

2016

Modul Praktikum Pemrograman Web

TEKNIK INFORMATIKA
DESTY RODIAH, S.KOM M.T

DAFTAR ISI

MENGENAL HTML	5
Tujuan Praktikum	5
Pengenalan HTML.....	5
Bentuk Format Dasar Dokumen HTML	5
Tag – Tag dasar HTML	5
Tag untuk Pemformatan Teks	5
Tag List	7
Tabel	9
Link dan Gambar	11
Link	12
Gambar.....	12
Soal	13
DASAR PEMBUATAN WEBSITE DENGAN HTML	15
Tujuan Praktikum	15
Elemen Form <form>	15
Properti Masukan Pada Elemen Form <input>, <textarea>, <select>.....	15
Frame	18
Soal:	20
CSS.....	22
Tujuan Praktikum	22
Pengenalan CSS.....	22
Penulisan CSS	23
SOAL	24
JAVASCRIPT.....	26
Tujuan Praktikum	26
Aplikasi Untuk Menjalankan JavaScript.....	26
Cara Memasukkan JavaScript menggunakan tag <script> (internal JavaScript)	26
Cara Memasukkan JavaScript Menggunakan tag <script src=" "> (external JavaScript)	27
Cara Memasukkan JavaScript Menggunakan Event Handler (Inline JavaScript)	27
LATIHAN: LatihanJSEventHandler.html.....	27
Cara Memasukkan JavaScript Menggunakan URL (href:="javascript:")	28

LATIHAN: LatihanJSURL.html	28
Cara Menampilkan Hasil Program JavaScript	28
Fungsi Alert untuk Menampilkan Hasil Program JavaScript	29
Aturan Dasar Penulisan Kode Program JavaScript	31
Cara Penulisan Komentar dalam JavaScript	31
Aturan Penulisan Tanda Semicolon pada Akhir Baris	32
SOAL	33
PENGENALAN PHP.....	34
Tujuan Praktikum	34
Dasar Teori PHP	34
Cara Kerja PHP	34
DASAR PHP	34
Variabel.....	36
Tipe Data.....	37
Konstanta.....	38
Komentar.....	38
Operator	39
Escape Character	43
STRUKTUR KENDALI DALAM PHP	44
Tujuan Praktikum	44
Struktur Kontrol.....	44
Struktur Kondisi	44
Struktur Perulangan (Looping).....	51
Berinteraksi dengan File	54
Menyimpan File.....	55
Fungsi PHP-MySQL	61
Tujuan Praktikum	61
Pengenalan Fungsi PHP-MySQL	61
MEMBUAT APLIKASI LOGIN DENGAN MENGGUNAKAN BASIS DATA	67
Membuat Form Login.....	67
SOAL	68
MEMBUAT APLIKASI BUKU TAMU DENGAN MENGGUNAKAN BASIS DATA 69	
Membuat Buku Tamu	69
SOAL	71
MEMBUAT APLIKASI SEARCHING DENGAN MENGGUNAKAN BASIS DATA 72	
Membuat Form Searching.....	72
SOAL	73

MEMBUAT APLIKASI POLLING DENGAN MENGGUNAKAN BASIS DATA.....	74
Membuat Jejak Pendapat	74
SOAL	76
CODEIGNITER.....	77
Tujuan Praktikum	77
Pengenalan CodeIgniter	77
Menggunakan CodeIgniter.....	77
LATIHAN:	83
CODEIGNITER DAN DATABASE.....	89
Tujuan Praktikum	89
Manipulasi Data pada Database	89
LATIHAN	92
TEMPLATE BOOTSTRAP	104
Tujuan Praktikum	104
Pengenalan Bootstrap.....	104
Cara Menginstal Bootstrap.....	105
Membuat Table Dengan Bootstrap.....	107
Membuat table border dengan bootstrap.....	110
Membuat Table strip dengan bootstrap.....	112
Membuat table hover dengan bootstrap.....	114
Membuat table responsive dengan bootstrap	114
Tampilan Gambar Dengan Bootstrap	114
Membuat gambar thumbnail dengan bootstrap.....	115
Membuat gambar circle dengan bootstrap	116
Membuat gambar round dengan bootstrap	117
Membuat gambar responsive dengan bootstrap	117
Membuat Tombol Dengan Bootstrap	117
Membuat Pesan Alert Dengan Bootstrap.....	120
Membuat Tombol Close Pada Pesan Alert Bootstrap	121
TEMPLATE DAN CODEIGNITER	124
Tujuan Praktikum	124
Pendahuluan	124
TUTORIAL MEMBUAT CRUD CODEIGNITER OTOMATIS DENGAN GROCERY	128
Tujuan Praktikum	128
Pendahuluan Grocery Crud	128
Integrasi Grocery CRUD Dengan Codeigniter	128

