

WEB PROGRAMMING I



KATA PENGANTAR

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

Pada web Programming I materi bahasan meliputi pengenalan dasar web programming, perintah HTML, PHP, CSS, Javascript sampai dengan pengenalan web menggunakan framework codeigniter. Diakhir perkuliahan mahasiswa diharapkan mampu mengimplementasikannya dalam bentuk final proect yang harus dipresentasikan sebagai syarat kelulusan mata kuliah Web Programming I. Teknik penyajiannya dilakukan secara terpadu dan sistematis.

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

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

Jakarta, Desember 2023

Maruloh, M.Kom

DAFTAR ISI

Halaman

KATA PENGANTAR.....	2
DAFTAR ISI	3
MINGGU KE 1 Konsep Dasar Web dan Pengenalan HTML	4
MINGGU KE 2 Pengenalan PHP, Variabel, Konstanta, Tipe Data, Komentar dan Operator .	23
MINGGU KE 3 Pengenalan Form, Percabangan, dan Perulangan.....	35
MINGGU KE 4 CSS dan Javascript	52
MINGGU KE 5 Membuat Template Web.....	64
MINGGU KE 6 Pengenalan Codeigniter	71
MINGGU KE 7 Model View Controller	80
MINGGU KE 9 Membuat Template Web Sederhana	92
MINGGU KE 10 Membuat Form Validasi.....	101
MINGGU KE 11 Membuat Program Create Read Update Delete.....	106
MINGGU KE 12 Membuat Program Create Read Update Delete.....	114
MINGGU KE 13-15 Presentasi Project	119

MINGGU KE 1

Konsep Dasar Web dan Pengenalan HTML

Deskripsi:

Membahas mengenai pengenalan Web Programming dan perintah dasar membuat Web dengan menggunakan perintah HTML. Pertemuan ini juga membahas mengenai perintah Tag-tag HTML yang biasa digunakan dalam web

Tujuan Pembelajaran:

Setelah melakukan bagian ini mahasiswa mampu:

1. konsep dasar pemrograman Web
2. memahami konsep dasar HTML
3. menggunakan perintah HTML dalam pemrograman

A. 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 mengidentifikasi 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* sertain, 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
- b. Internet Information Service (IIS), hanya dapat berjalan di sistem operasi MS Windows

B. Struktur Navigasi

Struktur Navigasi adalah “Susunan menu atau hirarki dari suatu situs yang menggambarkan isi dari setiap halaman dan *link* atau navigasi tiap halaman pada suatu situs *web*”. Struktur Navigasi dapat dikatakan sebagai penggambar dari hubungan atau rantai kerja dari seluruh elemen yang akan digunakan dalam aplikasi.

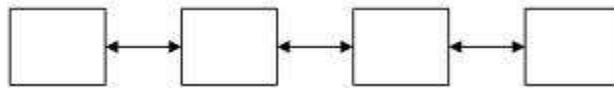
Struktur Navigasi dapat digolongkan menurut kebutuhan akan objek, kemudahan pemakaian, keinteraktifitasannya, dan kemudahan membuatnya yang berpengaruh terhadap waktu pembuatan suatu situs *web*. Dalam penggambarannya Struktur Navigasi

terbagi kedalam 4 Struktur yang berbeda yaitu: *Linier*, *Non Linier*, *Hierarchical* (Hirarki) dan *Composit* (Campuran).

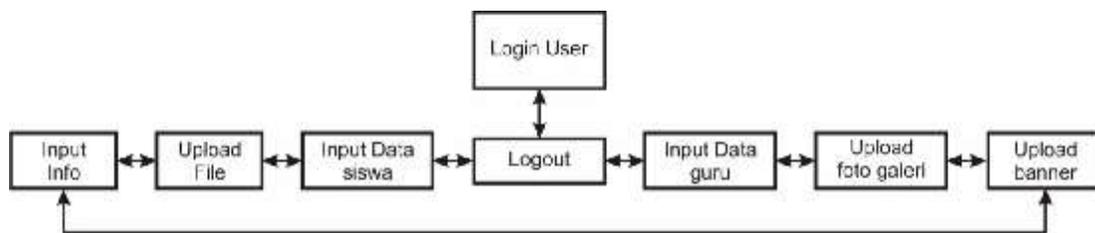
Ada 4 macam bentuk dasar dari peta navigasi yang biasa digunakan dalam proses pembuatan aplikasi web, yaitu:

1. Struktur Navigasi Linier

Struktur navigasi linier hanya mempunyai satu rangkaian cerita yang berurut, yang menampilkan satu demi satu tampilan layar secara berurut menurut urutannya. Tampilan yang dapat ditampilkan pada struktur jenis ini adalah satu halaman sebelumnya atau satu halaman sesudahnya, tidak dapat dua halaman sebelumnya atau dua halaman sesudahnya.

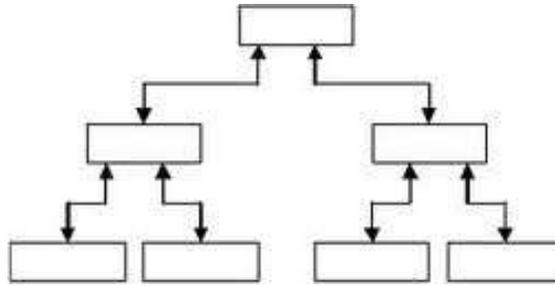


Contoh :

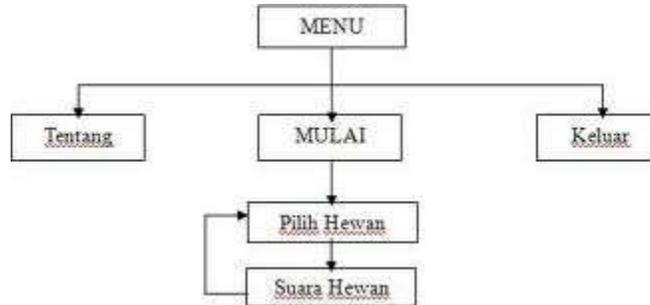


2. Struktur Navigasi Hirarki

Struktur navigasi hirarki biasa disebut struktur bercabang, merupakan suatu struktur yang mengandalkan percabangan untuk menampilkan data berdasarkan kriteria tertentu. Tampilan pada menu pertama akan disebut sebagai *Master Page* (halaman utama pertama), halaman utama ini mempunyai halaman percabangan yang disebut *Slave Page* (halaman pendukung). Jika salah satu halaman pendukung dipilih atau diaktifkan, maka tampilan tersebut akan bernama *Master Page* (halaman utama kedua), dan seterusnya. Pada struktur navigasi ini tidak diperkenankan adanya tampilan secara linier.

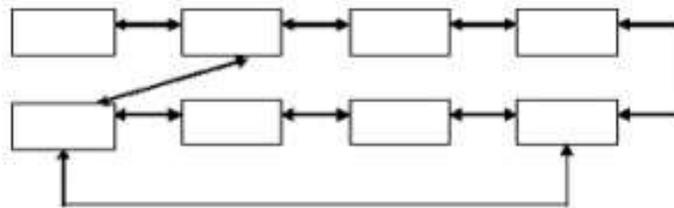


Contoh :

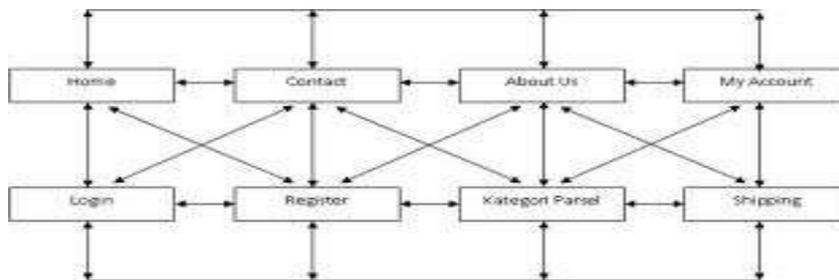


3. Struktur Navigasi Non-Linier

Struktur navigasi non-linier atau struktur tidak berurut merupakan pengembangan dari struktur navigasi linier. Pada struktur ini diperkenankan membuat navigasi bercabang. Percabangan yang dibuat pada struktur nonlinier ini berbeda dengan percabangan pada struktur hirarki, karena pada percabangan nonlinier ini walaupun terdapat percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama yaitu tidak ada *Master Page* dan *Slave Page*.

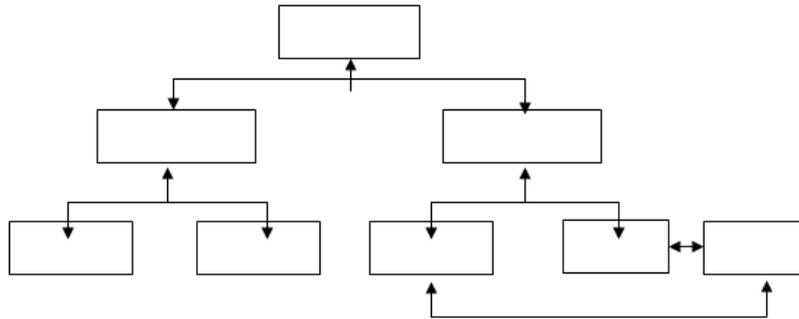


Contoh :

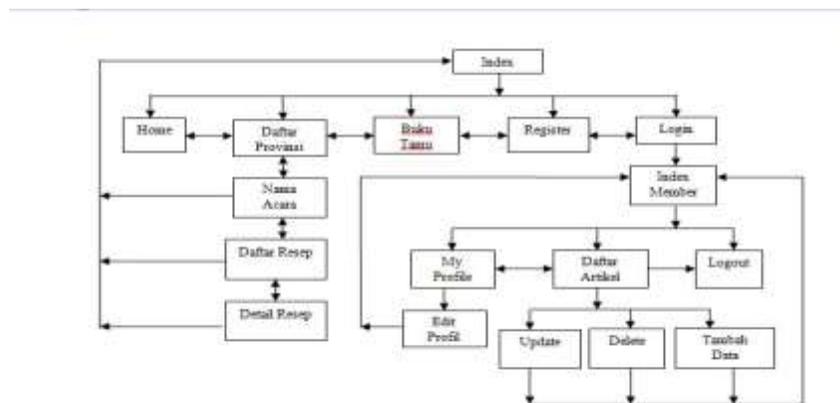


4. Struktur Navigasi Campuran

Struktur navigasi campuran merupakan gabungan dari ketiga struktur sebelumnya yaitu linier, non-linier dan hirarki. Struktur navigasi ini juga biasa disebut dengan struktur navigasi bebas. Struktur navigasi ini banyak digunakan dalam pembuatan website karena struktur ini dapat digunakan dalam pembuatan website sehingga dapat memberikan ke-interaksian yang lebih tinggi.



Contoh :



C. Pengertian HTML (Hypertext Markup Language)

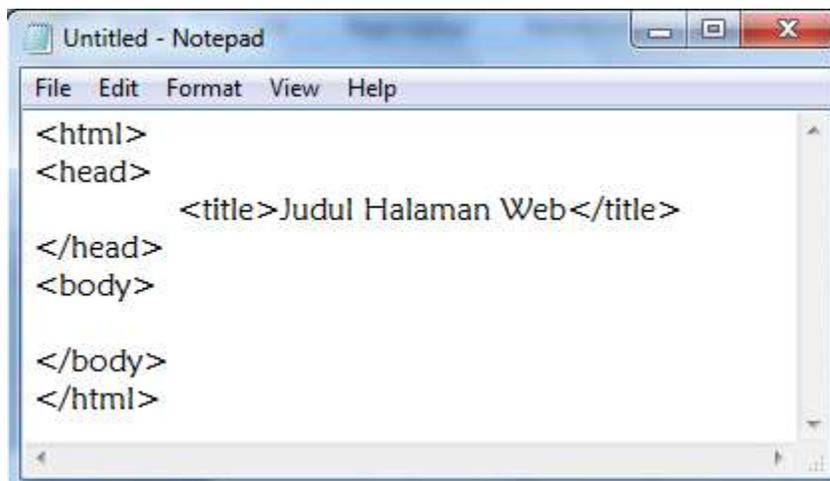
Hypertext Markup Language (HTML) adalah sebuah bahasa untuk menampilkan konten di web. HTML sendiri adalah bahasa pemrograman yang bebas, artinya tidak dimiliki oleh siapapun, pengembangannya dilakukan oleh banyak orang di banyak Negara dan bias dikatakan sebagai sebuah bahasa yang dikembangkan bersama-sama 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 Notepad dan XAMPP Versi 1.8.1 untuk web servernya dengan bahasa pemrograman PHP Versi 5.

D. 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 HTML standar terlihat seperti ini :



```
Untitled - Notepad
File Edit Format View Help
<html>
<head>
    <title>Judul Halaman Web</title>
</head>
<body>

</body>
</html>
```

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 Notepad, kemudian ketikkan perintah di bawah ini. Simpan dengan nama **Contoh01.php**

```

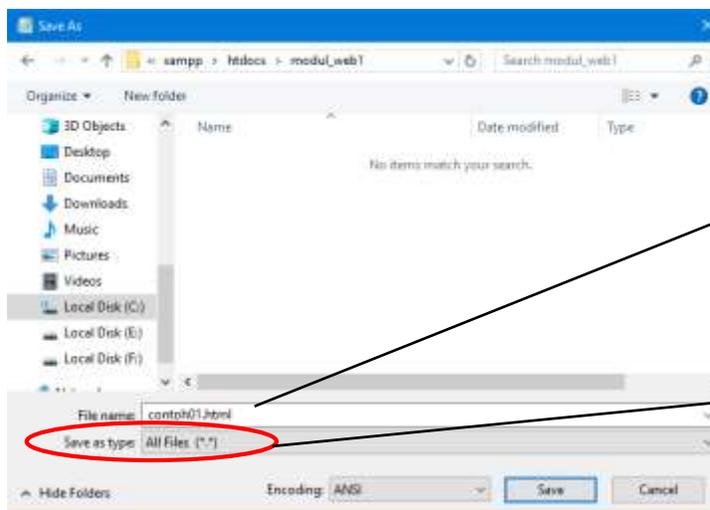
<html>
<head>
    <title>Contoh 01 </title>
</head>
<body>
    Halo... <br>
    ini script HTML pertamaku
</body>
</html>

```

→ Judul Web
} Perintah dibuat di antara <body> dan </body>

Kemudian simpan file di atas di dalam folder `c:\XAMPP\htdocs\` buat folder baru untuk menyimpan file di dalam folder `htdocs`. Simpan file dengan nama `contoh01.html`. Pembuatan nama file pada saat penyimpanan harus diakhiri dengan extention “.html”

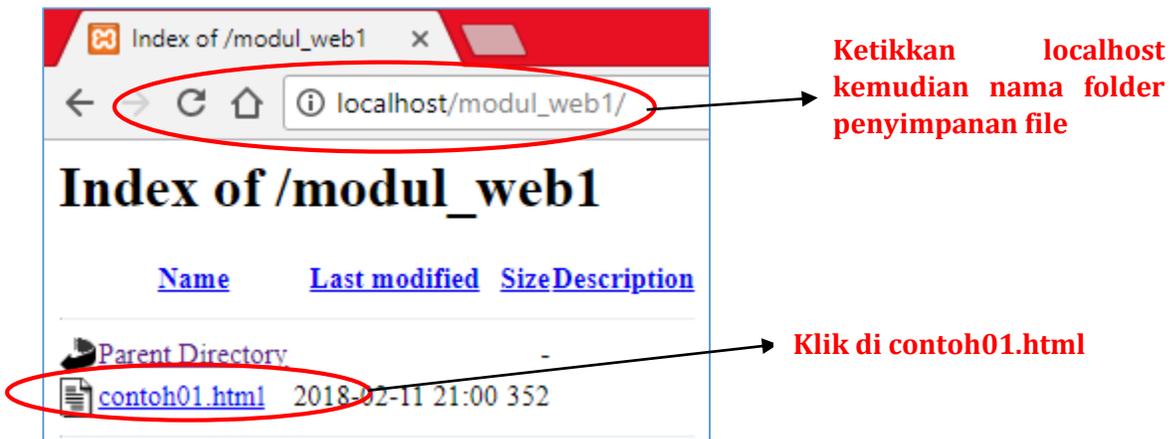
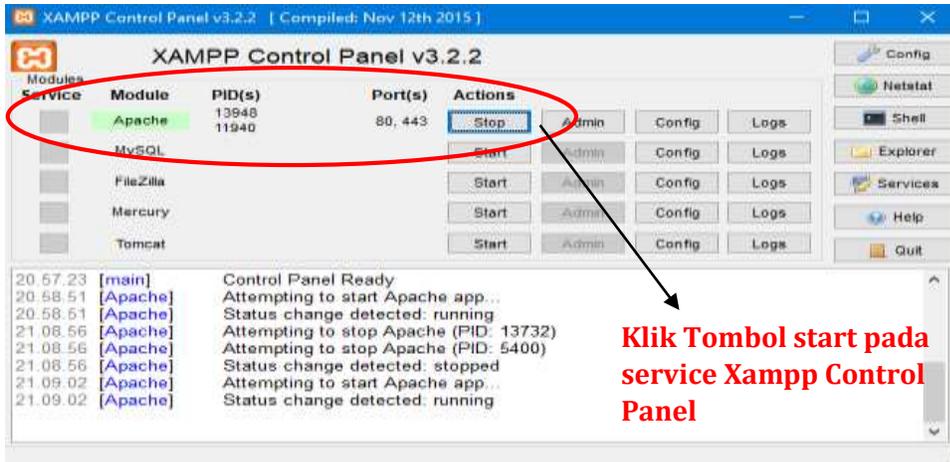
Cara penyimpanan dengan Notepad, perhatikan cara berikut :



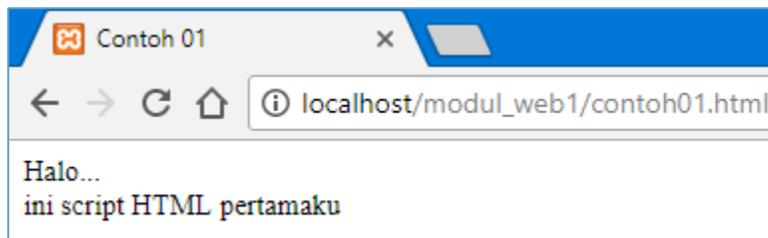
Sertakan extension .html pada nama filenya contoh : contoh01.html

Pastikan memilih All Files pada Save as Type

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 contoh01.html Sebelum di ketikkan alamat file tersebut, pastikan anda telah menjalankan Module Apache pada Xampp Control Panel. Lihat gambar di bawah ini :



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 garis miring (/). Biasanya Tag dituliskan secara berpasangan, misalnya <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 :

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

- c. Tag untuk garis datar yaitu <hr>
- d. 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 Cititation, 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 Notepad, kemudian ketikkan perintah di bawah ini. Simpan dengan nama **Contoh02.html**

```
<html>
<head>
  <title>Contoh 02</title>
</head>
<body bgcolor="#00CCFF" text="#FF0000">
<p> Ini adalah contoh penggunaan formatting TAG dalam HTML <br />
  masing-masing TAG memiliki atribut masing-masing<br />
</p>
<font color="#000000">Ini juga termasuk contoh penggunaan
formatting TAG<br /></font>
<h1><marquee width="50%" bgcolor="#000099">
Ini juga salah satu penggunaan Tag</marquee></h1>
</body>
</html>
```

Simpan dengan nama contoh02.html

Hasil Tampilan



E. 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

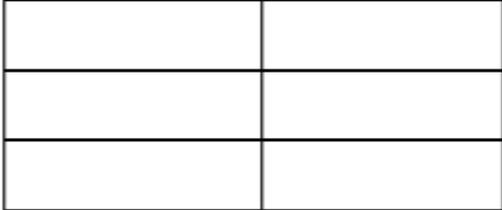
cara yang bisa kita lakukan adalah dengan menggabungkan beberapa sel menjadi satu. Cara ini disebut merge atau penggabungan sel.

