atribut colspan dan rowspan

Baris 1, Kolom 1	Baris 1, Kolom 2	Baris 1, Kolom 3
Baris 2, Kolom 1	Baris 2, Kolom 2 & 3	
Baris 3 & 4, Kolom 1	Baris 3, Kolom 2	Baris 3, Kolom 3
	Baris 4, Kolom 2	Baris 4, Kolom 3

Tugas 01:

1. Buatlah script html sehingga menghasilkan tampilan berikut ini :



2. Buat script HTML untuk bentuk tampilan di bawah ini :

DAFTAR PRODUK		
	Nama Produk	Canon EOS M10 Kit EF-M 15-45mm
	Harga	Rp 4,899,000
	Fitur Produk	• Kamera mirrorless • Efektifitas Piksel : 18 MP • Tipe sensor : CMOS DIGIC 6 • Layar : 3.0 Inch • Memiliki lampu flash dengan jarak jangkauan hingga 5 meter

Bagian 2 – Pengenalan Tipe Data

Pengenalan PHP, Variabel, Konstanta, Tipe Data, Komentar dan Operator

Deskripsi:

Pertemuan ini membahas mengenai pemrograman dengan PHP (PHP Hypertext Preprocessor) dan perbedaannya dengan penggunaan HTML. Pertemuan ini juga membahas mengenai penggunaan variabel, konstanta, tipe data, komentar dan operator dalam pemrograman.

Tujuan Pembelajaran:

Setelah melakukan bagian ini mahasiswa mampu:

1. Memahami konsep dasar PHP
2. Memahami perbedaan HTML dan PHP
3. Memahami dan menggunakan variabel, konstanta, tipe data, dan komentar dalam PHP
4. Memahami dan menggunakan perintah operator dalam PHP

A. Pengertian PHP (*PHP Hypertext Preprocessor*)

PHP atau *PHP Hypertext Preprocessor* adalah sebuah bahasa script berbasis server (server-side) yang mampu mem-parsing kode php dari kode web dengan ekstensi .php, sehingga menghasilkan tampilan website yang dinamis di sisi client (browser). Dengan PHP, anda bisa menjadikan halaman HTML menjadi lebih powerful dan bisa dipakai sebagai aplikasi lengkap, misalnya untuk beragam aplikasi cloud computing.

PHP adalah bahasa script yang sangat cocok untuk pengembangan web dan dapat dimasukkan ke dalam HTML. PHP awalnya dikembangkan oleh seorang programmer bernama Rasmus Lerdorf pada tahun 1995, namun semenjak itu selalu dikembangkan oleh kelompok independen yang disebut Group PHP dan Kelompok ini juga yang mendefinisikan standar de facto untuk PHP karena tidak ada spesifikasi formal. Saat ini pengembangannya dipimpin oleh duo maut, Andi Gutmans dan Zeev Suraski.

Yang menyebabkan PHP banyak dipakai oleh banyak orang adalah karena PHP adalah perangkat lunak bebas (Open Source) yang dirilis di bawah lisensi PHP. Artinya untuk menggunakan bahasa pemrograman ini gratis, bebas, dan tidak terbuka.

B. Memasukkan Kode PHP

Tidak seperti halaman HTML biasa, kode PHP tidak akan diberikan oleh server secara langsung ketika ada permintaan dari client (browser), namun melalui pemrosesan dari sisi server, makanya PHP disebut skrip server-side.

Kode PHP dimasukkan ke dalam kode HTML dengan cara menyelipkannya di dalam kode HTML. Untuk membedakan kode PHP dengan kode HTML, di depan kode PHP tersebut diberi tag pembuka dan diakhir kode PHP diberi tag penutup.

Dengan adanya kode PHP, sebuah halaman web bisa melakukan banyak hal yang dinamis, seperti mengakses database, membuat gambar, membaca dan menulis file, dan sebagainya. Hasil akhir pengolahan kode PHP akan dikembalikan lagi dalam bentuk kode HTML untuk ditampilkan di browser. Ada 4 jenis tag yang bisa digunakan untuk memasukkan kode PHP.

Jenis Tag	Tag Pembuka	Tag Penutup
Tag Standar	<code>< ? php</code>	<code>? ></code>
Tag Pendek	<code>< ?</code>	<code>? ></code>
Tag ASP	<code>< %</code>	<code>% ></code>
Tag Script	<code><script language = "php"></code>	<code></script></code>

Yang dapat langsung diterapkan disemua platform adalah tag standard dan tag script. Di dalam modul ini bahasa pemrograman yang digunakan adalah PHP Versi 8 sehingga jenis tag yang harus digunakan adalah tag standar. Untuk tag lainnya perlu penyetingan di server oleh administrator server.

Contoh Script PHP

Buka file baru di Notepad. Kemudian ketikkan script seperti di bawah ini :



```
contoh4.php
materi_php > contoh4.php
1  <?php
2  echo "Hello World...<br>";
3  echo "Selamat Belajar PHP";
4  ?>
```

Simpan file dengan nama contoh4.php

Untuk melihat hasilnya buka browser masuk ke dalam localhost dan folder penyimpanan.

Pilih file contoh4.php maka akan tampil hasilnya :



Contoh4.php merupakan contoh script php yang berdiri sendiri tanpa ada tambahan script yang lain. Perintah echo merupakan perintah yang digunakan untuk mencetak. Script PHP bisa juga digabung dalam tag HTML.

Perbedaan HTML dengan PHP

- HTML dapat diakses langsung tanpa melalui akses server saat ada permintaan dari client(browser)
- PHP harus di akses melalui server saat ada permintaan dari client(browser)



Dari 2 gambar di atas dapatkah anda melihat perbedaannya, tanpa melihat extension nama filenya?