Konfigurasi Dasar Codeigniter.....	128
Membuat Database Dan Tabel	129
Cara Menggunakan GROCERY CRUD	130
Menampilkan Custom Field	132
Menggunakan Validation	133
Menampilkan Data Dari tabel Yang Berelasi	134
Upload File/ Upload Field.....	136
Menambahkan Button Action Pada Halaman Utama.....	138
Fitur Callback GROCERY	139
Kesimpulan.....	142
REFERENSI	143

MENGENAL HTML

Tujuan Praktikum

1. Memahami beberapa tag-tag dasar HTML
2. Mampu mengimplementasikannya dalam script-script halaman HTML

Pengenalan HTML

HTML kependekan dari Hyper Text Markup Langguage. Dokumen HTML adalah file teks murni yang dapat dibuat dengan editor teks sembarang. Dokumen ini dikenal sebagai web page. Dokumen HTML merupakan dokumen yang disajikan dalam browser web surfer. Dokumen ini umumnya berisi informasi atau interface aplikasi di dalam Internet.

Ada dua cara untuk membuat sebuah web page: dengan HTML editor atau dengan editor teks biasa (misalnya notepad). Untuk latihan atau mencoba materi sebaiknya menggunakan notepad.

Bentuk Format Dasar Dokumen HTML

```
<HTML>
  <head>
    <title>judul halaman </title>
  </head>
  <body>
    Isi document yang akanditampilkan di browser
  </body>
</HTML>
```

Tag – Tag dasar HTML

Start Tag	Kegunaan
<html>	Mendefinisikan sebuah dokumen html
<body>	Mendefinisikan isi/badan suatu dokumen
<h1> - <h6>	Mendefinisikan heading ke 1 s/d heading ke 6
<p>	Mendefinisikan sebuah paragraph
 	Menyisipkan sebuah line break
<hr>	Mendefinisikan sebuah garis horizontal
<!-->	Mendefinisikan komentar dalam kode sumber HTML

Tag untuk Pemformatan Teks

Start Tag	Kegunaan
	Mendefinisikan teks tebal
	Mendefinisikan teks yang ditekankan
<i>	Mendefinisikan teks yang dimiringkan
<sub>	Subscript
<sup>	Superscript
<u>	Mendefinisikan teks yang bergaris bawah
<pre>	Mendefinisikan teks preformatted

Latihan 1 : Heading.html

```

<html>
    <head><title> Halaman Pertamaku</title></head>
    <body>
        <h1>ini adalah heading 1</h1>
        <h2><font color=" blue">ini adalah heading 2</font></h2>
        <h3>ini adalah heading 3</h3>
        <h4>ini adalah heading 4</h4>
        <h2>ini adalah heading 2 </h2>
        <h1>ini adalah heading 1</h1>
        <p>ini adalah paragraph akan ditampilkan terus menerus secara otomatis
        terpengaruh browser anda tidak
        terpengaruh                                         spasi
        otomatis akan teratur sendiri<br>
        ganti baris<br>ganti baris <br>ganti baris. <br>
        <hr>
        <!-- Komentar tidak ditampilkan di browser-->
    </body>
</html>

```

HASIL:



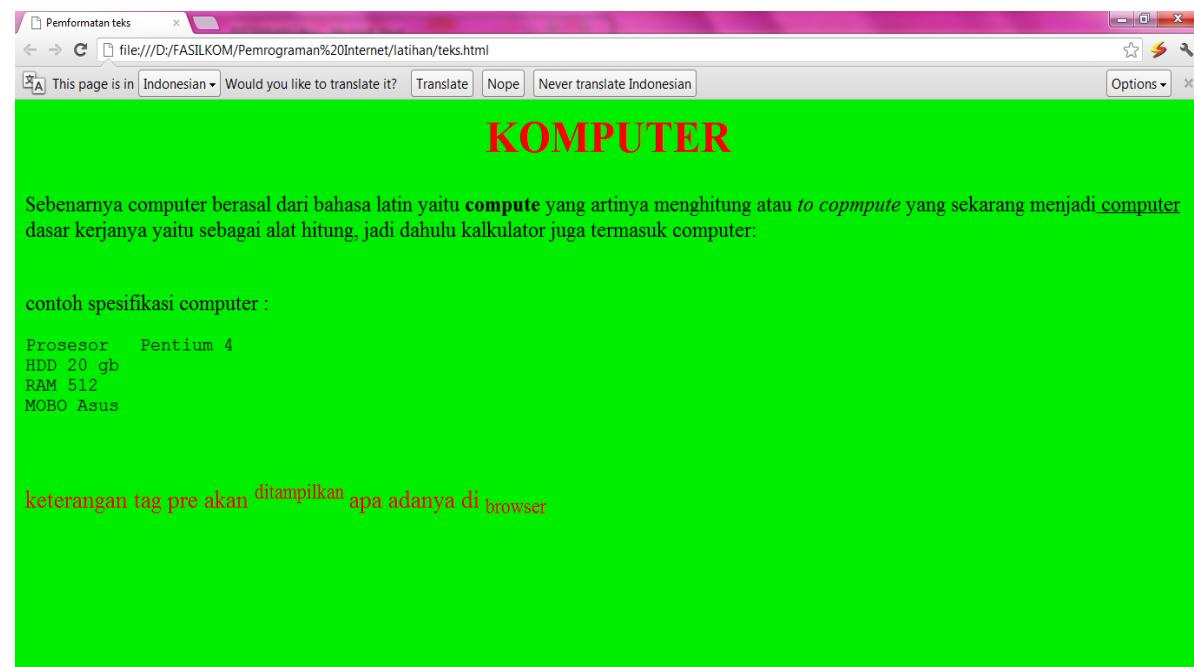
ANALISIS :