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)

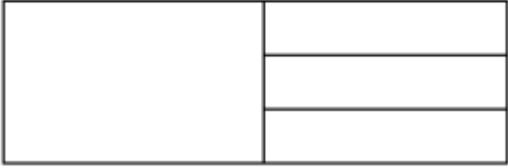
Tabel dengan kode HTML dibawah ini sel-sel kolom pertama akan digabung:

```
<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 Horisontal (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 **Contoh03.html**

```

<html>
<head>
  <title>Contoh 03</title>
</head>
<body>
<h1>Tabel Data Siswa</h1>
<table border=1 bgcolor=pink>
  <tr>
    <th>Nim</th>
    <th>Nama</th>
    <th>Alamat</th>
    <th>Tempat, Tanggal Lahir</th>
    <th>Jurusan</th>
  </tr>
  <tr>
    <td>12110001</td>
    <td>Anita</td>
    <td>Cengkareng</td>
    <td>Jakarta, 20 Agustus 1990</td>
    <td>Teknik Informatika</td>
  </tr>
  <tr>
    <td>12110002</td>
    <td>Aditya</td>
    <td>Tangerang</td>
    <td>Semarang, 01 Januari 1989</td>
    <td>Teknik Informatika</td>
  </tr>
  <tr>
    <td>12110003</td>
    <td>Firman</td>
    <td>Bogor</td>
    <td>Jakarta, 18 September 1988</td>
    <td>Teknik Informatika</td>
  </tr>
</body>
</html>

```

Jika dilihat di browser, maka terlihat sebagai berikut :

Tabel Data Siswa

Nim	Nama	Alamat	Tempat, Tanggal Lahir	Jurusan
12110001	Anita	Cengkareng	Jakarta, 20 Agustus 1990	Tehnik Informatika
12110002	Aditya	Tangerang	Semarang, 01 Januari 1989	Tehnik Informatika
12110003	Firman	Bogor	Jakarta, 18 September 1988	Tehnik Informatika

Penggunaan Cellpadding dan cellspacing

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

```
<!DOCTYPE html>
<html>
<head>
<title>Penggunaan atribut Cellpadding dan cellspacing dalam Tabel</title>
</head>
<body>
<h3>Belajar atribut cellpadding & cellspacing dalam Tabel</h3>
<table border="1" cellspacing="0" cellpadding="0">
  <tr>
    <td>Baris 1, Kolom 1</td>
    <td>Baris 1, Kolom 2</td>
    <td>Baris 1, Kolom 3</td>
  </tr>
  <tr>
    <td>Baris 2, Kolom 1</td>
    <td>Baris 2, Kolom 2</td>
    <td>Baris 2, Kolom 3</td>
  </tr>
</table>
<br />
```

```
<table border="1" cellspacing="3" cellpadding="5">
  <tr>
    <td>Baris 1, Kolom 1</td>
    <td>Baris 1, Kolom 2</td>
    <td>Baris 1, Kolom 3</td>
  </tr>
  <tr>
    <td>Baris 2, Kolom 1</td>
    <td>Baris 2, Kolom 2</td>
    <td>Baris 2, Kolom 3</td>
  </tr>
</table>
</body>
</html>
```

Hasil di browser :

Belajar atribut cellpadding & cellspacing dalam Tabel

Baris 1, Kolom 1	Baris 1, Kolom 2	Baris 1, Kolom 3
Baris 2, Kolom 1	Baris 2, Kolom 2	Baris 2, Kolom 3

Baris 1, Kolom 1	Baris 1, Kolom 2	Baris 1, Kolom 3
Baris 2, Kolom 1	Baris 2, Kolom 2	Baris 2, Kolom 3

Penggunaan Rowspan dan colspan

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

```
<html>
<head>
<title>Contoh Penggunaan Atribut Colspan dan Rowspan Tag Tabel</title>
</head>
<body>
<h1>Contoh atribut colspan dan rowspan </h1>
<table border="1">
  <tr>
    <td>Baris 1, Kolom 1</td>
    <td>Baris 1, Kolom 2</td>
    <td>Baris 1, Kolom 3</td>
  </tr>
  <tr>
    <td>Baris 2, Kolom 1</td>
    <td colspan="2">Baris 2, Kolom 2 & 3</td>
  </tr>
  <tr>
    <td rowspan="2"> Baris 3 & 4, Kolom 1</td>
    <td> Baris 3, Kolom 2</td>
    <td> Baris 3, Kolom 3</td>
  </tr>
  <tr>
    <td> Baris 4, Kolom 2</td>
    <td> Baris 4, Kolom 3</td>
  </tr>
</table>
</body>
</html>
```

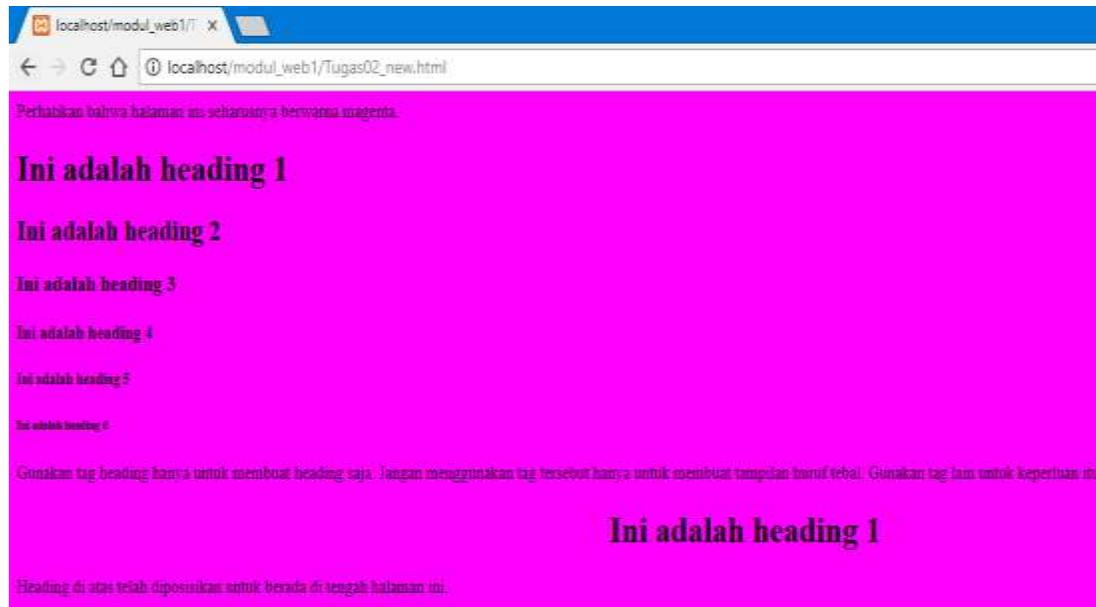
Hasil di browser :

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	<ul style="list-style-type: none">• Kamera mirrorless• Efektifitas Piksel : 18 MP• Tipe sensor : CMOS DIGIC 6• Layar : 3.0 Inch• Memiliki lampu flash dengan jarak jangkauan hingga 5 meter

MINGGU KE 2

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 kelompokindependen 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.

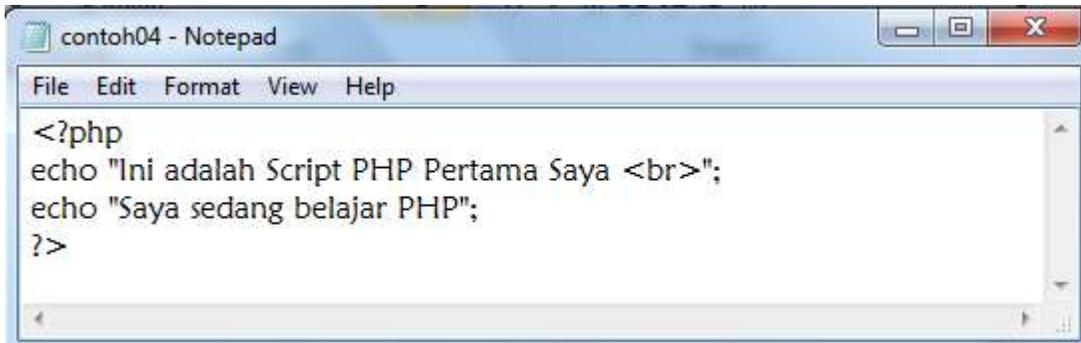
Dengan adanya kode PHP, sebuah halaman web bisa melakukan banyak hal yang dinamis, seperti mengakses database, membuat gambar, membaca dan menulis file, dan sebagainya. Hasil akhir pengolahan kode PHP akan dikembalikan lagi dalam bentuk kode HTML untuk ditampilkan di browser. Ada 4 jenis tag yang bisa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

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

Yang dapat langsung diterapkan disemua platform adalah tag standard dan tag script. Di dalam modul ini bahasa pemrograman yang digunakan adalah PHP Versi 5 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 :



```
<?php
echo "Ini adalah Script PHP Pertama Saya <br>";
echo "Saya sedang belajar PHP";
?>
```

Simpan file dengan nama **contoh04.php**

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

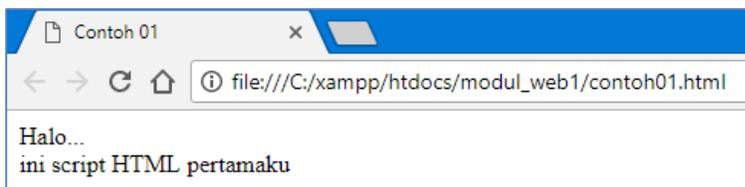
Pilih file contoh04.php maka akan tampil hasilnya :

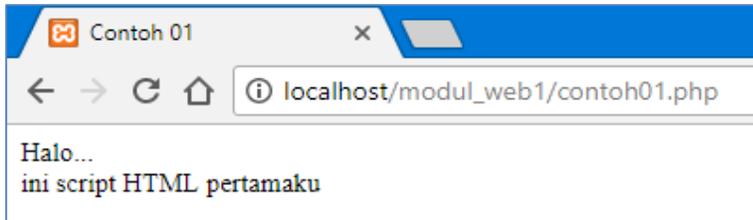


Contoh04.php merupakan contoh script php yang berdiri sendiri tanpa ada tambahan script yang lain. Perintah **echo** merupakan perintah yang digunakan untuk mencetak. Script PHP bisa juga digabung dalam tag HTML.

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

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.

Contoh05.php:

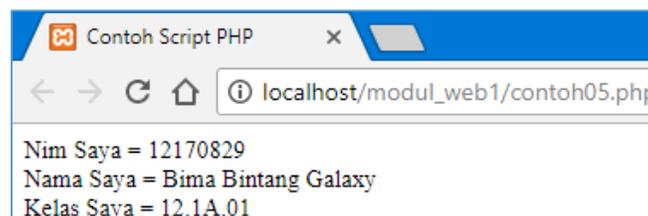
```
<html>
<head>
    <title> Contoh Script PHP</title>
</head>
<?php

    $nim ="12170829";
    $nama ="Bima Bintang Galaxy";
    $kelas ="12.1A.01";

    echo "Nim Saya = $nim<br>";
    echo "Nama Saya = $nama<br>";
    echo "Kelas Saya = $kelas<br>";

?>
</body>
</html>
```

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);

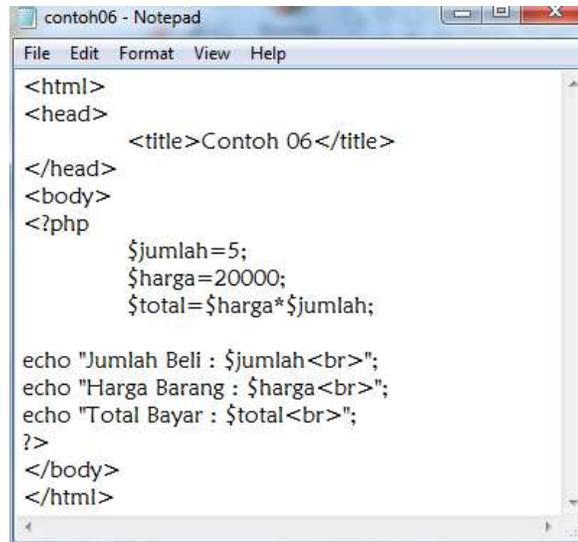
Anda juga bisa mengubah jenis variable tertentu dengan perintah :

(jenis_variabel) \$nama_variabel;

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

\$var_string = (string) \$angka;

Contoh06.php:



```
contoh06 - Notepad
File Edit Format View Help
<html>
<head>
  <title>Contoh 06</title>
</head>
<body>
<?php
  $jumlah=5;
  $harga=20000;
  $total=$harga*$jumlah;

  echo "Jumlah Beli : $jumlah<br>";
  echo "Harga Barang : $harga<br>";
  echo "Total Bayar : $total<br>";
?>
</body>
</html>
```

Hasil Tampilan :



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.

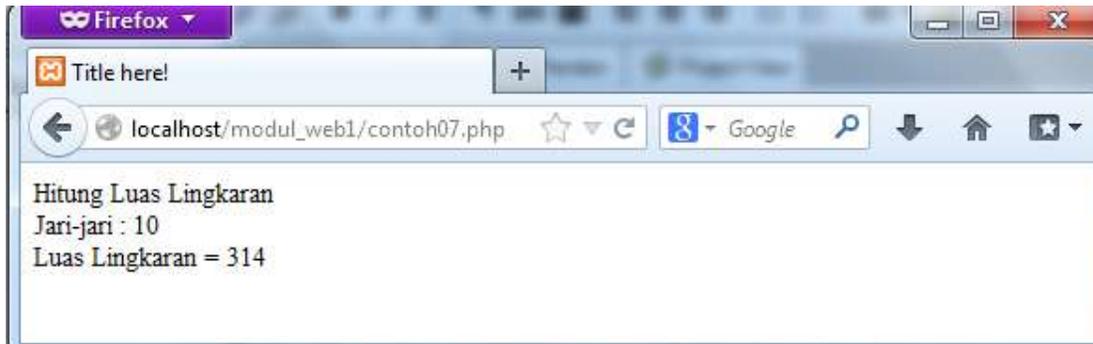
Contoh07.php

```
<html>
<head>
  <title> Menghitung Luas Lingkaran</title>
</head>
<body>
<?php
//konstanta untuk nilai judul
define("Judul","Hitung Luas Lingkaran");
//konstanta untuk nilai phi
define("PHI",3.14);
echo Judul;
$r = 10;
echo "<br> Jari-jari : $r <br>";
$luas = PHI*$r*$r;
echo "Luas Lingkaran = $luas";
?>
</body>
</html>
```

Deklarasi komentar

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:

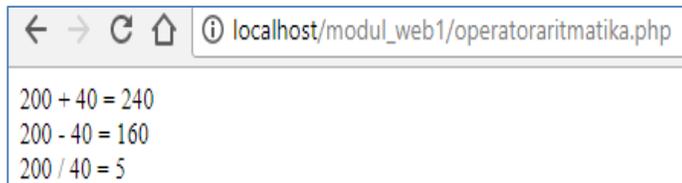
Operator	Nama	Contoh	Hasil
+	Penambahan	1+4	5
-	Pengurangan	1-4	-3
/	Pembagian	1/4	0.25
*	Perkalian	1*4	4
%	Sisa Pembagian	5%2	1
++	Inkremen	X=5; X++	X=6
-	Dekremen	X=5; X-	X=4

Contoh script :

Operatoraritmatika.php

```
operatoraritmatika - Notepad
File Edit Format View Help
<?php
$bil1=200;
$bil2=40;
$hasil = $bil1+$bil2;
echo "$bil1 + $bil2 = $hasil<br>";
$hasil = $bil1-$bil2;
echo "$bil1 - $bil2 = $hasil<br>";
$hasil = $bil1*$bil2;
echo "$bil1 * $bil2 = $hasil<br>";
$hasil = $bil1/$bil2;
echo "$bil1 / $bil2 = $hasil<br>";
?>
```

Hasil :



b) Operator Perbandingan

Operator perbandingan digunakan untuk menghasilkan 2 nilai yang hasil akhirnya adalah nilai Boolean true dan false. Operator ini sangat berguna dalam pemrograman karena bisa menentukan arah pemrograman. Operator perbandingan di PHP adalah :

Operator	Nama	Contoh	Hasil
==	Sama dengan	6 == 6	False
!=	Tidak sama dengan	3 != 3	False
>	Lebih besar	1 > 5	False
>=	Lebih besar atau sama dengan	3 >= 4	False
<	Lebih kecil	2 < 4	True
<=	Lebih kecil atau sams dengan	5 <= 4	False

Opertorperbandingan.php