Ya, untuk file dengan extension html, kita dapat melihat hasilnya langsung di browser,

tanpa harus menjalankan akses server. Namun, untuk file dengan extension php, kita harus menjalankannya melalui akses server, yaitu localhost, dan penyimpanan filenya pun, disimpan pada htdocs yang ada di folder xampp/htdocs

C. Variabel

Variabel merupakan sebuah istilah yang menyatakan sebuah tempat yang menampung nilai-nilai tertentu di mana nilai di dalamnya bisa diubah-ubah. Variable penting karena tanpa adanya variable tidak bisa menyimpan nilai tertentu untuk diolah. Variabel ditandai dengan adanya tanda dolar (\$) yang kemudian bisa diikuti dengan angka, huruf, dan underscore. Namun variable tidak bisa mengandung spasi.

Berikut ini contoh pendefinisian variable. Untuk mendefinisikan variable, hanya perlu menuliskannya maka otomatis variable dikenali oleh PHP.

\$nama

\$no_telp

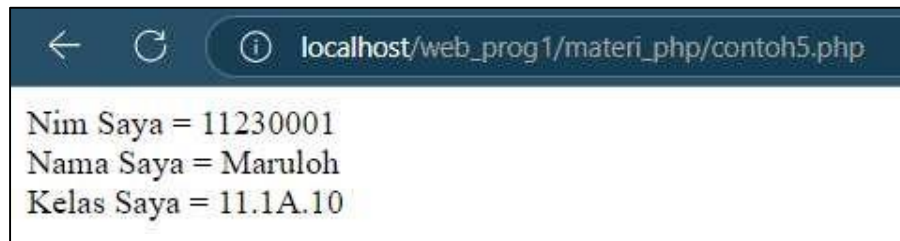
\$_pekerjaan

Variable merupakan tempat untuk menyimpan data dalam tipe tertentu, variable bisa berupa null (belum ada isinya), angka, string, objek, array, Boolean, dan isinya bisa diubah-ubah nantinya.

contoh5.php:

```
materi_php >  contoh5.php > ...  
1  <?php  
2  $nim = "11230001";  
3  $nama = "Maruloh";  
4  $kelas = "11.1A.10";  
5  
6  echo "Nim Saya  = $nim<br>";  
7  echo "Nama Saya = $nama<br>";  
8  echo "Kelas Saya = $kelas<br>";
```

Hasil :



D. Tipe Data

Berbeda dengan bahasa pemrograman lain, variable di PHP lebih fleksibel. Kita tidak perlu mendefinisikan jenisnya ketika mendefinisikan pertama kali. Ada 6 Tipe data dasar yang dapat diakomodasi di PHP, seperti terlihat di tabel.

Tipe	Contoh	Penjelasan
Integer	134	Semua angka bukan pecahan
Double	5.1234	Nilai pecahan
String	"asep"	Kumpulan karakter
Boolean	False	Salah satu nilai True atau False
Object		Sebuah instance dari class
Array		Larik

Untuk mengetahui tipe data sebuah variable, kita bisa menggunakan perintah `gettype`, misalnya :

```
Print gettype($nama_variabel);
```

Anda juga bisa mengubah jenis variable tertentu dengan perintah :

```
(jenis_variabel) $nama_variabel;
```

Misalnya untuk mengubah variable menjadi string, kita dapat menggunakan perintah:

```
$var_string = (string) $angka;
```

contoh6.php:

```
materi_php > contoh6.php > ...
1      <?php
2      $jumlah = 5;
3      $harga = 20000;
4      $total = $harga * $jumlah;
5      echo "Jumlah Beli : $jumlah <br>";
6      echo "Harga Barang : $harga <br>";
7      echo "Total Bayar : $total <br>";
8      ?>
```

Hasil Tampilan :



```
← ↻ ⓘ localhost/web_prog1/materi_php/contoh6.php
Jumlah Beli : 5
Harga Barang : 20000
Total Bayar : 100000
```

E. Konstanta

Selain variable, sebuah program umumnya juga memungkinkan adanya konstanta. Konstanta fungsinya sama seperti variable namun nilainya statis/konstan dan tidak bisa berubah. Cara untuk mendefinisikan konstanta adalah :

Define (“NAMA_KONSTANTA”, nilai_konstanta);

Setelah didefinisikan, kita dapat langsung menggunakannya dengan mengetikkan nama konstanta tersebut. Nama konstanta umumnya diketik menggunakan huruf besar.

F. Komentar

Program merupakan kegiatan menuliskan bahasa yang dipahami oleh mesin. Walaupun bahasa yang digunakan adalah bahasa tingkat tinggi, namun tent masih tidak semudah dipahami oleh bahasa biasa. Untuk itu kita bisa menggunakan komentar. Berikut ini contoh pembuatan komentar di php.

//komentar satu baris

#ini juga komentar satu baris

/*komentar Banyak

baris Kode di sini
tidak
Dieksekus oleh parser */

Contoh script konstanta & komentar

contoh7.php



```
1 <?php
2 //konstanta untuk nilai judul
3 define("Judul", "Hitung Luas Lingkaran");
4 //konstanta untuk nilai phi
5 define("PHI", 3.14);
6 echo Judul;
7 $r = 10;
8 echo "<br> Jari-jari : $r <br>";
9 $luas = PHI * $r * $r;
10 echo "Luas Lingkaran = $luas";
11 ?>
```

Annotations:

- Line 2: //konstanta untuk nilai judul → Deklarasi komentar
- Line 5: define("PHI", 3.14); → Deklarasi konstanta

Hasil :



G. Operator

Sebuah bahasa pemrograman juga wajib untuk mampu mengolah nilai operand (variable atau konstanta yang dioperasikan) menggunakan operator, seperti menjumlah, membagi, dan sebagainya.

Operator merupakan symbol yang berfungsi untuk melakukan aksi / operasi tertentu terhadap nilai operand yang pada umumnya dari hasil operasi tersebut menghasilkan nilai baru. Sementara operand adalah nilai yang dilibatkan dalam operasi oleh operator.