Latihan 2 : Latihan 2.html

```
<html>
    <head><title>Pemformatan teks</title></head>
    <body bgcolor=" green">
        <h1 align="center"> <font color="red">KOMPUTER</font></h1>
        <p>Sebenarnya computer berasal dari bahasa latin yaitu <b>compute</b>
        yang artinya menghitung atau <i> to compute</i> yang sekarang
        menjadi<u>
            computer</u> dasar kerjanya yaitu sebagai alat hitung, jadi dahulu
            kalkulator juga termasuk computer:<br> <br></p>
            contoh spesifikasi computer: <br>

            <pre>
                Prosesor      Pentium 4
                HDD 20 gb
                RAM 512
                MOBO Asus </pre><br>
                <p>      <font color="red" size="4">keterangan      tag      pre      akan
                <sup>ditampilkan </sup>apa
                adanya di <sub>browser</sub></font>
            </pre>
        </body>
</html>
```

HASIL :



ANALISIS :

Tag List

Start Tag	Kegunaan
	Mendefinisikan sebuah list ordered

	Mendefinisikan sebuah list unorder
	Mendefinisikan sebuah item dalam list
	Mendefinisikan sebuah list definisi

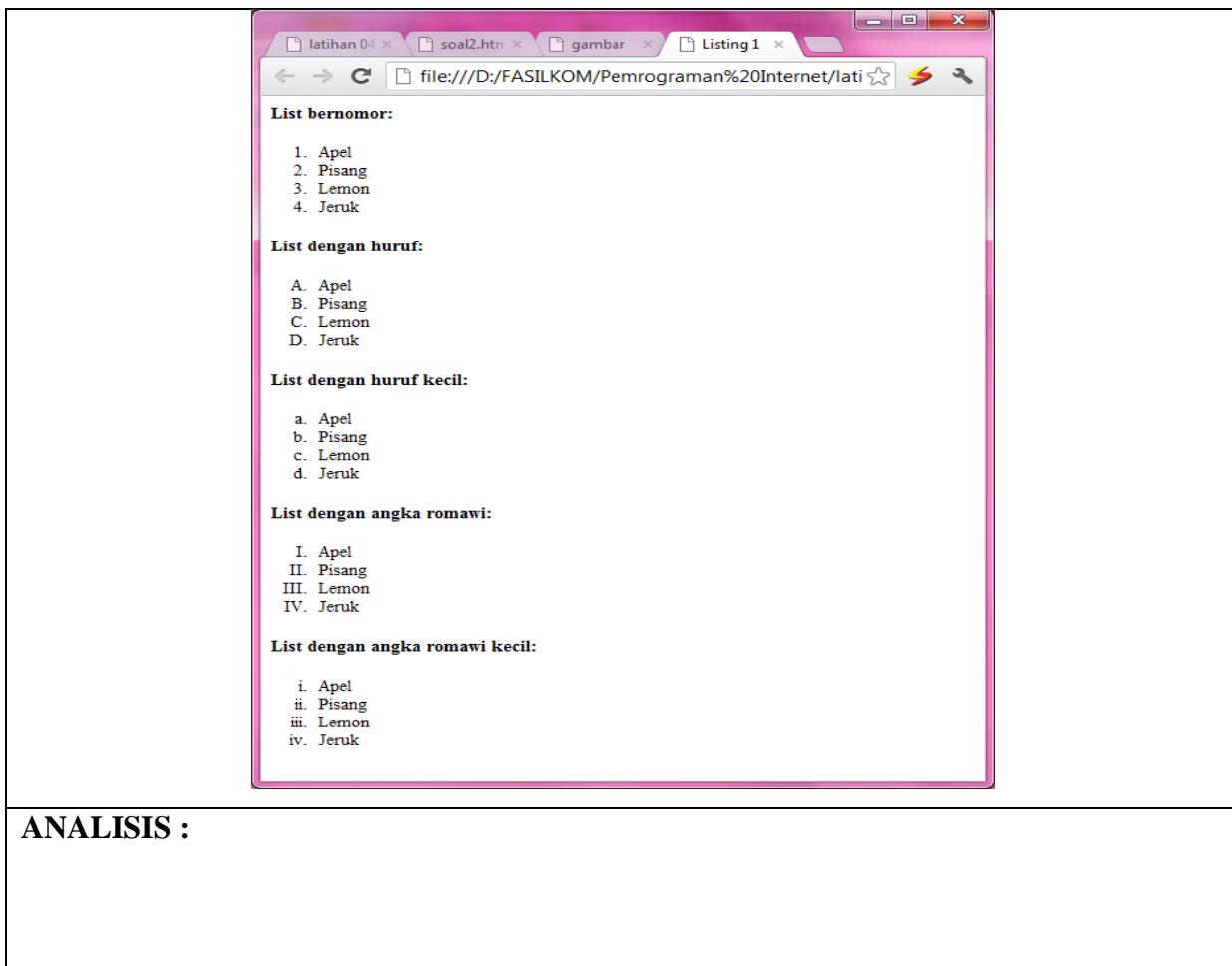
Latihan 3 : Listing1.html

```

<html>
    <head>
        <title>Listing 1 </title>
    </head>
    <body>
        <h4>List bernomor:</h4>
        <ol>
            <li>Apel</li>
            <li>Pisang</li>
            <li>Lemon</li>
            <li>Jeruk</li>
        </ol>
        <h4>List dengan huruf:</h4>
        <ol type="A">
            <li>Apel</li>
            <li>Pisang</li>
            <li>Lemon</li>
            <li>Jeruk</li>
        </ol>
        <h4>List dengan huruf kecil:</h4>
        <ol type="a">
            <li>Apel</li>
            <li>Pisang</li>
            <li>Lemon</li>
            <li>Jeruk</li>
        </ol>
        <h4>List dengan angka romawi:</h4>
        <ol type="I">
            <li>Apel</li>
            <li>Pisang</li>