```
operatorperbandingan - Notepad
File Edit Format View Help
<?php
$bil1=200;
$bil2=40;
$teks1="PHP";
$teks2="php";

$hasil=$bil1==$bil2;
echo "$bil1 == $bil2 = $hasil<br>";

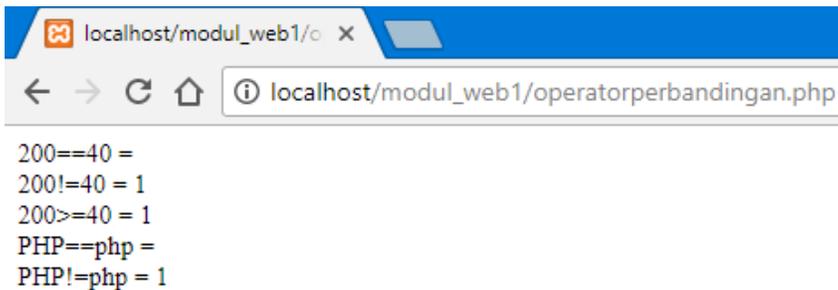
$hasil=$bil1!=$bil2;
echo "$bil1 != $bil2 = $hasil<br>";

$hasil=$bil1>=$bil2;
echo "$bil1 >= $bil2 = $hasil<br>";

$hasil=$teks1==$tek2;
echo "$teks1 == $teks2 = $hasil<br>";

$hasil=$teks1!=$tek2;
echo "$teks1 != $teks2 = $hasil<br>";
?>
```

Hasil :



```
localhost/modul_web1/c x
localhost/modul_web1/operatorperbandingan.php
200==40 = 0
200!=40 = 1
200>=40 = 1
PHP==php = 0
PHP!=php = 1
```

c) Operator Logika

Operator untuk menyusun kalimat ekspresi/ungkapan logika. Hasil operasi ini akan didapatkan nilai satu jika benar dan nol jika salah.

Operator	Fungsi
AND atau &&	Operasi logika AND
OR atau	Operasi logika OR
XOR	Operasi logika eksklusife OR
!	Ingkaran/negasi

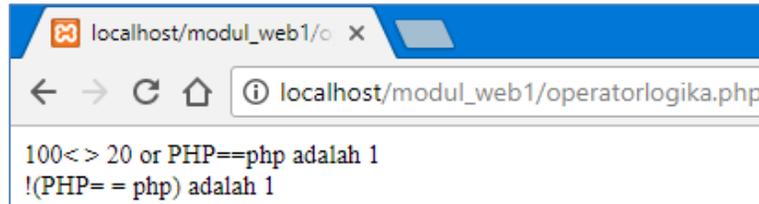
Operatorlogika.php

```
operatorlogika - Notepad
File Edit Format View Help
<?php
$bill=100;
$bil2=20;
$teks1="PHP";
$teks2="php";

$hasil=($bill<>$bil2) or ($teks1==teks2);
echo "$bill <> $bil2 or $teks1==teks2 adalah $hasil<br>";

$hasil=!($teks1==$teks2);
echo "!($teks1==teks2) adalah $hasil";
?>
```

Hasilnya :



```
localhost/modul_web1/o x
localhost/modul_web1/operatorlogika.php
100<> 20 or PHP==php adalah 1
!(PHP= = php) adalah 1
```

d) Operator String

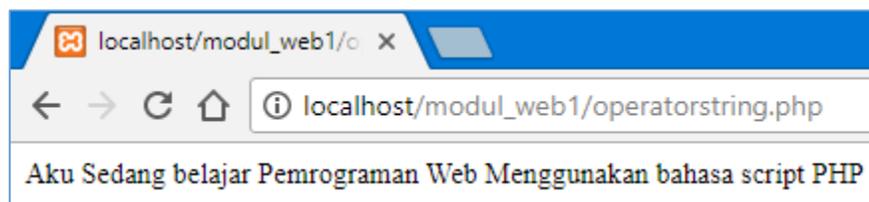
Dalam PHP juga tersedia operator string, yaitu digunakan untuk operasi penggabungan teks. Adapun symbol yang digunakan yaitu berupa karakter titik (.)

Operatorstring.php

```
<?php
$teks1="Aku Sedang belajar";
$teks2="Pemrograman Web";
$teks3="Menggunakan bahasa script PHP";
$hasil=$teks1 . $teks2 . $teks3;

echo "$hasil ";
?>
```

Hasilnya :



```
localhost/modul_web1/o x
localhost/modul_web1/operatorstring.php
Aku Sedang belajar Pemrograman Web Menggunakan bahasa script PHP
```

Tugas 02 :

1. Buat script php untuk tampilan di bawah ini. Tentukan variabelnya :



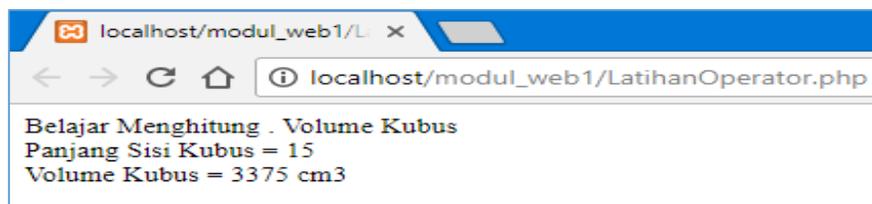
2. Buat script menggunakan bahasa pemrograman PHP untuk menghitung volume Kubus menggunakan fungsi operator aritmatika dan operator string dengan ketentuan sebagai berikut :

Panjang sisi kubus = 15cm

Hitung volume balok dengan rumus = sisi x sisi x sisi

Buat variable teks1 yang berisi = "Belajar Menghitung" dan teks2 yang berisi = "Volume Kubus". Buat perintah untuk menggabungkan nilai dari variable teks1 dan teks2 menggunakan operator string.

Outputnya sebagai berikut :



MINGGU KE 3

Pengenalan Form, Percabangan, dan Perulangan

Deskripsi:

Pertemuan ini membahas mengenai pembuatan form dalam web serta metode pengiriman nilai pada form. Dibahas juga mengenai perintah percabangan (IF dan Switch case) serta perintah perulangan (looping)

Tujuan Pembelajaran:

Setelah melakukan bagian ini mahasiswa mampu:

1. memahami dan menggunakan Form
2. memahami dan menggunakan perintah percabangan dalam PHP
3. memahami dan menggunakan perintah Perulangan dalam PHP

A. Komponen Form

Sebuah *website* dinamis seringkali memerlukan interaksi antara *browser client* dan *server* bisa berupa pemasukan data teks, angka, atau *upload file* untuk diproses oleh *server*. Untuk memudahkan suatu data yang dikirimkan oleh *browser client*, dibutuhkan adanya FORM HTML. Penggunaan *form* misalnya untuk pendaftaran keanggotaan, pemasukan kode kartu kredit, *login user*, transaksi perbelanjaan, dan *upload file*.

Dalam FORM HTML terdapat beberapa komponen yang bisa digunakan, antara lain :

1) Form

```
<FORM ACTION=action METHOD=method ENCTYPE=media type> </FORM>
```

2) Text Box

Text box : untuk menginput data string ataupun angka.

```
< INPUT TYPE=TEXT NAME="nama_variabel" VALUE="value">
```

3) Text Area

Text area : untuk menginput string ataupun angka yang terdiri atas banyak baris.

```
<textarea rows=" " cols=" " name="nama_variabel"> </textarea>
```

4) **Radio buton**

Laki-Laki Perempuan

Radio buton : untuk memilih satu pernyataan dari beberapa pernyataan yang disediakan.

```
<input type="radio" name="nama_variabel" value=" " >Isi_Radio
```

5) **Combo Box**

Combo box untuk menampilkan daftar data.

```
<select name="nama_variabel" value=" ">  
<option>Combo1</option>  
<option>Combo2</option></select>
```

6) **Check Box**

Check box untuk memilih satu atau lebih pernyataan dari beberapa pernyataan yang disediakan.

```
<input type="checkbox" name="nama_variabel" value="ON" checked>
```

7) **Submit**

Submit untuk mengirimkan semua variable data pada komponen-komponen form yang ada.

```
<input type="submit" name="submit" value="submit">
```

8) **Reset**

Reset untuk membatalkan semua penginputan yang telah dituliskan.

```
<input type="reset" name="reset" value="reset">
```

B. Pengolahan Data Dari Form

Form di HTML dikenal dengan adanya tag <FORM> dan ditutup dengan tag </FORM>. Di dalam tag pembuka <FORM> diikuti dengan atribut action dan method. Action menjelaskan ke h alaman yang digunakan untuk memproses input, sementara method digunakan untuk mengatur cara mem-parsing konten

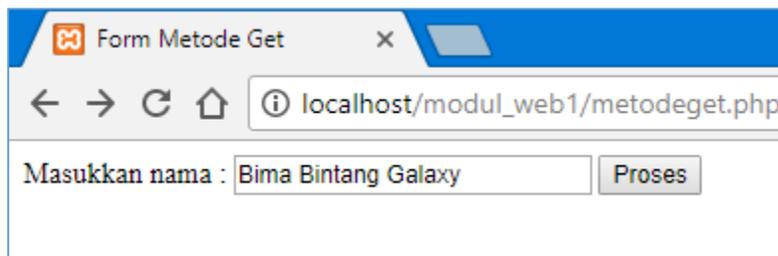
Web menerima input dari user atau pengunjung menggunakan metode GET dan POST. GET akan mengirimkan data bersama dengan URL, sedangkan POST akan mengirimkannya secara terpisah. User mengirimkan data input dengan mengisi teks atau pilihan pada attribut form html.

Proses Form menggunakan Metode GET.

File **metodeget.php**

```
<html>
<head>
    <title>FORM METODE GET</title>
</head>
<body>
<form action="metodegetproses.php" method="get">
Masukkan nama : <input type="text" name="nama" size="25">
<input type="submit" value="Proses">
</form>
</body>
</html>
```

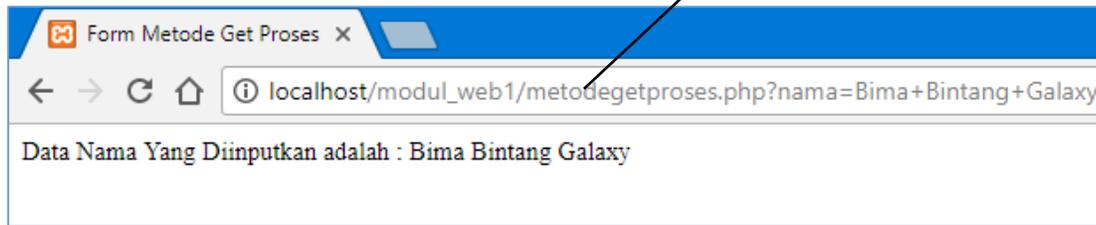
Hasilnya :



Buat file untuk memproses variable yang diberikan oleh file metodeget.php, beri nama filenya : **metodegetproses.php**

```
<html>
<head>
    <title>Method Get Proses</title>
</head>
<body>
Data Nama Yang Diinputkan Adalah : <?php echo $_GET["nama"]; ?>
</body>
</html>
```

Hasilnya :

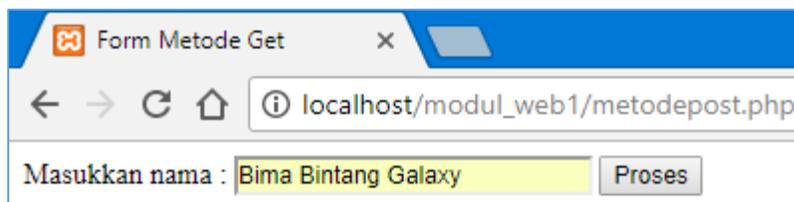


Proses Form menggunakan metode : POST

Untuk membuat inputan, dan beri nama file : metodepost.php

```
<html>
<head>
  <title>FORM METODE POST</title>
</head>
<body>
<form action="metodepostproses.php" method="post">
Masukkan nama : <input type="text" name="nama" size="25">
<input type="submit" value="Proses">
</form>
</body>
</html>
```

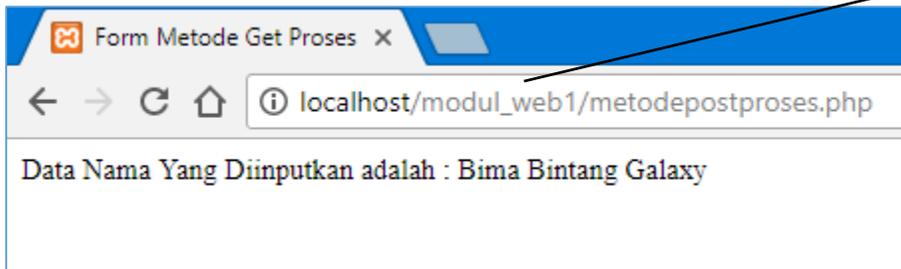
Hasilnya :



Buat file untuk memproses variable yang diberikan oleh file metodepost.php beri nama filenya : **metodepostproses.php**

```
<html>
<head>
  <title>Method Post Proses</title>
</head>
<body>
Data Nama Yang Diinputkan Adalah : <?php echo $_POST["nama"]; ?>
</body>
</html>
```

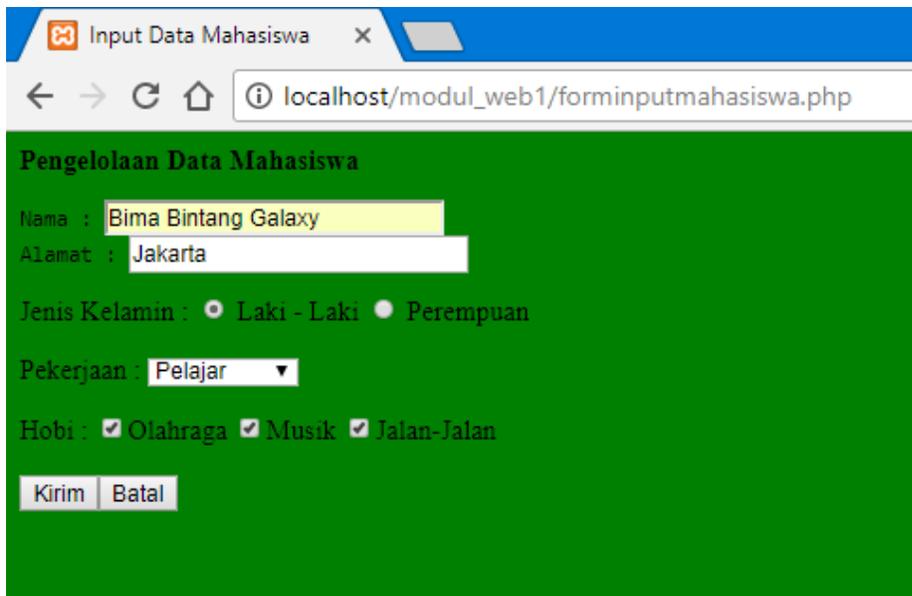
Hasilnya :



Karena menggunakan metode POST, data tidak dikirimkan bersama dengan URL

CONTOH Penggunaan Form :

Form input :



Form Output :



Data Mahasiswa	
Nama	Bima Bintang Galaxy
Alamat	Jakarta
Jenis Kelamin	Laki-Laki
Pekerjaan	Pelajar
Hobi	Olahraga,Musik,Jalan-Jalan

[INPUT DATA LAGI](#)

Script forminputmahasiswa.php

```
<html>
<head>
    <title>Input Data Mahasiswa</title>
</head>
<body bgcolor="green">
<form action="tampilmahasiswa.php" method="post">
<b>Pengelolaan Data Mahasiswa</b>
<br>
<pre>
Nama : <input type="text" name="nama" size="25" maxlength="50">
Alamat : <input type="text" name="alamat" size="25" maxlength="50">
</pre>
Jenis Kelamin :
<input type="radio" name="jeniskel" value="Laki-Laki"> Laki - Laki
<input type="radio" name="jeniskel" value="Perempuan"> Perempuan
<p>
Pekerjaan :
<select name="pekerjaan">
<option value="-Pilih-">
<option value="Pelajar">Pelajar
<option value="Karyawan">Karyawan
<option value="Wirausaha">Wirausaha
<option value="Lain-lain">Lain-lain
</select>
<p>
Hobi :
<input type="checkbox" name="hobi1" value="Olahraga">Olahraga
<input type="checkbox" name="hobi2" value="Musik">Musik
<input type="checkbox" name="hobi3" value="Jalan-Jalan">Jalan-Jalan
<p>
<input type="submit" value="Kirim"><input type="reset" value="Batal">
</form>
</body>
</html>
```

Script tampilmahasiswa.php

```
<html>
<head>
  <title> Data Mahasiswa </title>
</head>
<body>
<?php
$nama=$_POST['nama'];
$alamat=$_POST['alamat'];
$jeniskel=$_POST['jeniskel'];
$pekerjaan=$_POST['pekerjaan'];
$hobi1=$_POST['hobi1'];
$hobi2=$_POST['hobi2'];
$hobi3=$_POST['hobi3'];
?>
<table border=1 bgcolor="Cyan">
<tr>
<td colspan=2 align="center"><b>Data Mahasiswa</b></td>
</tr>
<tr>
<td>Nama</td><td><?php echo $nama; ?></td>
</tr>
<tr>
<td>Alamat</td><td><?php echo $alamat; ?></td>
</tr>
<tr>
<td>Jenis Kelamin</td><td><?php echo $jeniskel; ?></td>
</tr>
<tr>
<td>Pekerjaan</td><td><?php echo $pekerjaan; ?></td>
</tr>
<tr>
<td>Hobi</td><td><?php echo $hobi1, ", ", $hobi2, ", ", $hobi3; ?></td>
</tr></table>
<a href="forminputmahasiswa.php">INPUT DATA LAGI</a>
</body>
</html>
```

Contoh Pembuatan Form dengan Input, Proses, dan Output Dalam Satu File

Untuk membuat form input dan halaman untuk menampilkan dalam satu file, kita bisa menggunakan statement :

If (!Empty (nama_variabel))

Artinya jika variable yang dicari tidak kosong (alias ada) maka baru ditampilkan, sementara jika tidak ada maka tidak akan ditampilkan.

Yang perlu diketahui adalah digunakannya alamat action :

`$_server ['php_self']`

Artinya alamat action akan mengacu ke halaman itu sendiri dan bukan di halaman yang lain. Dengan demikian, jika form di submit, maka halaman yang dipanggil tetap halaman yang sama.

Contoh formdatadiri.php