1. Jenis-Jenis Operator

a) Operator Aritmatika

Operator ini digunakan untuk melakukan perhitungan matematika, sebagian berikut:

Operator	Nama	Contoh	Hasil
+	Penambahan	1+4	5

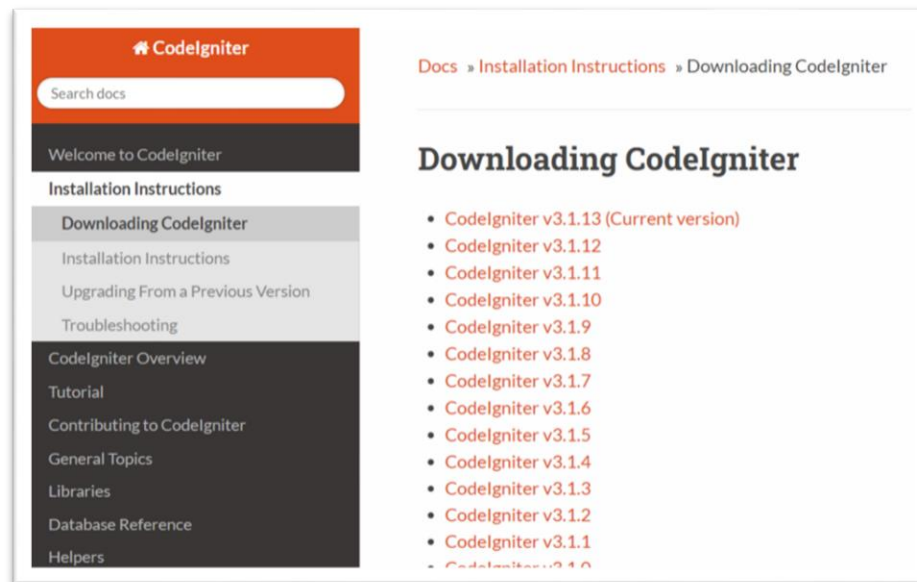
Bagian 3 - Instalasi & Konfigurasi CodeIgniter 3

B. Persiapan Project Pustaka Booking

1. Instalasi Codigniter 3

Kunjungi Situs Resmi pada link CodeIgniter

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



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

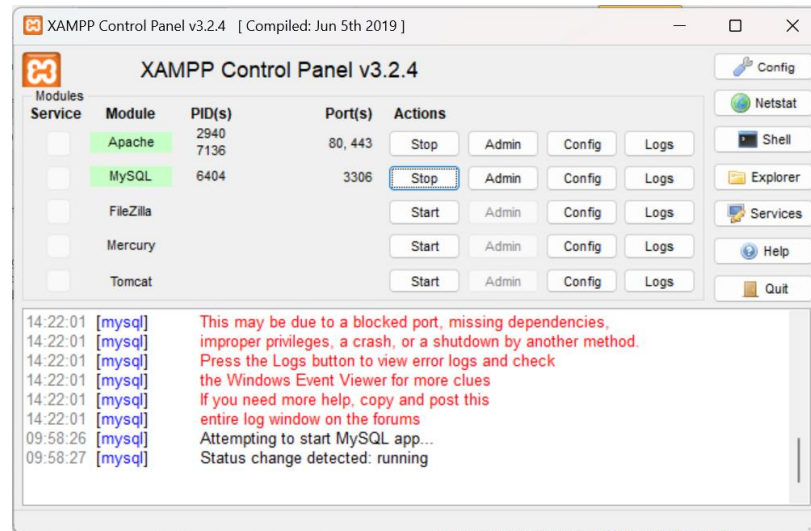
Name	Date modified	Type	Size
CodeIgniter-3.1.13	03/03/2022 20:21	File folder	
CodeIgniter-3.1.13	07/06/2024 9:53	Compressed (zipped)...	693 KB

Ubah Folder CodeIgniter-3.1.13 menjadi **rentalbuku**

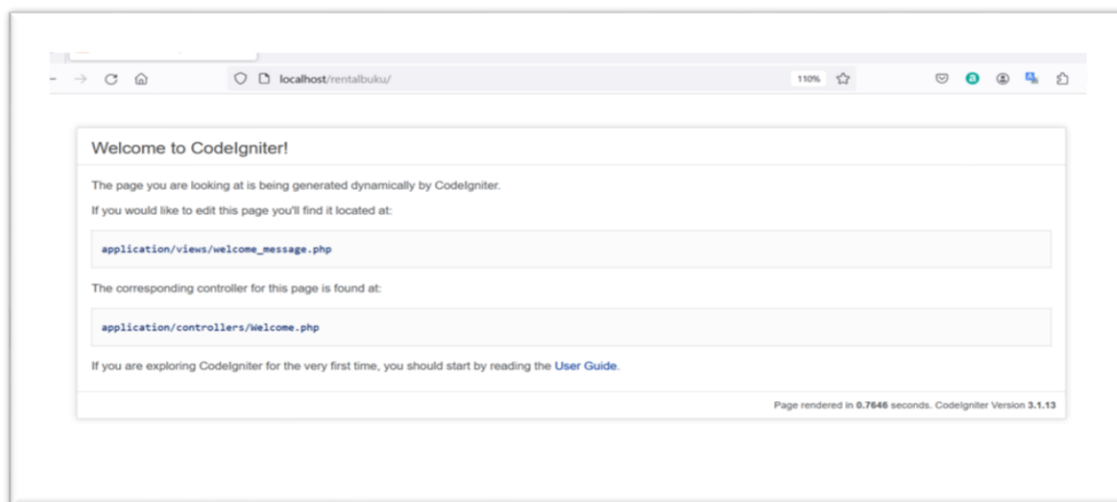
Name	Date modified	Type	Size
rentalbuku	03/03/2022 20:21	File folder	
CodeIgniter-3.1.13	07/06/2024 9:53	Compressed (zipped)...	693 KB

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.