```
<html>
<head>
  <title> Masukan Data</title>
</head>
<body>
<h1> Masukan Identitas Anda<h1>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
<pre>
Isikan Nama      : <input type="text" name="nama">
Isikan No Telp   : <input type="text" name="notelp">
Isikan Alamat    : <textarea name="alamat" rows="5" cols="40"></textarea>
<input type="submit" value="TAMPIL"><input type="reset" value="BATALL">
</pre>
</form>
<?php
$nama = $_POST['nama'];
$telp = $_POST['notelp'];
$alamat = $_POST['alamat'];
if(!empty($nama)){
echo "Nama      : $nama <br>"; }
if(!empty($telp)){
echo "No Telp   : $telp <br>"; }
if(!empty($alamat)){
echo "Alamat    : $alamat <br>"; }
?>
</body>
</html>
```

Hasil :

Inputkan datanya :



Masukan Data

localhost/modul_web1/FormDataDiri.php

Masukan Identitas Anda

Isikan Nama : Bima Bintang Galaxy

Isikan No Telp : 08791234567

Isikan Alamat : Jalan Anggrek No. 9 Bandung

TAMPIL BATALL

KLIK Tombol Tampil, maka akan tampil kiriman datanya.



Masukan Data

localhost/.../FormDataDiri.php

Masukan Identitas Anda

Isikan Nama :

Isikan No Telp :

Isikan Alamat :

TAMPI | BATAL

Nama : Bima Bintang Galaxy
No Telp : 08791234567
Alamat : Jalan Anggrek No. 9 Bandung

C. Pernyataan Seleksi

Sebagian besar bahasa pemrograman mengandung pernyataan seleksi. Pada dasarnya pernyataan seleksi adalah suatu mekanisme yang menjelaskan apakah pernyataan akan dikerjakan atau tidak, hal ini tergantung kondisi yang dirumuskan. Dalam bahasa pemrograman PHP pernyataan seleksi diterapkan dengan menggunakan statement IF dan Switch Case.

1. Statement IF

a. If Tunggal

Statement IF merupakan statement yang penting dan pasti terdapat di semua bahasa pemrograman. Statement ini berguna untuk membuat percabangan berdasarkan kondisi tertentu yang harus dipenuhi.

Bentuk umum Statement IF adalah sebagai berikut :

```
if ( kondisi )  
{  
    Statement;  
}
```

Prinsip kerjanya adalah perintah di atas akan dikerjakan apabila kondisi bernilai TRUE atau benar, sedangkan jika kondisi salah / FALSE maka statement di atas tidak akan dikerjakan

b. Pernyataan IF dan Else

Pernyataan ELSE merupakan bagian dari pernyataan if. Else digunakan untuk memberikan alternative perintah apabila kondisi bernilai salah / FALSE.

Bentuk umum : `if (kondisi)`
`{`
`Statement_1;`
`}`
`else`
`{`
`Statement_2;`
`}`

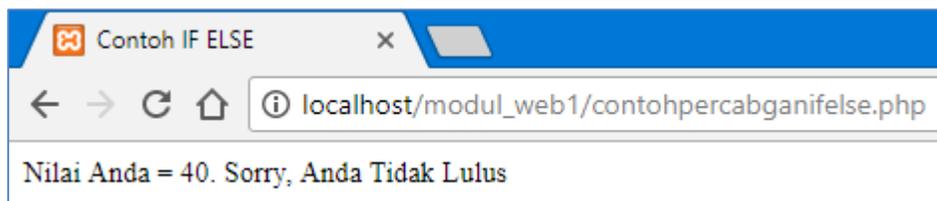
Contoh : contohpercanganifelse.php

```
<html>
<head>
    <title> Contoh IF ELSE</title>
</head>
<?php

    $nilai = 40;
    if ($nilai >= 60)
    {   echo "Nilai Anda = $nilai. Selamat, Anda Lulus" ;}
    else
    {   echo "Nilai Anda = $nilai. Sorry, Anda Tidak Lulus" ;}

?>
</body>
</html>
```

Hasilnya :



c. Pernyataan IF Majemuk

Jika pernyataan else memberikan alternative pilihan kedua, maka untuk pernyataan Elseif dapat digunakan untuk meumuskan banyak alternative pilihan (lebih dari dua pilihan).

Bentuk umum :

```
if ( kondisi_1 )  
{  
    Statement_1;  
}  
elseif ( kondis_2)
```

```

    {
        Statement_2;
    }
elseif ( kondisi_3)
{
    Statement_3;
}

else
{
    Statement_n;
}

```

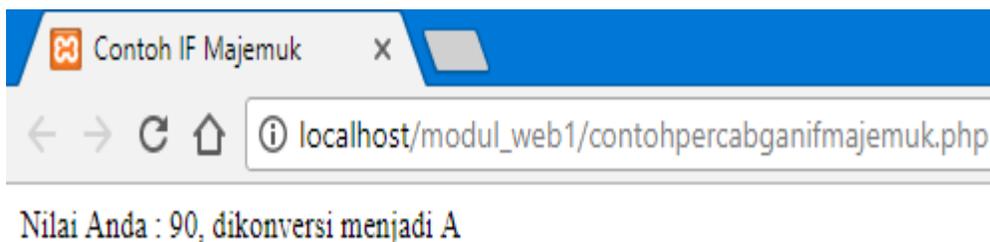
Contoh : contohpercabanganifmajemuk.php

```

<html>
<head>
    <title> Contoh IF Majemuk</title>
</head>
<?php
    $nilai = 90;
    if (($nilai >= 0)&&($nilai < 50))
    { $grade ="E";}
    elseif(($nilai >= 50)&&($nilai < 60))
    { $grade ="D";}
    elseif(($nilai >= 60)&&($nilai < 75))
    { $grade ="C";}
    elseif(($nilai >= 75)&&($nilai < 85))
    { $grade ="B";}
    elseif(($nilai >= 85)&&($nilai < 100))
    { $grade ="A";}
    else
    {$grade = "Nilai anda di luar jangkauan"; }
    echo "Nilai Anda : $nilai, dikonversi menjadi $grade";
?>
</body>
</html>

```

Hasilnya :



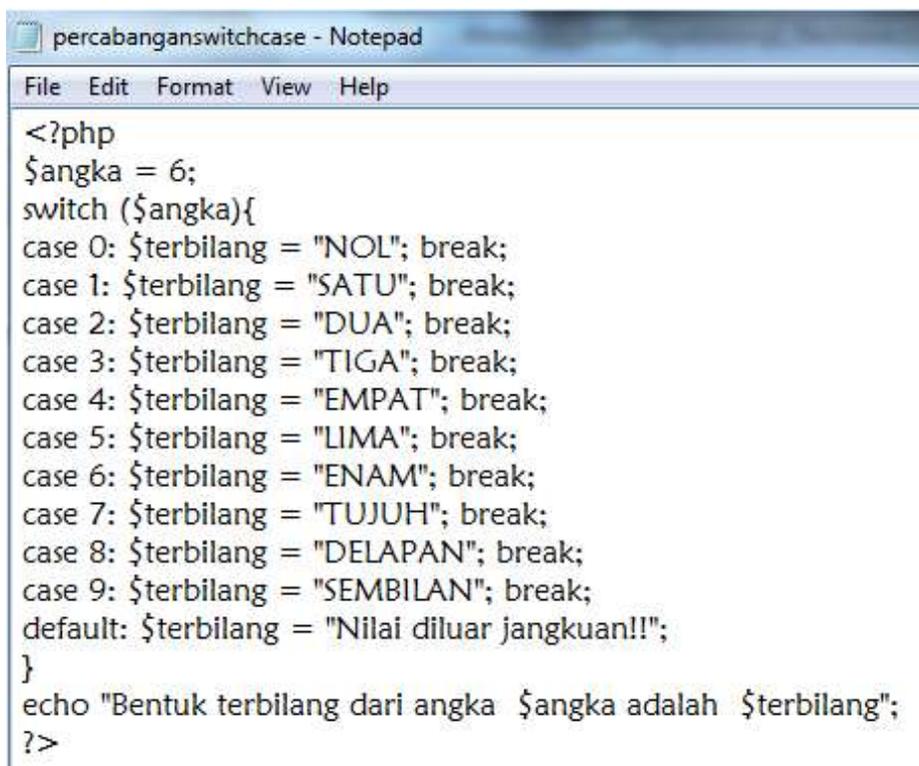
2. Statement Switch

Statement untuk pengatur alur program berikutnya adalah switch. Salah satu keuntungan switch adalah ada bisa langsung mengevaluasi satu statement dan memerintahkan aksi dalam jumlah yang lebih banyak.

Bentuk umum :

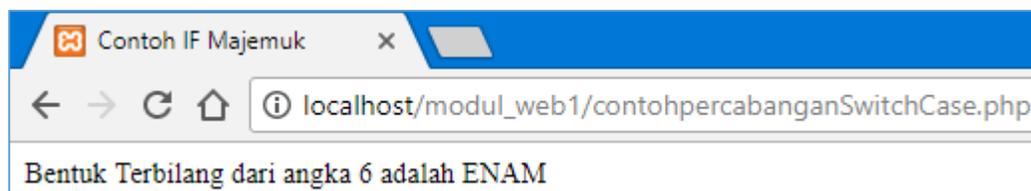
```
Switch ( nilai_ekspresi ){  
Case nilai_1 : statement_1; break;  
Case nilai_2 : statement_2; brea;  
Default: statement_n;}
```

Contoh :



```
percabanganswitchcase - Notepad  
File Edit Format View Help  
<?php  
$angka = 6;  
switch ($angka){  
case 0: $terbilang = "NOL"; break;  
case 1: $terbilang = "SATU"; break;  
case 2: $terbilang = "DUA"; break;  
case 3: $terbilang = "TIGA"; break;  
case 4: $terbilang = "EMPAT"; break;  
case 5: $terbilang = "LIMA"; break;  
case 6: $terbilang = "ENAM"; break;  
case 7: $terbilang = "TUJUH"; break;  
case 8: $terbilang = "DELAPAN"; break;  
case 9: $terbilang = "SEMBILAN"; break;  
default: $terbilang = "Nilai diluar jangkuan!!";  
}  
echo "Bentuk terbilang dari angka $angka adalah $terbilang";  
?>
```

Hasil:



```
Contoh IF Majemuk x  
localhost/modul_web1/contohpercabanganSwitchCase.php  
Bentuk Terbilang dari angka 6 adalah ENAM
```

D. Perulangan (Looping)

Perulangan / looping (kadang juga disebut iterasi) adalah sebuah instruksi program yang memerintahkan suatu tugas diulang - ulang berdasarkan kondisi tertentu.

1. Perulangan FOR

Merupakan bentuk perulangan yang sangat sederhana, dengan menggunakan fungsi ini, anda dapat melakukan pengulangan data sampai melampaui batas yang diinginkan.

```
for (init counter; test counter; increment counter) {  
    code to be executed;  
}
```

2. Perulangan WHILE

Pada bentuk perulangan ini, pernyataan akan terus dikerjakan apabila masih belum mencapai batas perulangan.

```
while (condition is true) {  
    code to be executed;  
}
```

3. Perulangan DO - WHILE

pernyataan akan dikerjakan terlebih dahulu sebelum melakukan pengecekan batas perulangan. Apabila masih belum mencapai batas perulangan maka pengulangan akan terus dilakukan.

```
do {  
    code to be executed;  
} while (condition is true);
```

4. Pernyataan foreach -

perulangan yang dilakukan untuk blok kode dari setiap elemen yang ada di array.

```
foreach ($array as $value) {  
    code to be executed;  
}
```

Contoh :

1. Perulangan FOR = contohfor.php

```
<html>
<head>
  <title> Perulangan FOR </title>
</head>
<body>
  nilai awal angka = 1
  <br><br>
  <?php
    for ($angka = 1; $angka <= 10 ; $angka++)
    {
      echo "Angka :".$angka."<br>";
    }
  ?>
</body>
</html>
```

2. Perulangan FOR dalam FORM = contohfor_form.php

```
<html>
<head>
  <title> Perulangan FOR </title>
</head>
<body>
  Penggunaan pada form :
  <br>
  <?php
    echo "<form name = form1 method=post>";
    echo "Tanggal" ;
    echo "<select name = tanggal>";
    for ($tanggal = 1 ;$tanggal <=31 ; $tanggal++)
    {
      echo "<option value=".$tanggal.">".$tanggal."</option>";
    }
    echo "</select>";
    echo "</form>";
  ?>
</body>
</html>
```

3. Perulangan WHILE = contohwhile.php

```
<html>
<head>
  <title> Penggunaan WHILE </title>
</head>
<body>
  Menggunakan WHILE
  <br>
  <?php
    $jumlah=1;
    while ($jumlah <=5)
    {
      echo $jumlah++;
      echo "<br>";
    }
  ?>
</body>
</html>
```

4. Perulangan DO - WHILE= contohdowhile.php

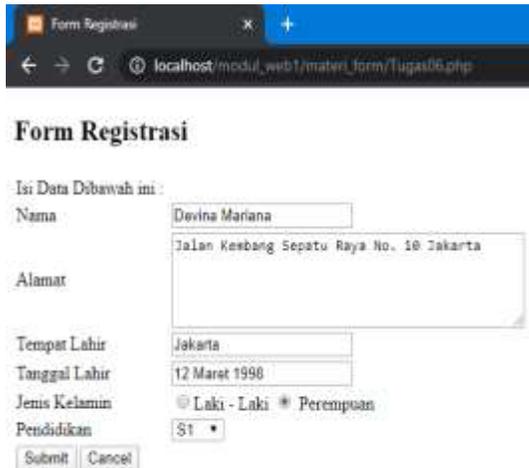
```
<html>
<head>
  <title> Penggunaan DO WHILE </title>
</head>
<body>
  Menggunakan DO WHILE
  <br>
  <?php
    $jumlah=10;
    do
    {
      echo $jumlah++;
      echo "<br>";
    }
    while ($jumlah <=1)
  ?>
</body>
</html>
```

5. Perulangan Foreach = contoforeach.php

```
<html>
<head>
  <title> Penggunaan Foreach </title>
</head>
<body>
  Menggunakan Foreach
  <br>
  <?php
    $warna = array("merah","biru","hijau","kuning");
    foreach ($warna as $nilai) {
      echo "$nilai <br>";
    }
  ?>
</body>
</html>
```

Tugas 03 :

1. Buat script program untuk membuat form input dengan data sbb :



The screenshot shows a web browser window with the address bar displaying 'localhost/modul_web1/materi_form/Tugas06.php'. The page title is 'Form Registrasi'. Below the title, there is a heading 'Isi Data Dibawah ini :'. The form contains several input fields: 'Nama' (filled with 'Devina Mariana'), 'Alamat' (filled with 'Jalan Kembang Sepatu Raya No. 10 Jakarta'), 'Tempat Lahir' (filled with 'Jakarta'), 'Tanggal Lahir' (filled with '12 Maret 1998'), 'Jenis Kelamin' (radio buttons for 'Laki - Laki' and 'Perempuan', with 'Perempuan' selected), and 'Pendidikan' (a dropdown menu with 'S1' selected). At the bottom of the form are 'Submit' and 'Cancel' buttons.

- Buat script program untuk memanggil data dari form input dengan bentuk sbb :



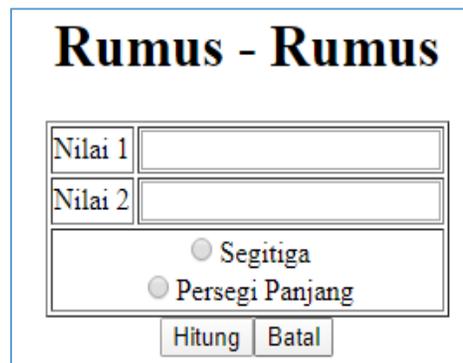
The screenshot shows a web browser window with the address bar displaying 'localhost/modul_web1/materi_form/Tugas06Proses.php'. The page title is 'Tugas 06'. Below the title, there is a heading 'Data Registrasi'. A table displays the registration data:

Nama	Devina Mariana
Alamat	Jalan Kembang Sepatu Raya No. 10 Jakarta
Tempat Lahir	Jakarta
Tanggal Lahir	12 Maret 1998
Jenis Kelamin	Perempuan
Pendidikan	S1

Below the table is a blue link labeled 'Back To Home'.

2. Buatlah script sehingga mendapatkan tampilan sebagai berikut :

Tampilan Input



The screenshot shows a web form titled 'Rumus - Rumus'. It contains two input fields labeled 'Nilai 1' and 'Nilai 2'. Below these fields are two radio buttons: 'Segitiga' and 'Persegi Panjang', with 'Persegi Panjang' selected. At the bottom of the form are 'Hitung' and 'Batal' buttons.

Tampilan Output

```
HASIL HITUNG RUMUS

Nilai a adalah = 5
Nilai b adalah = 6
Rumus Yang Dipilih adalah = Persegi Panjang
Hasil Perhitungan Rumus = 30
```

Ketentuan Soal :

- Jika Nilai 1 dan 2 = 0 maka tidak ada hasil perhitungan
- Jika Nilai 1 dan 2 terisi nilai maka
 - Jika memilih perhitungan rumus Segitiga ($1/2*(\text{nilai 1}*\text{nilai2})$)
 - Jika memilih perhitungan rumus Persegi Panjang ($\text{nilai 1}*\text{nilai2}$)

MINGGU KE 4

Javascript dan CSS

Deskripsi:

Membahas mengenai javascript dan CSS, sintaks penulisan, serta penggunaannya dalam pemrograman web.

Tujuan Pembelajaran:

Setelah melakukan bagian ini mahasiswa mampu:

1. Memahami sintaks Javascript dan menggunakan Javascript
2. memahami sintaks CSS dan menggunakan CSS

A. JavaScript

Javascript adalah bahasa script yang populer di internet dan dapat bekerja di sebagian besar penjelajah web browser seperti Internet Explorer (IE), Mozilla Firefox, Netscape, opera dan web browser lainnya. Kode javascript biasa dituliskan dalam bentuk fungsi (Function) yang ditaruh di bagian dalam tag <head> yang dibuka dengan tag <script language = " javascript">

Isi dari script javascript sama dengan konsep yang sudah dipelajari dalam materi PHP, yakni ada deklarasi variabel, penggunaan operator, percabangan, looping, dan fungsi. Di dalam java script juga sebuah komponen Alert yang digunakan untuk menampilkan kotak pesan pada browser ketika fungsinya di jalankan.

Untuk berlatih deklarasi script pada javascript, salin contoh-contoh berikut ini pada editor anda. Dan jalankan pada browser, amati tampilannya.

Latihan Javacript :

1. Menuliskan teks = contohjs1.html