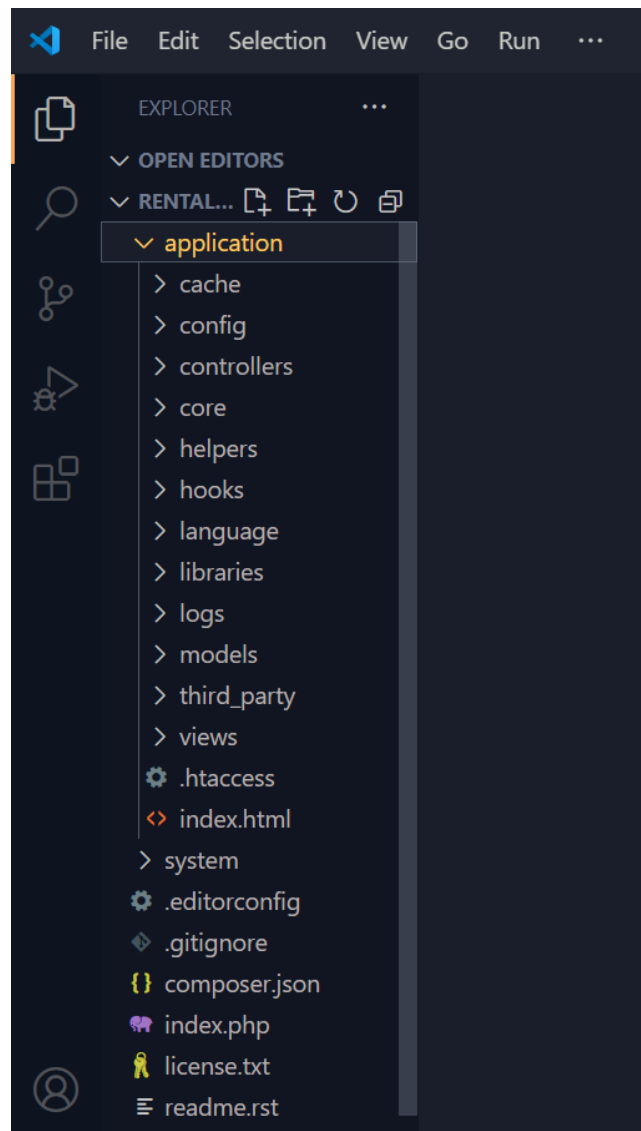


Setelah mengaktifkan **Apache** dan **MySQL** jalankan project kita pada browser pada url <http://localhost/rentalbuku/>. Jika pada tampilan browser sudah tampil seperti gambar dibawah ini maka project CodeIgniter 3 sudah ada di perangkat kita.