```
<html>
<body>
<script type="text/javascript">
document.write("Hello World!")
</script>
</body>
</html>
```

2. Memformat teks dengan tag HTML = contohjs2.html

```
<html>
<body>
<script type="text/javascript">
document.write("<h1>Hello World!</h1>")
</script>
</body>
</html>
```

3. JavaScript yang diletakkan pada bagian HEAD = contohjs3.html

```
<html>
<head>
<script type="text/javascript">

function message()
{
alert("This alert box was called with the
      onload event")
}
</script>
</head>
<body onload="message()">
</body>
</html>
```

4. JavaScript yang diletakkan pada bagian BODY = contohjs4.html

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
document.write("This message is written
              when the page loads")
</script>
</body>
</html>
```

5. Fungsi = contohjs5.html

```
<html>
<head>
  <script type="text/javascript">
    function myfunction()
    {
      alert("HELLO")
    }
  </script>
</head>
<body>
<form>
<input type="button" onclick="myfunction()"
  value="Panggil MyFunction">
</form>
<p>tekan tombol untuk memanggil fungsi
myfunction di dalam javascript</p>
</body>
</html>
```

6. Fungsi dengan argumen = contohjs6.html

```
<html>
<head>
<script type="text/javascript">
function myfunction(txt)
{
alert(txt)
}
</script>
</head>
<body>
<form>
<input type="button"
onclick="myfunction('Good Morning!')"
value="Selamat Pagi">
<input type="button"
onclick="myfunction('Good Evening!')"
value="Selamat Malam">
</form>
<p>
ketika di tekan salah satu tombol maka
fungsi akan di panggil dan pesan akan di
tampilkan
</p>
</body>
</html>
```

7. Memunculkan tanggal lengkap = contohjs7.html

```
<html>
<body>
<script type="text/javascript">
var d=new Date()
var weekday=new
Array("Sunday", "Monday", "Tuesday", "Wednesday"
      , "Thursday", "Friday", "Saturday")
var monthname=new
Array("Jan", "Feb", "Mar", "Apr", "May", "Jun", "
      Jul", "Aug", "Sep", "Oct", "Nov", "Dec")
document.write(weekday[d.getDay()] + " ")
document.write(d.getDate() + ". ")
document.write(monthname[d.getMonth()] + " ")

document.write(d.getFullYear())
</script>
</body>
</html>
```

B. Cascading Style Sheets (CSS)

CSS = Cascading Style Sheets (Bahasa lembar Gaya). CSS merupakan bahasa yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup / markup language. Jika kita berbicara dalam konteks web, bisa di artikan secara bebas sebagai : CSS merupakan bahasa yang digunakan untuk mengatur tampilan / desain suatu halaman HTML.

Beberapa hal yang dapat dilakukan dengan CSS.

1. Perancangan desain text dapat dilakukan dengan mendefinisikan fonts (huruf) , colors (warna), margins (ukuran), latar belakang (background), ukuran font (font sizes) dan lain-lain. Elemen-elemen seperti colors (warna) , fonts (huruf), sizes (ukuran) dan spacing (jarak) disebut juga "styles".
2. Cascading Style Sheets juga bisa berarti meletakkan styles yang berbeda pada layers (lapisan) yang berbeda.

Ada 3 cara untuk memasang CSS pada dokumen HTML yaitu:

1. External Style Sheet

Aturan CSS disimpan pada suatu file sehingga terpisah dari dokumen HTML. Kemudian tambahkan kode pemanggilan file CSS dalam dokumen HTML. Akhiran file CSS adalah .css

```
File CSS (misalnya style.css) berisi:  
p {text-align:justify;}  
  
Dokumen HTML berisi:  
<head>  
    <title>CSS secara eksternal</title>  
    <link rel="stylesheet" type="text/css" href="style.css" />  
</head>  
<body>  
    <p>Paragraf yang ini diatur CSS secara eksternal</p>  
</body>
```

2. Internal Style Sheet

Aturan CSS ditulis pada bagian HEAD dokumen HTML menggunakan tag <style>

```
<head>  
    <title>CSS secara internal</title>  
    <style type="text/css">  
        P {text-align:justify;}  
    </style>  
</head>  
<body>  
    <p>Paragraf yang ini diatur CSS secara internal</p>  
</body>
```

3. Inline Style Sheet

Aturan CSS ditulis langsung pada tag HTML yang akan diatur tampilannya menggunakan atribut style:

```
<p style="text-align:justify;">Paragraf ini diatur CSS secara inline</p>
```

SATUAN DALAM CSS

1. Statik

- * in -- satuan inchi
- * cm -- satuan centimeter
- * mm -- satuan milimeter

- * pt -- satuan point (1point = 1/72 inchi)
- * pc -- satuan pica (1pica = 12 point)
- * px -- satuan pixel (satu titik gambar terkecil dalam layar monitor)

2. Relatif

- * % -- satuan persen
- * em -- atau ems (1em = ukuran font yang tengah ada dalam elemen)
- * ex -- 1ex = x-height suatu font (x-height biasanya setengah ukuran font)

1. MENULIS CSS

Sintaks penulisan CSS sebagai berikut:



Penjelasan:

Aturan CSS terdiri 2 bagian:

a) Selector

Biasanya berupa tag HTML, id, class

id menggunakan tanda # didepan nama selector

class menggunakan tanda titik didepan nama selector

contoh :

h1 { color : blue ; } → tag html h1

#teks { color :green; } → id

.warna { color : red; } → class

b) Declaration

Berisi aturan-aturan css yang terdiri dari properti dan nilainya yang dipisahkan oleh tanda titik dua. Setiap aturan css harus diakhiri dengan tanda titik koma.

2. Selector ID dan Class pada CSS

Untuk selector id pada css ditandai dengan tanda #(pagar) contoh penulisan seperti berikut:

```
#teks
{
    color: blue;
    font-family: Calibri;
}
```

Penggunaanya dalam script HTML ::

```
<body>
    <p id="teks"> TEST </p>
</body>
```

Yang perlu di perhatikan jika menggunakan selector id :

- Sebuah elemen HTML hanya boleh memiliki 1 id
- Setiap halaman hanya boleh memiliki 1 elemen dengan id tersebut
- Dapat di gunakan sebagai penanda halaman untuk link
- Digunakan juga untuk javascript
- Sebaiknya tidak digunakan untuk css (lebih baik gunakan class)

Untuk selector class pada css ditandai dengan tanda .(titik) contoh penulisan seperti berikut:

```
.warna
{
    background-color: Lightgreen;
}
```

Penggunaanya dalam script HTML :

```
<body class="warna">
</body>
```

3. Properti-properti CSS

Properti CSS jumlahnya sangat banyak, berikut beberapa diantaranya:

Properti	Fungsi	Nilai	Contoh
color	Mengatur warna teks	Nama warna, kode hexa warna (#ffffff:putih, #000000:hitam, #ff0000:merah), rgb(0,250,100)	Color:#ff5590;
Background-color	Mengatur warna latar	Nama warna, kode hexa warna, rgb(200,200,200)	Background-color:rgb(200,0,55);
Background-image	Mengatur gambar latar	Nama file gambar	Background-image:url(banner.jpg);
Text-align	Mengatur perataan teks	Left, right, center, justify	Text-align:justify;
Text-decoration	Mengatur dekorasi teks	Underline, none	Text-decoration:underline;
Line-height	Mengatur tinggi baris	Piksel, prosentase, em	Line-height:120%;
Font-family	Mengatur jenis font	"times new roman", arial, georgia	Font-family:arial;
Font-size	Mengatur ukuran karakter	Piksel, prosentase, em, pt	Font-size:12pt;
Margin			
Padding			
Border-width	Mengatur ketebalan garis batas	Piksel, prosentase, thin, thick	Border-width:1px;
Border-style	Mengatur jenis garis batas	Solid, dotted, dashed, double, none	Border-style:solid;
float	Mengatur obyek agar mengambang	Left, right, none	Float:left;
clear	Menghentikan	Left, right, both, none	Clear:both;

Pseduo-Class

Adalah sebuah kelas semu yang dimiliki oleh elemen HTML, yang membuat kita dapat mendefinisikan style pada keadaan tertentu dari elemen tersebut. Pseduo-class terbagi menjadi beberapa type, sebagai berikut :

1. Yang berhubungan dengan link

- : link

Style default pada sebuah link (a yang memiliki href)

- : hover

Style ketika kursor mouse berada diatas sebuah link / elemen

- : active

Style ketika sebuah link di klik (keadaan aktif)

- : visisted

Style ketika sebuah link sudah pernah di kunjungi sebelumnya (m menggunakan browser yang sama)

2. Yang berhubungan dengan posisi elemen (ada pada css 3)

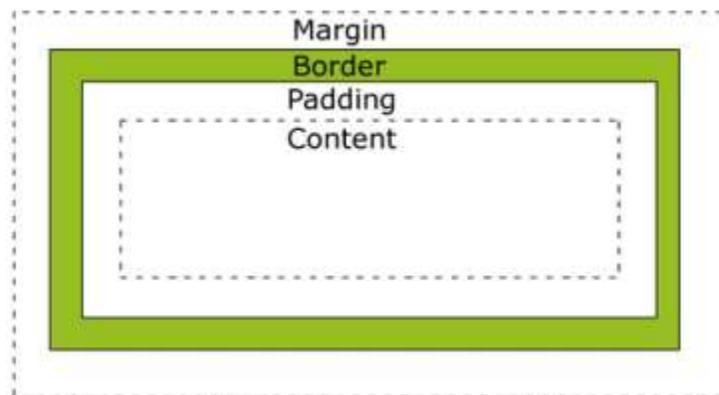
- : first-child

Memilih elemen pertama dari sebuah parent (elemen pembungkusnya)

- : last-child
Memilih elemen terakhir dari sebuah parent (elemen pembungkusnya)
- : nth-child(n)
Memilih elemen ke (n) dari sebuah parent (elemen pembungkusnya)
n bisa berarti urutan 1,2,3,..... atau pola (2n),(3n+2), atau ganjil dan genap, even & odd
- : first-of-type
Memilih elemen pertama dari sebuah jenis / tipe tag
- : last-of-type
Memilih elemen terakhir dari sebuah jenis / tipe tag

4. PADDING, MARGIN DAN BORDER

Dalam CSS dikenal istilah 'Box Model'. Perhatikan gambar berikut ini:



Padding : Menentukan jarak komponen body ke border atau Ukuran jarak bagian dalam

Border : Adalah garis tepi dari komponen

Margin : Adalah Ukuran jarak bagian luar atau ukuran jarak sesudah Border

CSS menggunakan konsep ini dalam mengatur tag-tag HTML. Pada gambar, bayangkan area 'Content' misalnya adalah sebuah paragraph. Obyek paragraph ini akan dianggap CSS memiliki area padding, border, dan margin disekitarnya. Keberadaan area-area ini berguna untuk pengaturan tata letak. Misalnya ingin diatur agar 2 buah gambar yang terletak

berdampingan tidak terlalu rapat, maka kita dapat memperbesar lebar dari area margin agar jarak antara gambar lebih lebar.

Padding

ditulis dengan CSS `padding:5px 5px 5px 5px;` urutan nilai angkanya adalah atas, kanan, bawah dan kiri, atau Anda bisa menggunakan `padding-left:5px;` → ini adalah untuk pengaturan padding bagian kiri `padding-right:5px;` → ini adalah untuk pengaturan padding kanan `padding-top:5px;` → untuk bagian atas dan `padding-bottom:5px;` → untuk bagian bawah, Ingat satuan px(pixels) bisa kamu ganti sesuai satuan yang lain yang sesuai

Border

Ditulis dengan CSS `border:1px dotted #000000;` → urutan penggunaannya adalah ukuran border, style border dan warna border, atau bisa menggunakan `border-width:1px;` → ini adalah ketebalan border `border-style:dotted;` → ini adalah jenis bordernya bisa kamu ganti dengan dashed, solid, double, groove, ridge, inset, outset dan lainnya `border-color:#FFFFFF;` → ini adalah warna dari border.. kamu bisa mengganti code warnanya (www.colorschemer.com/online)

Margin

Ditulis dengan CSS `margin:5px 5px 5px 5px;` → urutannya atas, kanan, bawah dan kiri, atau bisa menggunakan seperti padding diatas `margin-left:5px;`
`margin-right:5px;`
`margin-top:5px;`
`margin-bottom:5px;`
keterangan lainya bisa mengikuti keterangan padding diatas

Latihan CSS :

- Salin script berikut ini, lalu simpan di folder modul_web1 dengan nama style_css.css

```
/*pseduo class link*/
.hello:link{
    color :orange;
}
.hello:hover{
    color: green;
    font-family: arial;
    font-size: 20px;
}
.hello:active{
    font-style: italic;
}
.hello:visited {
    color: red;
}

/* css carikan a,di dalam li yang merupakan
anak pertama*/
li:first-child a{
    color:green;
}
/* css carikan a,di dalam li yang merupakan
anak terakhir*/
li:last-child a{
    color:green;
}
/* css carikan a, di dalam li yang merupakan
anak genap*/
li:nth-child(even) a{
    color:green;
}
/* css carikan a, di dalam li yang merupakan
anak ganjil*/
li:nth-child(odd) a{
    color:violet;
}

/* css carikan p, di dalam halaman yang urutannya pertama*/
p:first-of-type {
    color: red;
}

/* css carikan p, di dalam halaman yang urutannya terakhir*/
p:last-of-type {
    color: orange;
}
```

- Salin script berikut ini, lalu simpan di folder modul_web1 dengan nama latihan_css.html

```
<!DOCTYPE html>
<html>
<head>
  <title> Latihan CSS</title>
  <link rel="stylesheet" type="text/css" href="style_css.css">
</head>
<body>
<a href="#" class="hello">Hello World</a>
<ul>
  <li><a href="#">Link 1</a></li>
  <li><a href="#">Link 2</a></li>
  <li><a href="#">Link 3</a></li>
  <li><a href="#">Link 4</a></li>
  <li><a href="#">Link 5</a></li>
  <li><a href="#">Link 6</a></li>
  <li><a href="#">Link 7</a></li>
  <li><a href="#">Link 8</a></li>
  <li><a href="#">Link 9</a></li>
  <li><a href="#">Link 10</a></li>
</ul>
<p>Lorem Ipsum adalah contoh teks atau dummy dalam industri percetakan dan penataan huruf atau typesetting. Lorem Ipsum telah menjadi standar contoh teks sejak tahun 1500an, saat seorang tukang cetak yang tidak dikenal mengambil sebuah kumpulan teks dan mengacaknya untuk menjadi sebuah buku contoh huruf. </p>
<p> Ia tidak hanya bertahan selama 5 abad, tapi juga telah beralih ke penataan huruf elektronik, tanpa ada perubahan apapun. Ia mulai dipopulerkan pada tahun 1960 dengan diluncurkannya lembaran-lembaran Letraset yang menggunakan kalimat-kalimat dari Lorem Ipsum, dan seiring munculnya perangkat lunak Desktop Publishing seperti Aldus PageMaker juga memiliki versi Lorem Ipsum.</p>
</body>
</html>
```

- Hasil di web browser :

[Hello World](#)

- [Link 1](#)
- [Link 2](#)
- [Link 3](#)
- [Link 4](#)
- [Link 5](#)
- [Link 6](#)
- [Link 7](#)
- [Link 8](#)
- [Link 9](#)
- [Link 10](#)

Lorem Ipsum adalah contoh teks atau dummy dalam industri percetakan dan penataan huruf atau typesetting. Lorem Ipsum telah menjadi standar contoh teks sejak tahun 1500an, saat seorang tukang cetak yang tidak dikenal mengambil sebuah kumpulan teks dan mengacaknya untuk menjadi sebuah buku contoh huruf.

Ia tidak hanya bertahan selama 5 abad, tapi juga telah beralih ke penataan huruf elektronik, tanpa ada perubahan apapun. Ia mulai dipopulerkan pada tahun 1960 dengan diluncurkannya lembaran-lembaran Letraset yang menggunakan kalimat-kalimat dari Lorem Ipsum, dan seiring munculnya perangkat lunak Desktop Publishing seperti Aldus PageMaker juga memiliki versi Lorem Ipsum.

MINGGU KE 5

Membuat Template Web

Deskripsi:

Pada pembahasan materi kali ini kita akan membuat Design Web Responsive. Design Web Responsive adalah sebuah metode atau pendekatan sistem web desain yang bertujuan memberikan pengalaman berselancar yang optimal dalam berbagai perangkat, baik mobile maupun komputer

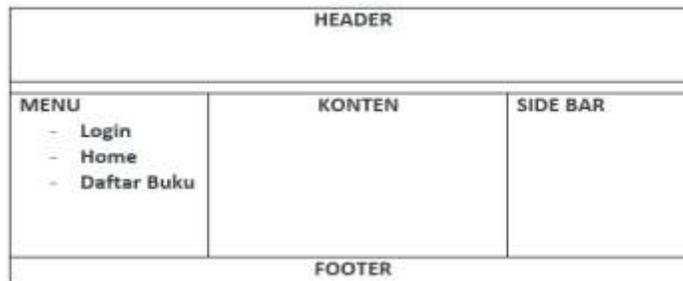
Tujuan Pembelajaran:

Setelah melakukan bagian ini mahasiswa mampu:

1. Memahami dan menggunakan HTML dalam membuat template web
2. Memahami dan menggunakan CSS dalam membuat template web

Langkah-langkah yang perlu di gunakan dalam pembuatan web ini adalah :

1. Tentukan tema website
 - Tema web perpustakaan
2. Membuat sketsa / blueprint / rancangan dari halaman web yang akan di buat
Seperti contoh berikut ini :



3. Web responsive berarti web yang tampilannya mengikuti ukuran layar gadget yang di gunakan untuk akses web tersebut
 - Buat folder di xampp/htdocs dengan nama : perpustakaan
 - Didalam folder perpustakaan, siapkan 2 buah folder :

- images → untuk simpan gambar
 - css → untuk simpan file css
- Buka editor, salin script css berikut : simpan di folder css dengan nama style.css

```

*{
  padding      : 0;
  margin       : 0;
  box-sizing   : border-box;
}
body{
  background   : #dedede;
  font-family  : 'Verdana',arial,sans-serif;
  font-size    : 15px;
}

p{
  margin-bottom : 20px;
  line-height   : 1.5em;
}

h3{
  margin-bottom : 20px;
  border-bottom : 1px solid #aaa;
}

a{
  text-decoration : none;
  color           : #333;
}

a:hover{
  color : #666;
}

.container{
  max-width  : 1080px;
  margin     : 20px auto;
  background : #fff;
  overflow   : hidden;
  padding    : 10px;
}

.header{
  border      : 1px solid #dedede;
  padding    : 10px;
  margin     : 10px;
  background : #9E9AFB;
}

/* main */
.left{

```

```

width      : 250px;
border     : 1px solid #dedede;
padding    : 10px;
margin     : 10px;
float      : left;
}

.left ul{
    list-style-type : none;
}

.left ul li{
    display : block;
}

.left ul li a{
    display      :block;
    border-bottom : 1px solid #dedede;
    margin-bottom : 10px;
    padding      : 10px 5px;
    font-color   : #D3D2ED;
}

.left ul li a:hover{
    color:#461AF3;
}

.middle{
width      : 500px;
border     : 1px solid #dedede;
padding    : 10px;
margin     : 10px;
float      : left;
}

.middle img{
    max-width : 100%;
    height    : auto;
}

.middle a{
    font-wight:bold;
}

.right{
width      : 250px;
border     : 1px solid #dedede;
padding    : 10px;
margin     : 10px;
float      : left;
}

.right ul{
    list-style-type : none;
}

.right ul li{
    display : block;
}

```

```

}

.right ul li a{
    display      :block;
    border-bottom : 1px solid #dedede;
    margin-bottom : 10px;
    padding      : 10px 5px;
}

.right ul li a:hover{
    color:#461AF3;
}

.footer{
    clear      : both;
    border     : 1px solid #dedede;
    padding    : 15px;
    margin     : 10px;
    background : #9E9AFB;
}

@media screen and (max-width:959px){
    #container{
        width : 100%;
    }
    #left-column{
        width : 70%;
    }
    #right-column{
        width : 30%;
    }
    img{
        width : 100%;
    }
}

/*MEDIA QUERIES ( Responsive )
******/
@media screen and (max-width:1080px)
{
    .container{
        width : 100%;
    }
    .left{
        width      : 25%;
        background : #D6CCFE;
    }
    .middle{
        width : 68%;
        float : right;
    }
    .right{
        clear      : both;
        padding    : 1% 4%;
    }
}

```

```

        width      : auto;
        float      : none;
        background  : #D6CCFE;
    }
}

/* untuk ukuran layar 700px kebawah */
@media screen and (max-width: 780px)
{
    .header,
    .footer{
        text-align : center;
    }
    .left {
        width      : auto;
        float      : none;
    }

    .middle {
        width      : auto;
        float      : none;
    }

    .right {
        width      : auto;
        float      : none;
    }
}
}

```

Buatlah script untuk tampilan berikut : index.html simpan di dalam folder perpus



Tampilan diatas ini kodenya:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title> Website Responsive</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/main.css">
</head>
<body>
  <div class="container">
    <div class="header">
      <h1 align="center"> Selamat Datang DI Perpustakaan XYZ </h1>
    </div>
    <div class="main">
      <div class="left">
        <h3 align="center">Menu</h3>
        <ul>
          <li><a href="#">Login</a></li>
          <li><a href="#">Home</a></li>
          <li><a href="#">Daftar Buku</a></li>
        </ul>
      </div>
      <div class="middle">
        <h3 align="center">Berita</h3>
        <p align="center"><strong><a href="#" >Membuat Design Web
Responsive</a></strong></p>
        <p></p>
        <p><a href="#">Baca Selengkapnya >> </a>
      </div>
      <div class="right">
        <h3 align="center">Buku Terpopuler</h3>
        <p>
          <ul>
            <li><a href="#">Web Design</a></li>
            <li><a href="#">Pemrograman</a></li>
            <li><a href="#">Database</a></li>
          </ul>
        </p>
      </div>
    </div>
    <div class="footer">
      <p align="center">Coppright &copy; 2018 Belajar CSS Responsive
    </a></p>
  </div>
</div>
</body>
</html>
```

Penjelasan Source Code

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

merupakan syntax yang berfungsi untuk menampilkan web sesuai dengan layar.

```
<link rel="stylesheet" href="css/main.css">
```

merupakan External Style sheet yang digunakan untuk mengload file css .

Pada Struktur HTML berikut terdapat 3 class utama yaitu header , main dan footer .

Yaitu :

```
<div class="container">
  <div class="header">digunakan untuk bagian header</div>
  <div class="main">
    <div class="left">Digunakan kolom bagian Kiri...</div>
    <div class="middle">Digunakan kolom bagian tengah...</div>
    <div class="right">Digunakan kolom bagian kanan...</div>
  </div>
  <div class="footer">Digunakan bagian footer...</div>
</div>
```

MINGGU KE 6

Pengenalan Codeigniter

Deskripsi:

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

Tujuan Pembelajaran:

Setelah melakukan bagian ini mahasiswa mampu:

1. Memahami konsep dasar Framework Codeigniter
2. memahamai kelebihan dari codeigniter
3. memahami cara instalasi codeigniter

A. Framework Web

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

Framework web untuk PHP:

- ✓ Codeigniter
- ✓ Yii
- ✓ Slim framework
- ✓ Zend framework
- ✓ Laravel
- ✓ Symfony
- ✓ CakePHP
- ✓ Phalcon
- ✓ FuelPHP, dll

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

B. Codeigniter

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

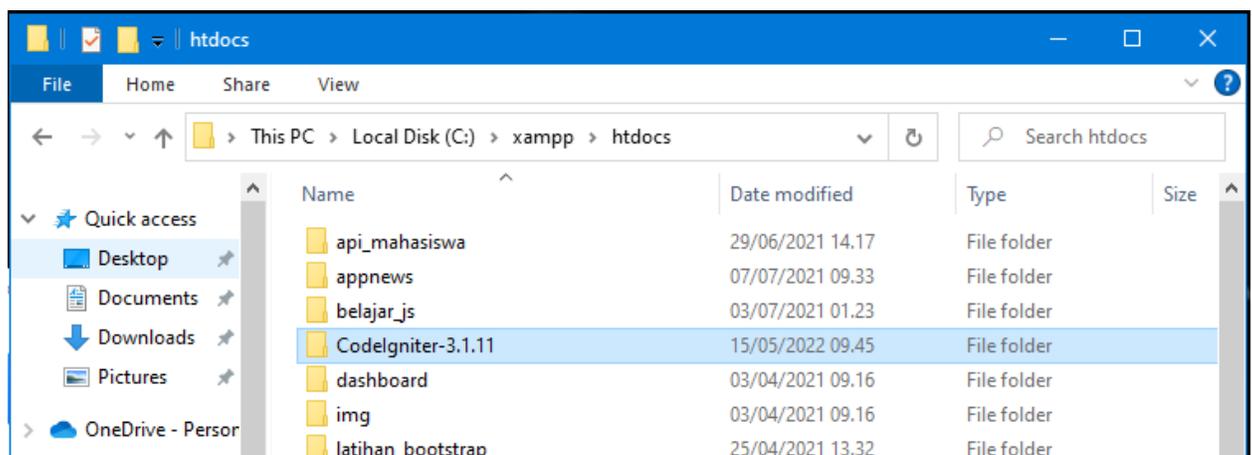
C. Instalasi Software

Software yang dibutuhkan pada pembelajaran kali ini yaitu Codeigniter versi 3.1.11 yang dapat didownload pada link <https://codeigniter.com> atau dapat anda dapat download pada link berikut:

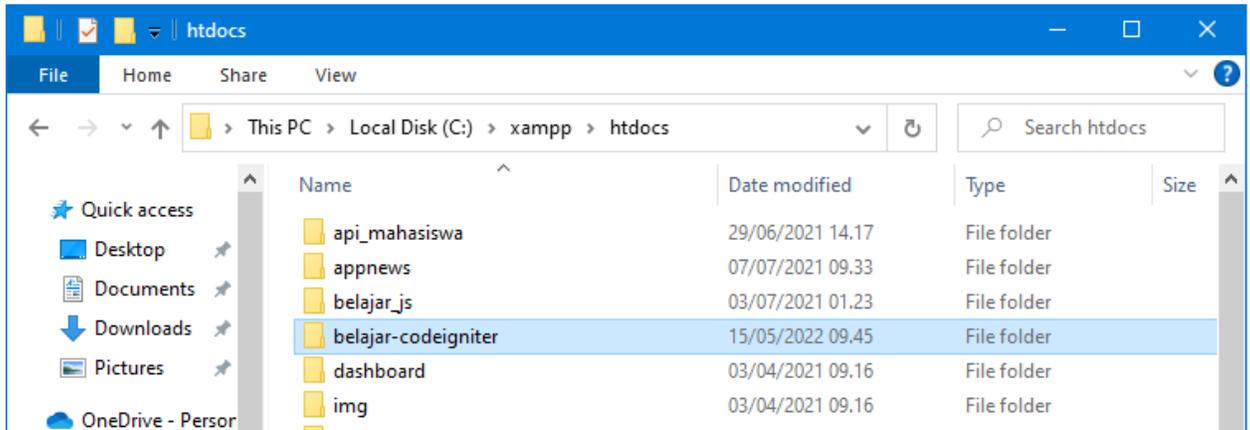
https://drive.google.com/file/d/1Xb-w_qabJCcLKGt-WxXA0h04vOQccKcq/view?usp=sharing

1. Instalasi Codeigniter

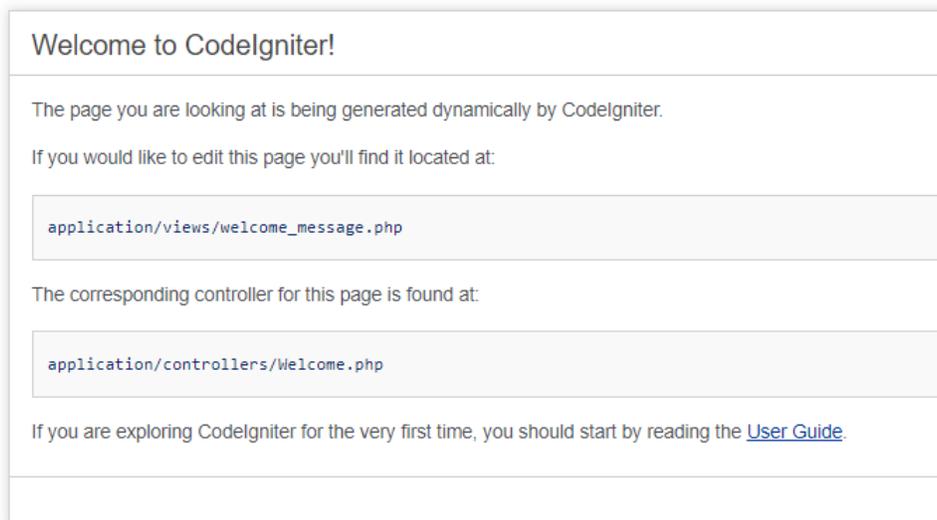
- a. Download Package codeigniter pada situ resminya yang sudah disebutkan di atas.
- b. Setelah didownload, ekstrak file codeigniter pada direktori **C:\xampp\htdocs**



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



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

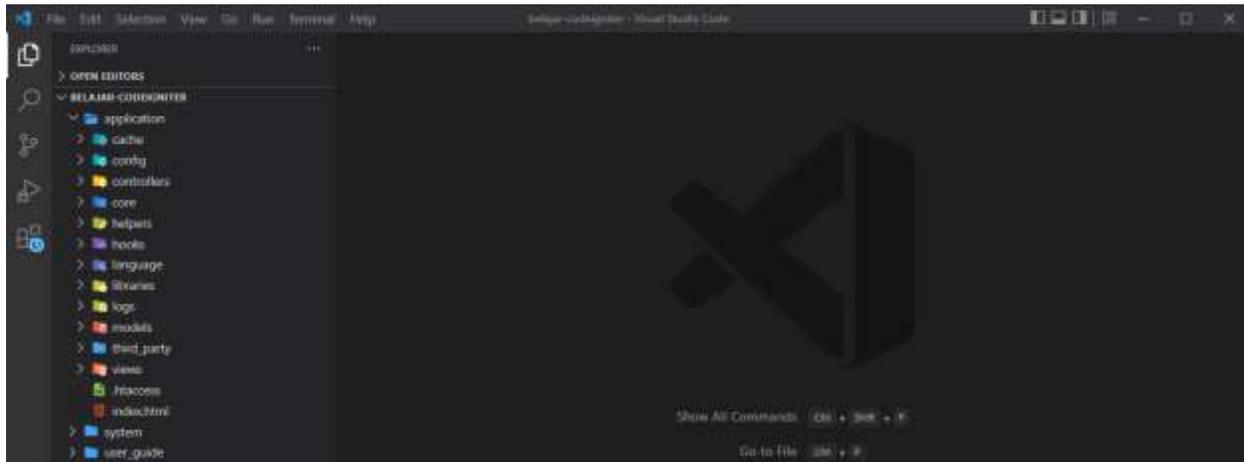


- e. Jika sudah tampil halaman **Welcome Codeigniter**, itu artinya anda telah berhasil melakukan instalasi codeigniter.

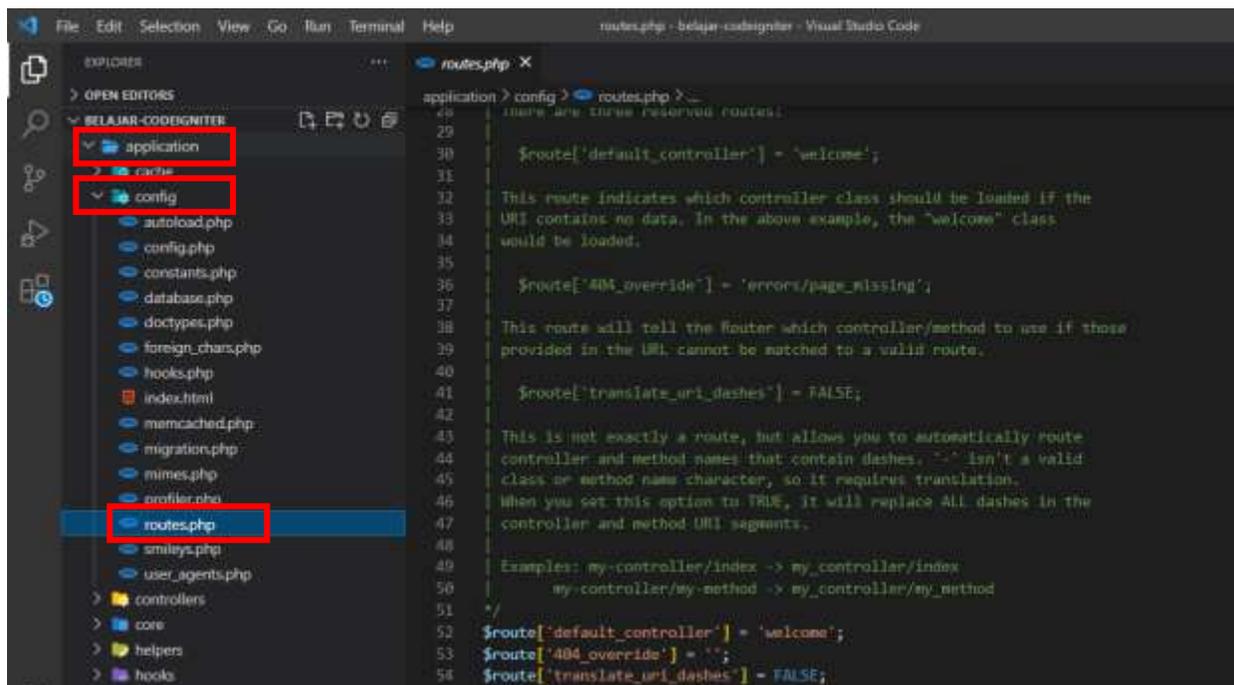
2. Struktur Folder Codeigniter

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

nanti, Folder **system** berisi library atau kumpulan fungsi-fungsi dasar Codeigniter (CI), sedangkan **User Guide** berisi sebagai buku panduan atau dokumentasi dari codeigniter.



Pertama kali kita fokus pada folder **application/config** di dalam folder config ini terdapat file-file konfigurasi utama dan pertama kali untuk memudahkan pembuatan dan pengembangan semua file yang akan dibuat dan semua berawal dari file **routes.php** Apabila kita buka file tersebut, maka kita akan menemukan **default_controller** yang telah terisi dengan nama **“welcome”**, yang mana **default_controller** ini berfungsi ketika user tidak melakukan atau memasukkan apapun pada URL browser, maka secara otomatis mencari controller dengan nama **“welcome”**.



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

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

class Welcome extends CI_Controller {

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

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

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

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

```
<title>Welcome to CodeIgniter</title>

<style type="text/css">
  ::selection {
    background-color: #E13300;
    color: white;
  }

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

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

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

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

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

```

    }

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

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

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

<body>

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

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

            <p>If you would like to edit this page you'll find it located at:</p>
            <code>application/views/welcome_message.php</code>

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

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

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

```

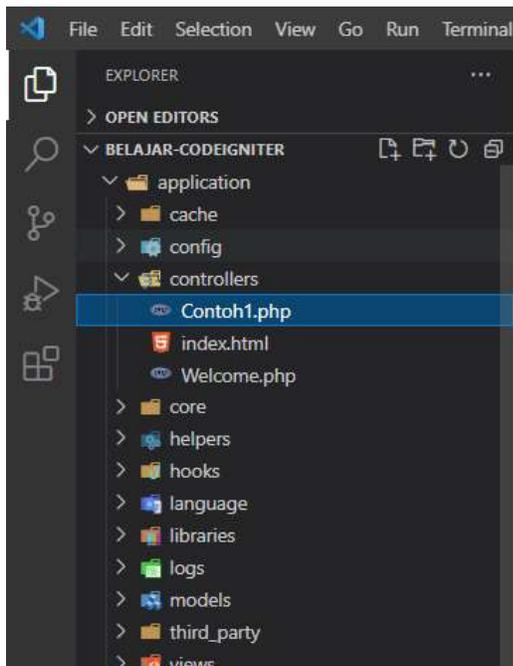
```
</div>

</body>

</html>
```

Contoh menampilkan Biodata

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



```
application > controllers > Contoh01.php > ...
1  <?php
2  class Contoh1 extends CI_Controller
3  {
4      public function index()
5      {
6          echo "<h1>Perkenalkan</h1>";
7          echo "Nama saya Imam Nawawi
8              Saya tinggal di daerah Ciputat
9              olah raga yang saya sukai adalah
10             Bulutangkis";
11     }
12 }
13
```

Untuk melihat hasilnya silahkan ketik URL berikut **localhost/belajar-codeigniter/index.php/contoh1**



Perkenalkan

Nama saya Imam Nawawi Saya tinggal di daerah Ciputat olah raga yang saya sukai adalah Bulutangkis

MINGGU KE 7

Model View Controller

Deskripsi:

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

Tujuan Pembelajaran:

Setelah melakukan bagian ini pembaca mampu:

1. Memahami konsep dan aturan penggunaan MVC
2. Menentukan controller standar (default)
3. memahami cara menghilangkan index.php
4. memahami dan menggunakan script parsing data

A. Pengertian Model View Controller (MVC)

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

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

a. Model

Model adalah kelas yang merepresentasikan atau memodelkan tipe data yang akan digunakan oleh aplikasi. Model juga dapat didefinisikan sebagai bagian penanganan yang berhubungan dengan pengolahan atau manipulasi database. seperti misalnya mengambil data dari database, menginput dan pengolahan database lainnya. semua

intruksi atau fungsi yang berhubungan dengan pengolahan database di letakkan di dalam model. Sebagai contoh, jika ingin membuat aplikasi untuk menghitung luas dan keliling lingkaran, maka dapat memodelkan objek lingkaran sebagai kelas model.