2. 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/';
```

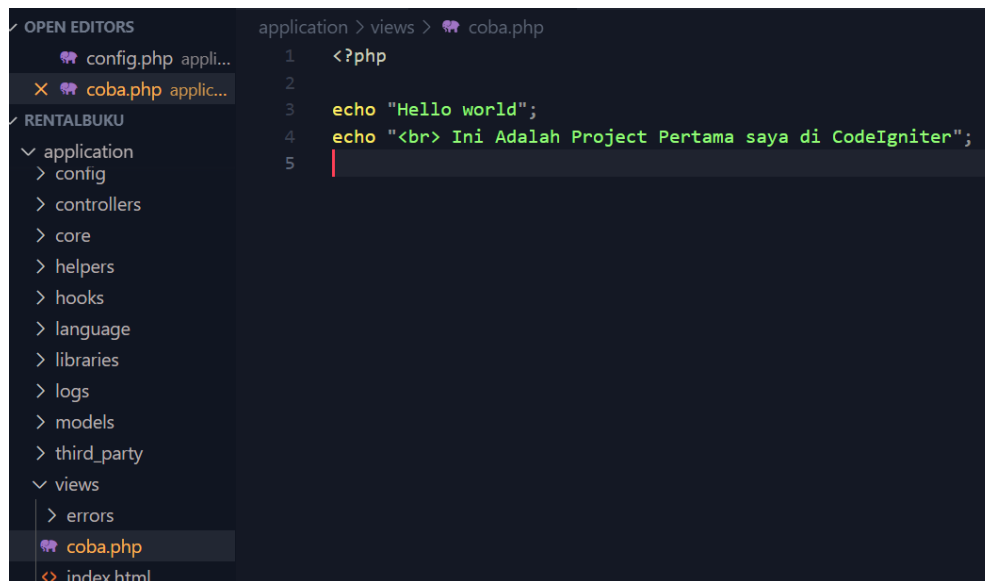
Simpan perubahan dengan menekan tombol **ctrl + s**

Bagian 4 - Struktur MVC di 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..!**

- Buat file baru dengan nama `coba.php` pada folder `view`



- 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 di load kembali pada browser adalah



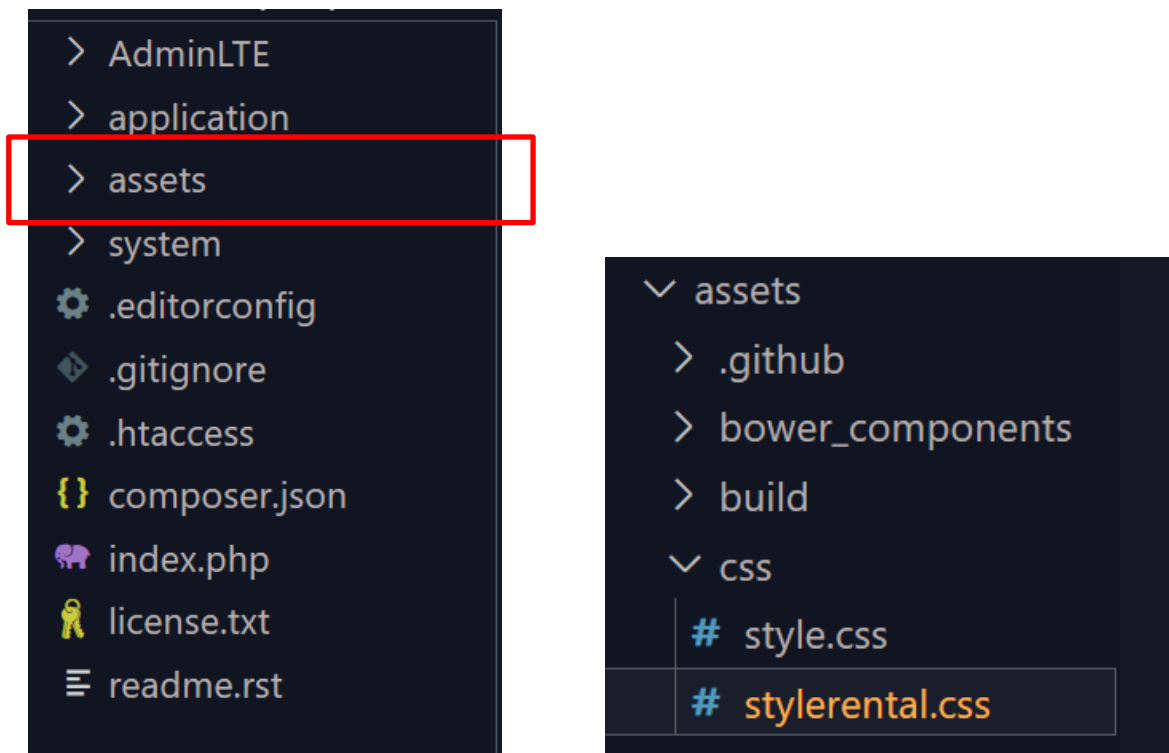
Bagian 5 - Membuat Template Sederhana

A. 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/stylereental.css'); ?&gt;"&gt;
&lt;/head&gt;

&lt;body&gt;
  &lt;div id="wrapper"&gt;
    &lt;header&gt;
      &lt;hgroup&gt;
        &lt;h1&gt;Rental Mobil&lt;/h1&gt;
        &lt;h3&gt;Membuat Template Sederhana dengan CodeIgniter&lt;/h3&gt;
      &lt;/hgroup&gt;
      &lt;nav&gt;
        &lt;ul&gt;
          &lt;li&gt;&lt;a href="<?php echo base_url() . 'buku' ?&gt;"&gt;Home&lt;/a&gt;&lt;/li&gt;
          &lt;li&gt;&lt;a href="<?php echo base_url() . 'buku/about' ?&gt;"&gt;About&lt;/a&gt;&lt;/li&gt;
        &lt;/ul&gt;
      &lt;/nav&gt;
      &lt;div class="clear"&gt;&lt;/div&gt;
    &lt;/header&gt;
    &lt;section&gt;
      &lt;h1&gt;&lt;?php echo $judul ?&gt;&lt;/h1&gt;
      &lt;p align='justify'&gt;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.&lt;/p&gt;
      &lt;p&gt;MVC adalah teknik atau konsep yang memisahkan
        komponen utama menjadi tiga komponen yaitu model, view dan
        controller.&lt;/p&gt;
      &lt;ol type="a"&gt;
        &lt;li&gt;Model&lt;/li&gt;
        &lt;p align='justify'&gt;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.&lt;/p&gt;
        &lt;p align='justify'&gt;Sebagai catatan, Semua model harus disimpan di
          dalam folder application\models&lt;/p&gt;
        &lt;li&gt;View&lt;/li&gt;
        &lt;p align='justify'&gt;View merupakan bagian yang menangani halaman user
</pre

```

```

        interface atau halaman yang muncul pada user(pada browser). Tampilan dari user
        interface di kumpulkan pada view untuk memisahkannya dengan controller dan model
        sehingga memudahkan web designer dalam
        melakukan pengembangan tampilan halaman website.</p>
        <li>Controller</li>
        <p align='justify'>Controller merupakan kumpulan intruksi aksi yang
        menghubungkan model dan view, jadi user tidak akan berhubungan
        dengan model secara langsung, intinya data yang tersimpan di
        database (model) di ambil oleh controller dan kemudian controller pula yang
        menampilkan nya ke view. Jadi controller lah yang mengolah
        intruksi.</p>
        <p align='justify'>Dari penjelasan tentang model view dan controller di atas dapat
        di simpulkan bahwa controller sebagai penghubung view
        dan model. Misalnya pada aplikasi yang menampilkan data dengan
        menggunakan metode konsep mvc, controller memanggil intruksi pada model
        yang mengambil data pada database, kemudian controller yang
        meneruskannya pada view untuk di tampilkan. Jadi jelas sudah dan sangat mudah
        dalam pengembangan aplikasi dengan cara mvc ini karena
        web designer atau front-end developer tidak perlu lagi berhubungan dengan
        controller, dia hanya perlu berhubungan dengan view untuk mendesign tampilann
        aplikasi, karena back-end developer yang
        menangani bagian controller dan modelnya. Jadi pembagian tugas pun menjadi
        mudah dan pengembangan aplikasi dapat di lakukan dengan
        cepat dan terstruktur.</p>
    </section>
    <footer>
        <a href="http://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');
?&gt;"&gt;
</pre

```

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');
?&gt;"&gt;
&lt;/head&gt;
</pre

```

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

Rental Buku

[Home](#) [About](#) [Buku](#)

Membuat Template Sederhana dengan CodeIgniter

Halaman Depan

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 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.

Bagian 6 - 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
);
```

4. Desain tabel user

Tabel user berfungsi untuk menyimpan data pengguna dalam sistem informasi perpustakaan, baik pengguna dengan peran sebagai admin maupun anggota (peminjam buku).

Nama Kolom	Tipe Data	Keterangan
id_user	INT (11)	PRIMARY KEY, AUTO_INCREMENT
nama	VARCHAR(100)	Nama lengkap pengguna
email	VARCHAR(100)	Email pengguna, UNIQUE
password	VARCHAR(255)	Password (hash)
alamat	TEXT	Alamat pengguna
role_id	INT (1)	1 = Admin, 2 = User biasa
is_active	TINYINT (1)	1 = Aktif, 0 = Tidak aktif
tanggal_daftar	DATETIME	Tanggal registrasi

5. Desain tabel Buku

Tabel buku digunakan untuk menyimpan informasi lengkap tentang koleksi buku yang tersedia di perpustakaan. Berikut adalah desain untuk tabel buku:

Kolom	Tipe Data	Keterangan
id_buku	INT (PK, AI)	ID unik buku, primary key dan auto increment
isbn	VARCHAR(20)	Nomor ISBN buku, harus unik
judul	VARCHAR(255)	Judul buku
pengarang	VARCHAR(100)	Nama pengarang buku
penerbit	VARCHAR(100)	Nama penerbit buku
tahun_terbit	YEAR	Tahun terbit buku
kategori	VARCHAR(100)	Kategori/genre buku
jumlah	INT	Jumlah eksemplar buku tersedia
rak	VARCHAR(50)	Lokasi rak buku di perpustakaan
deskripsi	TEXT	Deskripsi atau sinopsis buku
cover	VARCHAR(255)	Nama file gambar cover buku (jika ada)
created_at	TIMESTAMP	Waktu entri buku ke sistem

Bagian 7 - 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. Buatlah Link Baru pada taggar <head> dengan perintah berikut:

```
<li><a href="<?php echo base_url() . 'buku/tambah' ?>">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'); ?>"
enctype="multipart/form-data">
      <tr>
        <td><label for="title">Judul Buku</label></td>
        <td>:</td>
        <td><input type="text" id="title" name="judul" required size="35"></td>
      </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">&nbsp;</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 buatlah 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

Rental Buku

Membuat Template Sederhana dengan CodeIgniter

[Home](#)
[About](#)
[Buku](#)

Data Buku

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

[Tambah Buku](#)

Rental Buku

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:

Penulis:

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->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

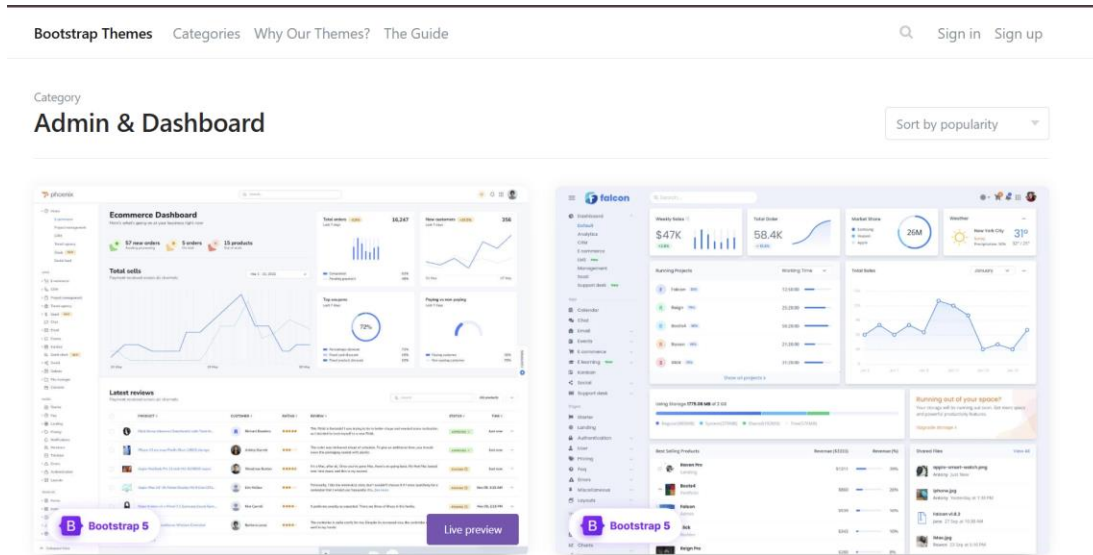
Bagian 8 - Membuat Template dengan menggunakan bootstrap

Bootstrap adalah framework CSS yang sangat populer yang memudahkan pengembangan web responsif dan modern. Dengan menggunakan Bootstrap, kita dapat dengan cepat membuat tata letak yang elegan dan konsisten tanpa harus menulis kode CSS yang rumit. Framework ini menyediakan berbagai komponen seperti grid system, form, button, navigasi, dan banyak lagi, yang dapat kita gunakan untuk mempercepat proses pengembangan.

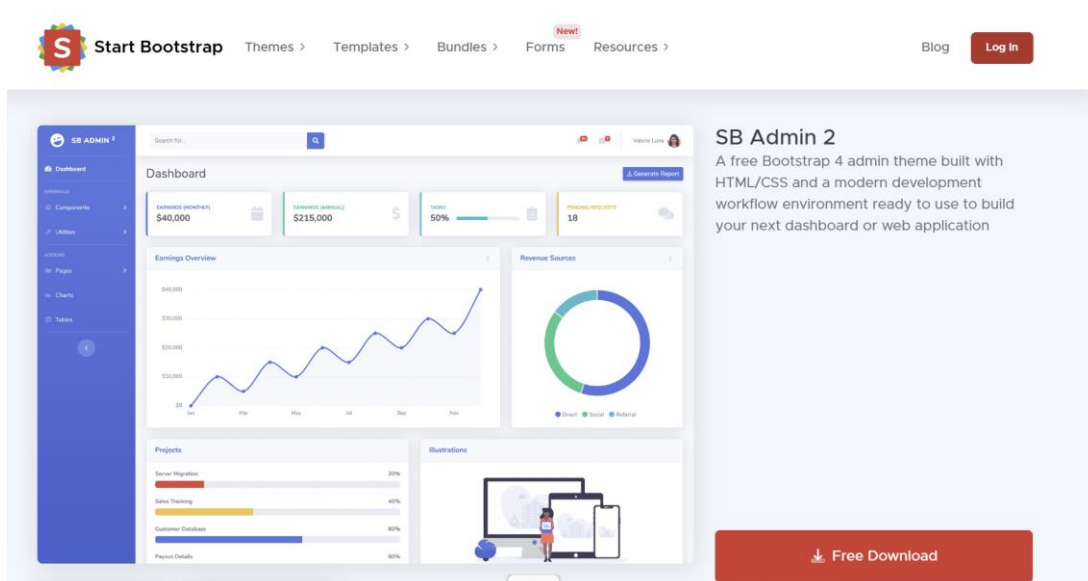


Pada bagian ini, kita akan mempelajari cara membuat template sederhana dengan menggunakan Bootstrap di dalam aplikasi CodeIgniter. Template ini akan mencakup header, footer, dan bagian konten utama yang dapat digunakan kembali di berbagai halaman. Dengan menggunakan template Bootstrap, kita dapat memastikan tampilan yang seragam dan responsif di seluruh aplikasi web kita.

Berikut adalah tampilan halaman bootstrap.



Untuk mengimplementasikan tampilan bootstrap pada website kita, maka terlebih dahulu download tampilan bootstrap yang sesuai dengan kebutuhan kita, misalnya kita akan menggunakan bootstrap admin versi free untuk pembelajaran.

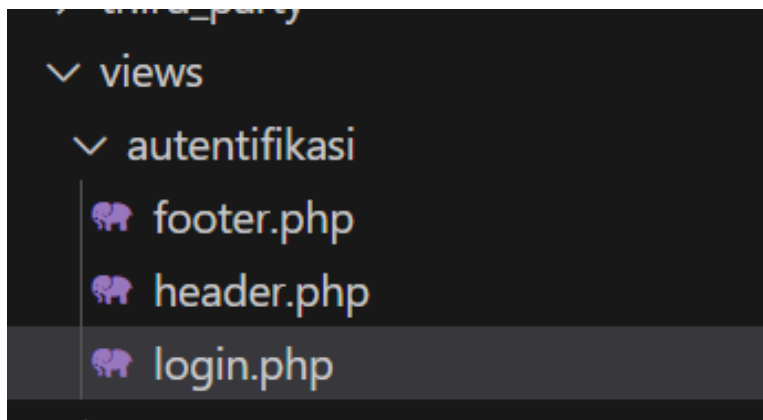


Pilih Button download pada link <https://startbootstrap.com/theme/sb-admin-2> untuk mendapatkan tampilan halaman-halaman website biasanya dalam format .zip

1. Membuat Halaman Login

Untuk Membuat Halaman admin yang sesuai dengan template bootstrap yang sudah di download. Pada project Codeigniter kita buat folder baru dengan nama rental_bootstrap, kemudian kita akan membuat halaman admin sesuai dengan template yang sudah di download dari bootstrap. Maka terlebih dahulu adalah membuat folder pada project rental_bootstrap pada view dengan nama, header.php, footer.php dan **login.php**

Header.php : tampilan yang akan dipanggil untuk halaman admin dan halaman lain yang terkait, jadi pada badan setiap file hanya coding utamanya saja yang di cantumkan.



Isi coding 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>SB Admin 2 - Login</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>