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

b. View

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

c. Controller

Controller merupakan kumpulan instruksi 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 instruksi.

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

Bentuk umum model seperti berikut:

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

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

```

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

```

Contoh1 penggunaan hanya controller

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

```

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

```

Untuk melihat hasilnya ketikkan url **localhost/belajar-codeigniter/index.php/latihan1**

Contoh2 menggunakan Controller dan Model

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

```

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

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

```

```
        return $this->hasil;
    }
}
```

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

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

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

Untuk melihat hasilnya ketikkan url

localhost/belajar-codeigniter/index.php/latihan1/penjumlahan/2/6

Contoh3 menggunakan Controller, View dan Model

View **view-latihan1.php** Simpan di **application\views**

```
<html>

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

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

```
</body>
</html>
```

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

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

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

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

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

Untuk melihat hasilnya ketikkan url

localhost/belajar-codeigniter/index.php/latihan1/penjumlahan/2/6

B. Menentukan Kontroler Standar (Default)

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

Contoh penulisan URI:

<http://localhost/belajar-codeigniter/index.php>

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

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

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

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

C. Menghilangkan index.php pada codeigniter

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

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

Kemudian hapus index.php pada pengaturan sehingga menjadi

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

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

File **belajar-codeigniter/.htaccess**

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

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

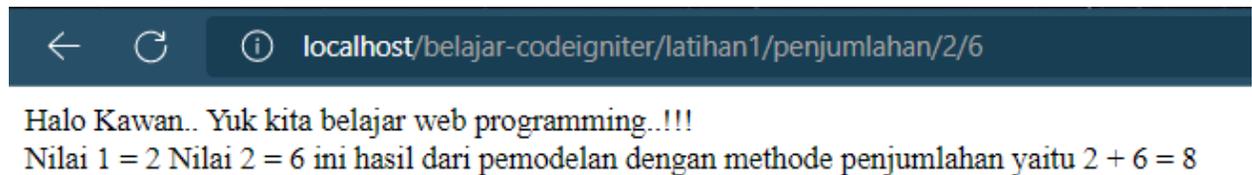
Jika biasanya untuk mengakses method dalam kontroler Latihan1 diatas harus dengan mengetikkan

```
http://localhost/ belajar-codeigniter/index.php/latihan1/penjumlahan/4/6
```

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

```
http://localhost/ belajar-codeigniter/ latihan1/penjumlahan/4/6
```

berikut hasil dalam browser



D. Aturan merancang Model, View, dan Controller

1. Model

Secara umum perancangan model dapat dibuat menggunakan aturan berikut:

- Model harus memiliki atribut yang dapat mewakili element tertentu.
- Model seharusnya berisi kode kode yang menangani proses bisnis untuk data bersangkutan, misalnya untuk melakukan proses validasi data.
- Model boleh berisi kode kode yang digunakan untuk memanipulasi data, misalnya untuk menambah baris data baru, mengubah baris data, dan menghapus baris data pada sebuah tabel yang terdapat dalam database.

- d. Model seharusnya tidak berisi kode kode yang berkaitan langsung dengan permintaan yang dilakukan oleh user melalui URL (tidak berisi `$_GET` maupun `$_POST`). Pekerjaan seperti ini seharusnya dilakukan oleh controller.
- e. Model seharusnya tidak berisi kode kode yang berkaitan dengan presentasi (tampilan) halaman web yang akan disajikan ke hadapan user. Pekerjaan semacam ini seharusnya dilakukan oleh view.

2. View

Berikut ini aturan yang dapat digunakan untuk merancang view.

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

3. Controller

Berikut ini aturan yang dapat digunakan untuk merancang controller.

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

E. Parsing Data (mengirimkan nilai) dari controller Ke View

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

Perhatikan contoh cara pengiriman nilai ke view codeigniter berikut ini.

Contoh5 mengirimkan nilai dari controller ke view

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

```
<html>

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

<body>
  <center>
    <form action="<?='http://localhost/belajar-
codeigniter/matakuliah/cetak'; ?>" method="post">
      <table>
        <tr>
          <th colspan="3">
            Form Input Data Mata Kuliah
          </th>
        </tr>
        <tr>
          <td colspan="3">
            <hr>
          </td>
        </tr>
        <tr>
          <th>Kode MTK</th>
          <th>:</th>
          <td>
            <input type="text" name="kode" id="kode">
          </td>
        </tr>
        <tr>
          <th>Nama MTK</th>
          <td>:</td>
          <td>
```

```

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

```

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

```

<?php
class Matakuliah extends CI_Controller
{
    public function index()
    {
        $this->load->view('view-form-matakuliah');
    }
}

```

```

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

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

```

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

```

<html>
<head>
    <title>Tampil Data Matakuliah</title>
</head>
<body>
    <center>
        <table>
            <tr>
                <th colspan="3">
                    Tampil Data Mata Kuliah
                </th>
            </tr>
            <tr>
                <td colspan="3">
                    <hr>
                </td>
            </tr>
            <tr>
                <th>Kode MTK</th>
                <th>:</th>
            </tr>

```

```

        <td>
            <?= $kode; ?>
        </td>
    </tr>
    <tr>
        <td>Nama MTK</td>
        <td>:</td>
        <td>
            <?= $nama; ?>
        </td>
    </tr>
    <tr>
        <td>SKS</td>
        <td>:</td>
        <td>
            <?= $sks; ?>
        </td>
    </tr>
    <tr>
        <td colspan="3" align="center">
            <a href="<?= 'http://localhost/belajar-
codeigniter/matakuliah'; ?>">Kembali</a>
        </td>
    </tr>
</table>
</center>
</body>
</html>

```

MINGGU KE 9

Membuat Template Web Sederhana

Deskripsi:

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

Tujuan Pembelajaran:

Setelah melakukan bagian ini pembaca mampu:

1. memahami dan membuat template web sederhana
2. memahami dan mengatur base_url()

A. Membuat Template Sederhana yang dinamis

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

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

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

class Web extends CI_Controller{

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

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

```
}
```

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

v_header.php

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

v_index.php

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

```

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

```

V_footer.php

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

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

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

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

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

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

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

```

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

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

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

.clear{
    clear: both;
}

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

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

section{
    padding: 20px;
}

```

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

B. Seting base_url () pada codeigniter

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

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

Kemudian diubah menjadi

```
$config['base_url'] = 'http://localhost/belajar-codeigniter/';
```

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

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

Hasil perintah di atas akan sama seperti ini

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

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

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

class Web extends CI_Controller{

    function __construct(){
        parent::__construct();
        $this->load->helper('url');
    }

    public function index(){
        $data['judul'] = "Halaman Depan";
```

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

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

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

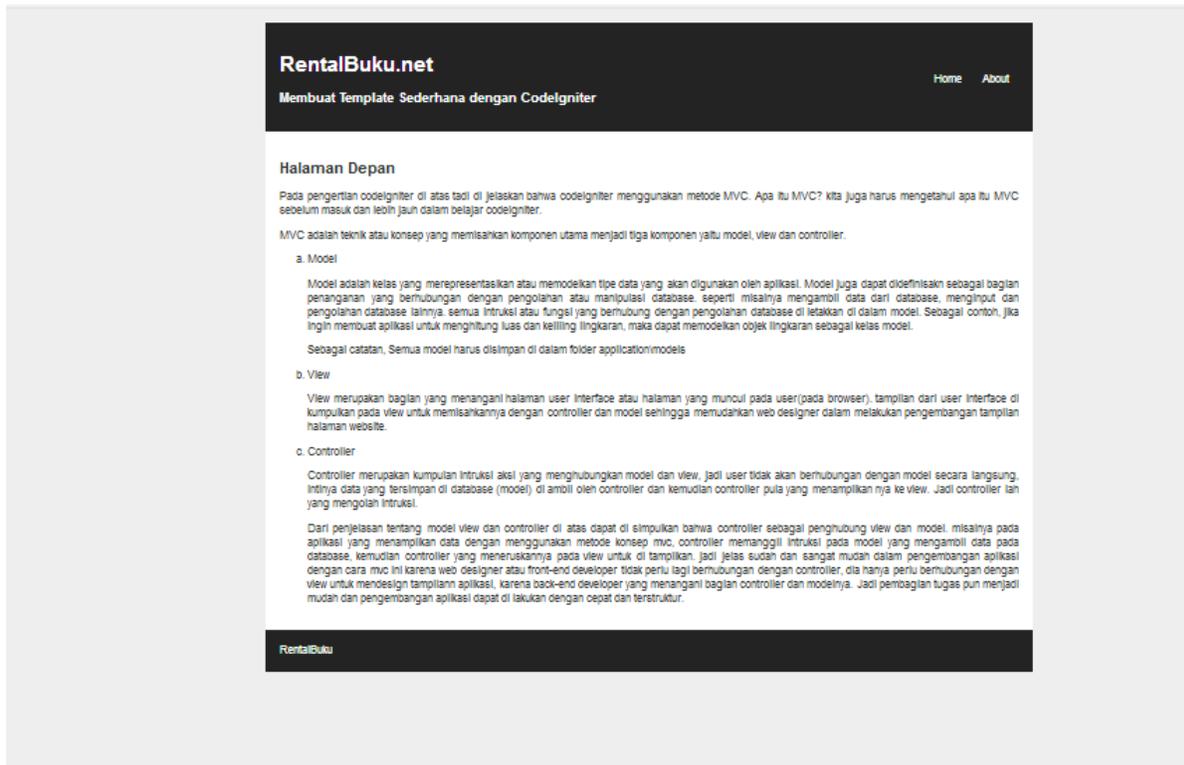
Kemudian diubah dengan ditambahkan kata 'url' menjadi seperti di bawah

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

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

<http://localhost/belajar-codeigniter/web>

hasilnya seperti gambar di bawah.



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

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

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

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

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

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

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

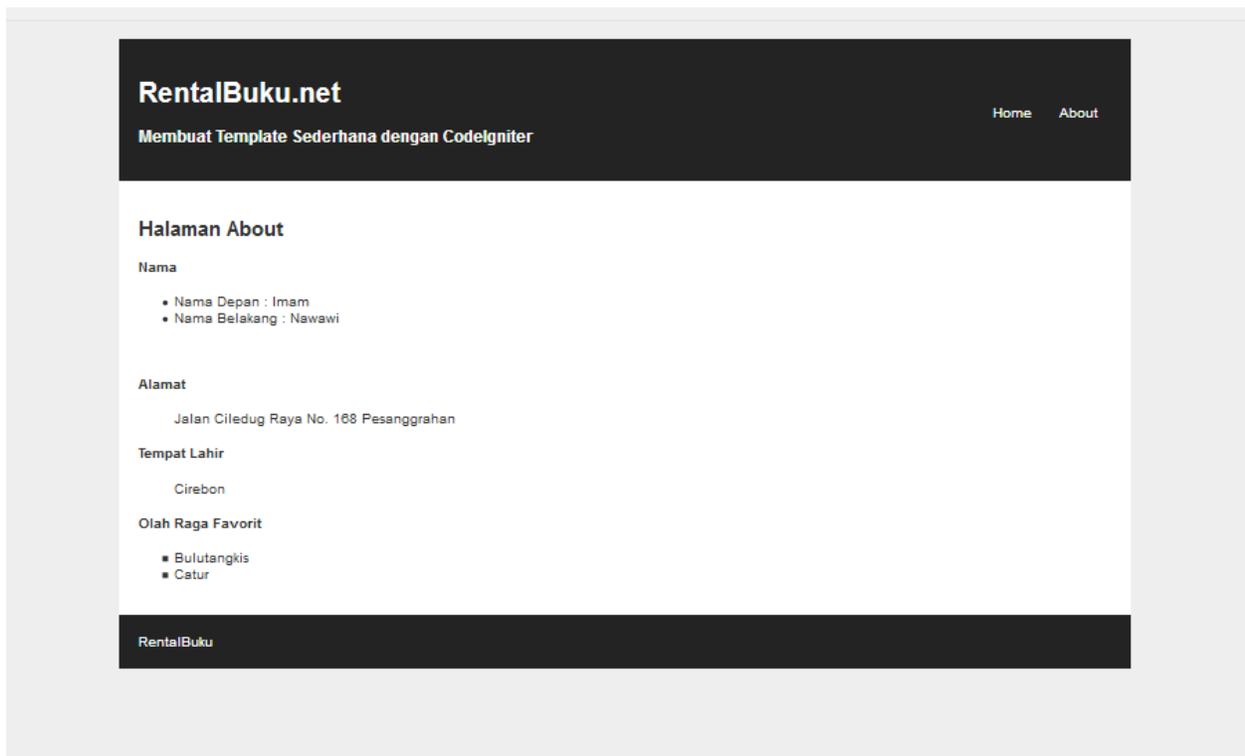
```
</ul>  
</section>
```

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

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

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

<http://localhost/belajar-codeigniter/web/about>



MINGGU KE 10

Membuat Form Validasi

Deskripsi:

Pertemuan ini membahas tentang pembuatan validasi untuk memasukkan data ke dalam database dengan menggunakan library `form_validasi`.

Tujuan Pembelajaran:

Setelah melakukan bagian ini pembaca mampu:

1. memahami cara menggunakan form validasi
2. memahami mekanisme validasi data dalam form

A. Form Validasi

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

Pembuatan form validasi pada codeigniter diperlukan adanya pemanggilan library **`form_validation`** pada codeigniter.

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

B. Mekanisme Validasi Data di dalam Form

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

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

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

C. Memuat Library Form_validation

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

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

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

Buka file tersebut lalu cari baris sintak seperti berikut:

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

Kemudian diubah dengan ditambahkan kata form_validation seperti berikut:

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

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

```
<?php
class Matakuliah extends CI_Controller
{
    public function index()
    {
        $this->load->view('view-form-matakuliah');
    }

    public function cetak()
    {
        $this->form_validation->set_rules('kode', 'Kode Matakuliah',
        'required|min_length[3]', [
```

```

        'required' => 'Kode Matakuliah Harus diisi',
        'min_lenght' => 'Kode terlalu pendek'
    ]);

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

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

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

```

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

Buka file view view-form-matakuliah.php, dan ubah script dengan menambahkan script `<?= form_error(.....); ?>` disetiap inputan yang sudah diberi aturan validasi seperti dibawah

```

<tr>
    <th>Kode MTK</th>
    <th>:</th>
    <td>
        <input type="text" name="kode" id="kode">
        <?= form_error('kode'); ?>
    </td>
</tr>
<tr>
    <th>Nama MTK</th>
    <td>:</td>
    <td>
        <input type="text" name="nama" id="nama">
        <?= form_error('nama'); ?>
    </td>
</tr>
<tr>

```

Kondisi jika inputan tidak valid karena kosong

Form Input Data Mata Kuliah

Kode MTK :
Kode Matakuliah Harus diisi

Nama MTK :
Nama Matakuliah Harus diisi

SKS :

Kondisi jika inputan tidak valid karena terlalu pendek (minimal 3 karakter)

<hr/> <p>Form Input Data Mata Kuliah</p> <hr/> <p>Kode MTK : <input type="text" value="1"/> Kode terlalu pendek</p> <p>Nama MTK : <input type="text" value="a"/> Nama terlalu pendek</p> <p>SKS : <input type="text" value="Pilih SKS"/> <input type="button" value="Submit"/></p>	<hr/> <p>Form Input Data Mata Kuliah</p> <hr/> <p>Kode MTK : <input type="text"/> Kode terlalu pendek</p> <p>Nama MTK : <input type="text"/> Nama terlalu pendek</p> <p>SKS : <input type="text" value="Pilih SKS"/> <input type="button" value="Submit"/></p>
--	--

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

Tugas Pertemuan 10

Untuk mengetahui sejauh mana pemahaman yang sudah anda dapat tentang materi dasar MVC dari pertemuan 6 sampai pertemuan 10, maka di pertemuan ini dibuatlah sebuah studi kasus yang harus anda selesaikan. Dengan konsep MVC yang sudah dijelaskan di pertemuan 6 s/d 10. Dari membuat tampilan input sampai menampilkan hasil submit dari tampilan input tersebut. Berikut studi kasis yang harus anda selesaikan:

1. Ada sebuah toko sepatu yang ingin dibuatkan program untuk menghitung otomatis dari transaksi penjualan sepatunya yang mana harga sepatu ditentukan berdasarkan merk dari sepatu tersebut.