```

Isi coding login.php

```

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

    <div class="container">

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

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

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

                    <!-- Nested Row within Card Body -->
                    <div class="row">

                        <div class="col-lg-6">
                            <div class="p-5">
                                <div class="text-center">
                                    <h1 class="h4 text-gray-900 mb-4">Welcome
Back!</h1>
                                </div>
                                <form class="user">
                                    <div class="form-group">
                                        <input type="email" class="form-control
form-control-user"
                                            id="exampleInputEmail" aria-
describedby="emailHelp"
                                            placeholder="Enter Email Address...">
                                    </div>
                                    <div class="form-group">
                                        <input type="password" class="form-
control form-control-user"
                                            id="exampleInputPassword"
placeholder="Password">

```

```

        </div>
        <div class="form-group">
            <div class="custom-control custom-
checkbox small">
                <input type="checkbox" class="custom-
control-input" id="customCheck">
                <label class="custom-control-label"
for="customCheck">Remember
                    Me</label>
            </div>
        </div>
        <a href="index.html" class="btn btn-primary
btn-user btn-block">
            Login
        </a>
        <hr>
        <a href="index.html" class="btn btn-google
btn-user btn-block">
            <i class="fab fa-google fa-fw"></i> Login
            with Google
        </a>
        <a href="index.html" class="btn btn-facebook
btn-user btn-block">
            <i class="fab fa-facebook-f fa-fw"></i>
            Login with Facebook
        </a>
    </form>
    <hr>
    <div class="text-center">
        <a class="small" href="forgot-
password.html">Forgot Password?</a>
    </div>
    <div class="text-center">
        <a class="small" href="register.html">Create
        an Account!</a>
    </div>
</div>
</div>
</div>
</div>

```

```
</div>
```

Isi 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>

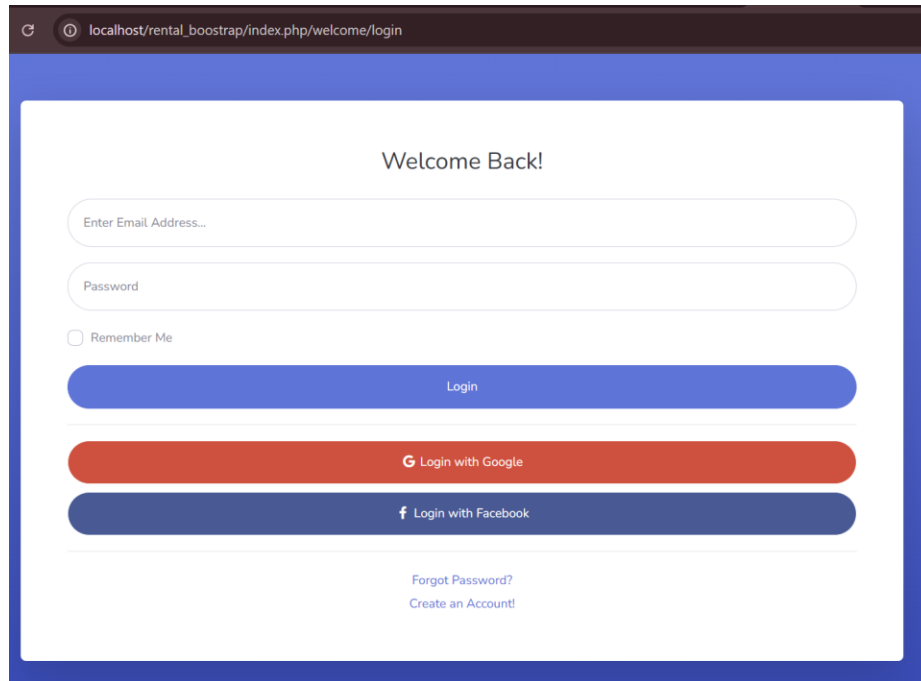
</body>

</html>
```

Pemanggilan di Controller, Tambahkan fungsi login pada controller welcome dan panggil view yang sudah dibuat sebagai berikut.

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

Hasilnya ketika dipanggil di browser akan menjadi seperti ini



2. Membuat Halaman Register

Untuk membuat halaman register kita hanya perlu, membuat view register di folder autentifikasi. Berikut codingnya

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

  <div class="container">

    <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-5 d-none d-lg-block bg-register-
image"></div>
          <div class="col-lg-7">

            <div class="text-center">
              <h1 class="h4 text-gray-900 mb-4">Create an
Account!</h1>
            </div>
            <form class="user">
              <div class="form-group row">
                <div class="col-sm-6 mb-3 mb-sm-0">
                  <input type="text" class="form-
control form-control-user" id="exampleFirstName">

```

```

placeholder="First Name">
</div>
<div class="col-sm-6">
    <input type="text" class="form-
control form-control-user" id="exampleLastName"
placeholder="Last Name">
</div>
</div>
<div class="form-group">
    <input type="email" class="form-control
form-control-user" id="exampleInputEmail"
placeholder="Email Address">
</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="exampleInputPassword"
placeholder="Password">
</div>
<div class="col-sm-6">
        <input type="password" class="form-
control form-control-user"
id="exampleRepeatPassword"
placeholder="Repeat Password">
</div>
</div>
<a href="login.html" class="btn btn-primary
btn-user btn-block">
    Register Account
</a>
<hr>
<a href="index.html" class="btn btn-google
btn-user btn-block">
    <i class="fab fa-google fa-fw"></i>
    Register with Google
</a>
<a href="index.html" class="btn btn-
facebook btn-user btn-block">
    <i class="fab fa-facebook-f fa-fw"></i>
    Register with Facebook
</a>
</form>
<hr>
<div class="text-center">

```

```

        <a class="small" href="forgot-
password.html">Forgot Password?</a>
    </div>
    <div class="text-center">
        <a class="small" href="login.html">Already
have an account? Login!</a>
    </div>

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

```

Kemudian di controller welcome tinggal menambahkan fungsi register sebagai berikut :

```

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

```

Hasil tampilan ketika di load di browser akan tampil halaman berikut:

The screenshot shows a web browser window with the URL `localhost/rental_bootstrap/index.php/welcome/register`. The page content is a registration form with the title "Buat Sebuah Akun!". The form contains the following elements:

- Input fields for "First Name" and "Last Name".
- An input field for "Email Address".
- Input fields for "Password" and "Repeat Password".
- A blue button labeled "Register Account".
- A red button labeled "G Register with Google".
- A dark blue button labeled "f Register with Facebook".
- At the bottom, there are two links: "Forgot Password?" and "Already have an account? Login!".

Dengan menerapkan konsep MVC ini kita hanya akan membuat satu page controller yang bisa digunakan untuk memanggil beberapa tampilan/view. Begitupun saat membuat view kita hanya perlu membuat satu view yang digunakan untuk beberapa tampilan atau halaman, sehingga dengan konsep MVC dengan framework ini membuat sebuah system informasi atau program menjadi lebih efektif dan efisien.

TUGAS...!!!

Setelah mempelajari konsep MVC maka tugas selanjutnya adalah membuat tampilan halaman beranda dengan menggunakan template bootstrap dengan mengikuti konsep pemrograman MVC.

DAFTAR PUSTAKA

1. Supomo, Putratama, 2018. “Pemograman Web Dengan Menggunakan PHP dan Framework Codeigniter”, Jakarta: Deepublish Publisher.
2. <https://codeigniter.com/en/docs3>.
3. Kadir, 2011. “Buku Pintar JQuery Dan Php Untuk Pemula”, Jakarta: bukuseru
4. Raharjo, 2018. “Belajar Otodidak Framework CodeIgniter Teknik Pemrograman Web dengan PHP dan Framework CodeIgniter”, Bandung: Informatika Bandung
5. Badiano, Yosef Murya, 2018. “Project PHP Membangun Sistem Informasi Akademik dengan Framework Codeigniter”, Jakarta: Langit Inspirasi
6. Basuki, Awan Pribadi, 2014. “Proyek Membangun Website Berbasis PHP dengan Codeigniter”, Jakarta: Loko Media.
7. Belajar Otodidak Framework CodeIgniter Teknik Pemrograman Web dengan PHP dan Framework CodeIgniter 3, Informatika Bandung.