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

Form Pembelian Sepatu

Nama Pembeli

No Telp/HP

Merk Sepatu

Ukuran Sepatu

Jumlah Beli

Struk Pembelian Sepatu [Kembali ke Form](#)

Nama Pembeli	:	Maruloh
No telp/HP	:	081511112222
Merk Sepatu	:	Adidas
Ukuran Sepatu	:	43
Harga	:	300000
Jumlah Beli	:	2
Total Harga	:	600000

MINGGU KE 11

Membuat Program Create Read Update Delete (CRUD)

Deskripsi:

Pada pertemuan ini dibahas bagaimana membuat program database dengan menggunakan framework codeigniter, yaitu Create, Read, Update, Delete. Akan tetapi khusus pertemuan ini materi terbatas hanya pada penyampaian perintah Create dan Read data. Untuk bahasan perintah Update dan delete akan dibahas dipertemuan terpisah, yaitu pertemuan 12

Tujuan Pembelajaran:

Setelah melakukan bagian ini pembaca mampu:

1. memahami penggunaan database dalam pemrograman
2. membuat halaman web sederhana untuk menyimpan dan menampilkan data

A. Pembuatan Database

Buatlah sebuah database dengan nama Kampus dan buatlah tabel didalamnya dan beri nama mahasiswa dengan struktur tabel sebagai berikut:

#	Nama	Jenis
1	nim 🗝️	varchar(8)
2	nama	varchar(30)
3	alamat	varchar(30)
4	telepon	varchar(15)

Isi tabel mahasiswa sebanyak 3 data

nim	nama	alamat	telepon
11200011	Azzahra Asyila Rahma	Cengkareng Jakarta Barat	085609876789
11200012	Abizar Azka Rafasya	Cimone Tangerang	081309876543
12200033	Firman	Jembatan Tiga Jakarta Barat	088888881111

B. Membuat Koneksi Database Codeigniter dengan MySQL

Sebelum melakukan seting konfigurasi pada file database, terlebih dahulu harus disiapkan databasenya. Untuk menghubungkan codeigniter dengan database buka file config database codeigniter yang terletak di **application/config/database.php** kemudian cari baris sintak seperti berikut:

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

Masukkan user dan nama database nya, untuk password jika tidak ada maka dikosongkan saja, sehingga menjadi berikut

```
$db['default'] = array(  
    'dsn' => '',  
    'hostname' => 'localhost',  
    'username' => 'root',  
    'password' => '',  
    'database' => 'kampus',  
    '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  
);
```

Daftarkan libraries database pada file autoload.php agar database yang kita punya bisa digunakan.

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

C. Menampilkan Data Mahasiswa

Membuat Model untuk Proses CRUD

Seperti dijelaskan sebelumnya, pada konsep MVC, kita menggunakan model untuk menggunakan data. Oleh karena itu kita akan membuat sebuah model dimana didalamnya nanti akan ada fungsi-fungsi pemanggilan dan pengolahan data, seperti simpan, update, dan hapus.

Buat sebuah file pada bagian model dengan nama **ModelMahasiswa.php** yang berisi perintah berikut

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

class ModelMahasiswa extends CI_Model
{
    public function tampilData()
    {
        return $this->db->get('mahasiswa');
    }
}
```

Dengan model seperti di atas, untuk menampilkan data atau mengambil data dari database kita bisa menggunakan function tampilData().

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

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

Ubah menjadi berikut:

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

Selanjutnya kita membuat controller yang berisi parsing data yang kita ambil dari model ke view. Beri nama controller dengan nama **Mahasiswa.php** yang berisi kode berikut:

```

<?php
class Mahasiswa extends CI_Controller
{
    function index()
    {
        $data['mahasiswa'] = $this->ModelMahasiswa->tampilData()->result();
        $this->load->view('data-mahasiswa', $data);
    }
}

```

Pada controller Mahasiswa.php tersebut kita memanggil view dengan nama data-mahasiswa. Oleh karena itu untuk menampilkan data mahasiswa dalam halaman web kita buat sebuah file view dengan nama **data-mahasiswa.php**, dengan script berikut:

```

<!DOCTYPE html>
<html>

<head>
    <title>Membuat CRUD dengan CodeIgniter</title>
</head>

<body>
    <center>
        <h1>DATA MAHASISWA</h1>
    </center>
    <center><?php echo anchor('mahasiswa/tambah', 'Tambah Data'); ?></center>
    <table style="margin:20px auto;" border="1">
        <tr>
            <th>No</th>
            <th>NIM</th>
            <th>Nama</th>
            <th>Alamat</th>
            <th>No.Telepon</th>
            <th>Action</th>
        </tr>
        <?php
        $no = 1;
        foreach ($mahasiswa as $m) {
            ?>
            <tr>
                <td><?php echo $no++ ?></td>
                <td><?php echo $m->nim ?></td>
                <td><?php echo $m->nama ?></td>
                <td><?php echo $m->alamat ?></td>
                <td><?php echo $m->telepon ?></td>

```

```

        <td>
            <?php echo anchor('mahasiswa/edit/' . $m->nim, 'Edit'); ?>
            <?php echo anchor('mahasiswa/hapus/' . $m->nim, 'Hapus'); ?>
        </td>
    </tr>
    <?php } ?>
</table>
</body>
</html>

```

Hasil dapat dilihat dengan url **localhost/belajar-codeigniter/mahasiswa**

DATA MAHASISWA

[Tambah Data](#)

No	NIM	Nama	Alamat	No.Telepon	Action
1	11200011	Azzahra Asyila Rahma	Cengkareng Jakarta Barat	085609876789	Edit Hapus
2	11200012	Abizar Azka Rafasya	Cimone Tangerang	081309876543	Edit Hapus
3	12200033	Firman	Jembatan Tiga Jakarta Barat	088888881111	Edit Hapus

D. Menyimpan Data

Sebelumnya untuk menyimpan data, kita tambahkan terlebih dahulu fungsi simpan data pada model ModelMahasiswa untuk melakukan penyimpanan

```

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

class ModelMahasiswa extends CI_Model
{
    public function tampilData()
    {
        return $this->db->get('mahasiswa');
    }

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

```

Untuk menyimpan data, kita akan membuat fungsi sesuai dengan pendeklarasian di file view sebelumnya

DATA MAHASISWA

[Tambah Data](#)

Nama	Alamat	No.tel
ra Asyila Rahma	Cengkareng Jakarta Barat	0856098
r Azka Rafasya	Cimone Tangerang	0813098
n	Jembatan Tiga Jakarta Barat	0888888

```
<center>
  <h1>DATA MAHASISWA</h1>
</center>
<center><?php echo anchor('mahasiswa/tambah', 'Tambah Data'); ?></center>
```

Dari script view tersebut, kita akan membuat sebuah fungsi dengan nama **tambah()** di controller Mahasiswa.php.

```
<?php
class Mahasiswa extends CI_Controller
{
    function index()
    {
        $data['mahasiswa'] = $this->ModelMahasiswa->tampilData()->result();
        $this->load->view('data-mahasiswa', $data);
    }

    function tambah()
    {
        $this->load->view('input-mahasiswa');
    }
}
```

Selanjutnya kita akan membuat sebuah file pada bagian view dengan nama **input-mahasiswa.php**. file ini berfungsi sebagai tempat untuk mengisi data melalui form dalam web. Berikut script **input-mahasiswa.php**

```
<!DOCTYPE html>
<html>

<head>
  <title>Membuat CRUD dengan CodeIgniter</title>
</head>

<body>
  <center>
    <h3>Tambah data baru</h3>
  </center>
  <form action="<?php echo base_url() . 'mahasiswa/tambah_aksi'; ?>"
method="post">
  <table style="margin:20px auto;">
```

```

        <tr>
            <td>NIM</td>
            <td><input type="text" name="nim"></td>
        </tr>
        <tr>
        <tr>
            <td>Nama</td>
            <td><input type="text" name="nama"></td>
        </tr>
        <tr>
            <td>Alamat</td>
            <td><input type="text" name="alamat"></td>
        </tr>
        <tr>
            <td>No.Telepon</td>
            <td><input type="text" name="telepon"></td>
        </tr>
        <tr>
            <td></td>
            <td><input type="submit" value="Tambah"></td>
        </tr>
    </table>
</form>
</body>

</html>

```

Ketika link Tampil data di klik akan menampilkan form inputan seperti yang kita buat

DATA MAHASISWA

[Tambah Data](#)

Tambah data baru	
NIM	<input type="text"/>
Nama	<input type="text"/>
Alamat	<input type="text"/>
No.Telepon	<input type="text"/>
	<input type="submit" value="Tambah"/>

Kita kembali ke script view input-mahasiswa.php. pada script tersebut, data-data yang kita input akan dikirim menggunakan metode POST ke controller Mahasiswa dan fungsi tambah_aksi.

```
<form action="<?php echo base_url() . 'mahasiswa/tambah_aksi'; ?>" method="post">
```

Oleh karena itu kita akan membuat sebuah fungsi dengan nama **tambah_aksi()** pada controller **Mahasiswa.php**. dimana fungsinya adalah menerima data yang dikirim oleh form

inputan, untuk selanjutnya akan disimpan di tabel mahasiswa. Berikut merupakan script dari fungsi `tambah_aksi()`

```
function tambah_aksi()
{
    $nim = $this->input->post('nim');
    $nama = $this->input->post('nama');
    $alamat = $this->input->post('alamat');
    $telepon = $this->input->post('telepon');

    $data = array(
        'nim' => $nim,
        'nama' => $nama,
        'alamat' => $alamat,
        'telepon' => $telepon
    );
    $this->ModelMahasiswa->simpanMahasiswa($data);
    redirect('mahasiswa/index');
}
```

Hasil:

DATA MAHASISWA

[Tambah Data](#)

Tambah data baru

NIM

Nama

Alamat

No. Telepon

No	NIM	Nama	Alamat	No.Telepon	Action
1	11200011	Azzahra Asyila Rahma	Cengkareng Jakarta Barat	085609876789	Edit Hapus
2	11200012	Abizar Azka Rafasya	Cimone Tangerang	081309876543	Edit Hapus
3	11200909	Wiro	Tegal	0987890989	Edit Hapus
4	12068493	Maruloh	jakarta	765456765	Edit Hapus
5	12200033	Firman	Jembatan Tiga Jakarta Barat	088888881111	Edit Hapus

MINGGU KE 12

Membuat Program Create Read Update Delete

Deskripsi:

Pada pertemuan ini meneruskan bahasan CRUD, jika dipertemuan sebelumnya telah dibahas perintah menampilkan data dan menyimpan data, maka pertemuan ini membahas perintah Update dan hapus data pada codeigniter

Tujuan Pembelajaran:

Setelah melakukan bagian ini pembaca mampu:

1. membuat halaman web sederhana untuk mengupdate data
2. membuat halaman web sederhana untuk menghapus data

A. Modifikasi ModelMahasiswa

Karena berhubungan dengan data, maka kita akan memodifikasi ModelMahasiswa.php yang sebelumnya sudah memiliki fungsi tampil dan simpan data, dengan menambahkan fungsi editData, updateMahasiswa dan hapusMahasiswa

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

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

function hapus_data($where, $table)
{
    $this->db->where($where);
    $this->db->delete($table);
}
```

B. Memperbaharui Data Mahasiswa

Kita kembali ke form view mahasiswa, terdapat 2 link di bagian aksi yaitu Edit dan Hapus, dengan script berikut:

```

<td>
    <?php echo anchor('mahasiswa/edit/' . $m->nim, 'Edit'); ?>
    <?php echo anchor('mahasiswa/hapus/' . $m->nim, 'Hapus'); ?>
</td>

```

Dari script diatas kita bisa tahu bahwa saat link Edit diklik, maka dia akan mengarahkan ke controller Mahasiswa dan fungsi edit, dengan mengirim nilai NIM sesuai dengan data yang dipilih, begitupun dengan link Hapus, akan mengarahkan ke controller Mahasiswa dan fungsi hapus, dengan mengirimkan nilai NIM sesuai dengan data yang dipilih. Oleh karena itu kita akan membuat sebuah fungsi dengan nama **edit di controller Mahasiswa.php**

```

function edit($id)
{
    $where = array('nim' => $id);
    $data['mahasiswa'] = $this->ModelMahasiswa->editData($where, 'mahasiswa')->result();
    $this->load->view('edit-mahasiswa', $data);
}

```

Fungsi edit() memiliki kegunaan untuk menyeleksi data berdasarkan NIM dan menampilkannya ke view edit-mahasiswa.php. Jadi setelah ini kita akan membuat sebuah file View untuk menampilkan data yang akan kita edit. Beri nama file tersebut dengan **edit-mahasiswa.php**

```

<!DOCTYPE html>
<html>
<head>
    <title>Membuat CRUD dengan CodeIgniter</title>
</head>
<body>
    <center>
        <h3>Edit Data</h3>
    </center>
    <?php foreach ($mahasiswa as $m) { ?>
        <form action="<?php echo base_url() . 'mahasiswa/update'; ?>"
method="post">
            <table style="margin:20px auto;">
                <tr>
                    <td>Nama</td>
                    <td>
                        <input type="hidden" name="nim" value="<?php echo $m->nim
?>">
                        <input type="text" name="nama" value="<?php echo $m->nama
?>">

```

```

        </td>
    </tr>
    <tr>
        <td>Alamat</td>
        <td><input type="text" name="alamat" value="<?php echo $m-
>alamat ?>"></td>
    </tr>
    <tr>
        <td>No.Telepon</td>
        <td><input type="text" name="telepon" value="<?php echo $m-
>telepon ?>"></td>
    </tr>
    <tr>
        <td></td>
        <td><input type="submit" value="Simpan"></td>
    </tr>
</table>
</form>
<?php } ?>
</body>
</html>

```

Hasil:

DATA MAHASISWA

[Tambah Data](#)

No	NIM	Nama	Alamat	No.Telepon	Action
1	11200011	Azzahra Asyila Rahma	Cengkareng Jakarta Barat	085609876789	Edit Hapus
2	11200012	Abizar Azka Rafasya	Cumone Tangerang	081309876543	Edit Hapus
3	11200909	Wiro	Tegal	0987890989	Edit Hapus
4	12068493	Maruloh	jakarta	765456765	Edit Hapus
5	12200033	Firman	Jembatan Tiga Jakarta Barat	088888881111	Edit Hapus

Edit Data

Nama

Alamat

No.Telepon

Data-data tersebut akan dikirimkan ke fungsi **update()** pada controller Mahasiswa.php dengan metode POST, jadi langkah selanjutnya kita akan membuat fungsi **update()** untuk menampung data sekaligus menyimpan data yg terbaru

```
function update()
{
    $nim = $this->input->post('nim');
    $nama = $this->input->post('nama');
    $alamat = $this->input->post('alamat');
    $telepon = $this->input->post('telepon');

    $data = array(
        'nama' => $nama,
        'alamat' => $alamat,
        'telepon' => $telepon
    );

    $where = array(
        'nim' => $nim
    );

    $this->ModelMahasiswa->updateMahasiswa($where, $data, 'mahasiswa');
    redirect('mahasiswa/index');
}
```

Pada script diatas saat proses update dilakukan maka akan diarahkan ke halaman index pada controller mahasiswa (halaman tampil data)

DATA MAHASISWA

Edit Data

Nama

Alamat

No.Telepon

[Tambah Data](#)

No	NIM	Nama	Alamat	No.Telepon	Action
1	11200011	Azzahra Asyila Rahma	Cengkareng Jakarta Barat	085609876789	Edit Hapus
2	11200012	Abizar Azka Rafasya	Cimone Tangerang	081309876543	Edit Hapus
3	11200909	Wiro Waras	Tegal	0987890989	Edit Hapus
4	12068493	Maruloh	jakarta	765456765	Edit Hapus
5	12200033	Firman	Jembatan Tiga Jakarta Barat	088888881111	Edit Hapus

C. Menghapus Data Mahasiswa

Kita kembali ke form view mahasiswa. Perhatikan script untuk hapus

```
<td>
    <?php echo anchor('mahasiswa/edit/' . $m->nim, 'Edit'); ?>
    <?php echo anchor('mahasiswa/hapus/' . $m->nim, 'Hapus'); ?>
</td>
```

Dari script tersebut kita akan membuat sebuah fungsi dengan nama **hapus()** pada controller **mahasiswa.php**. berikut adalah scriptnya:

```
function hapus($id)
{
    $where = array('nim' => $id);
    $this->ModelMahasiswa->hapusMahasiswa($where, 'mahasiswa');
    redirect(['mahasiswa/index']);
}
```

Script tersebut pada dasarnya sama seperti update, mengirimkan nilai NIM sebagai id (kondisi) yang nantinya akan digunakan untuk menghapus data terpilih. Saat data terpilih untuk dihapus, halaman akan berada di index (halaman data mahasiswa)

DATA MAHASISWA

[Tambah Data](#)

No	NIM	Nama	Alamat	No.Telepon	Action
1	11200011	Azzahra Asyila Rahma	Cengkareng Jakarta Barat	085609876789	Edit Hapus
2	11200012	Abizar Azka Rafasya	Cimone Tangerang	081309876543	Edit Hapus
3	12068493	Maraloh	jakarta	765456765	Edit Hapus
4	12200033	Ferman	Jembatan Tiga Jakarta Barat	088888881111	Edit Hapus

DATA MAHASISWA

[Tambah Data](#)

No	NIM	Nama	Alamat	No.Telepon	Action
1	11200011	Azzahra Asyila Rahma	Cengkareng Jakarta Barat	085609876789	Edit Hapus
2	11200012	Abizar Azka Rafasya	Cimone Tangerang	081309876543	Edit Hapus
3	12200033	Ferman	Jembatan Tiga Jakarta Barat	088888881111	Edit Hapus

MINGGU KE 13-15
Presentasi Project