            <li>Lemon</li>
            <li>Jeruk</li>
        </ol>
        <h4>List dengan angka romawi kecil:</h4>
        <ol type="i">
            <li>Apel</li>
            <li>Pisang</li>
            <li>Lemon</li>
            <li>Jeruk</li>
        </ol>
    </body>
</html>

```

HASIL :



ANALISIS :

Tabel

Start Tag	Kegunaan
<table>	Mendefinisikan sebuah tabel
<th>	Mendefinisikan sebuah header tabel
<tr>	Mendefinisikan suatu barisan dalam tabel
<td>	Mendefinisikan suatu sel dalam tabel
<caption>	Mendefinisikan sebuah caption untuk tabel
<colgroup>	Mendefinisikan sekelompok kolom dalam tabel
<col>	Mendefinisikan nilai atribut untuk satu atau lebih kolom dalam sebuah tabel
<thead>	Mendefinisikan suatu head tabel
<tbody>	Mendefinisikan suatu badan tabel
<tfoot>	Mendefinisikan suatu footer tabel

Atribut Elemen Tabel

Width = panjang (lebar tabel, pixel atau persen)

Height = panjang (tinggi tabel, pixel atau persen)

Border = pixel (tebal garis tepi)

Cellspacing = pixel (spasi antar sel)

Cellpadding = pixel (spasi di dalam sel)

Align = [left | center | right] (perataan tabel)

bgcolor = warna (warna latar belakang tabel)

Atribut Table Row

align = [left|center|right] (perataan sebaris sel secara horisontal)

valign = [top|middle|bottom] (perataan sebaris sel secara vertikal)

bgcolor = warna (warna latar belakang baris)

Atribut Table Data

rowspan = angka (baris yang di span oleh sel)

colspan = angka (kolom yang di span oleh sel)

align = [left | center | right] (perataan horisontal)

valign = [top | middle | bottom] (perataan vertikal)

width = pixel (lebar sel, pixel atau persen)

height = pixel (tinggi sel, pixel atau persen)

bgcolor = warna (warna latar belakang sel)

Latihan 4 : Tabel.html

```
<html>
    <body>
        <p>
            Setiap tabel dimulai dengan tag table. Setiap baris tabel dimulai dengan tag tr.
            Setiap data dalam tabel dimulai dengan tag td.</p>

            <h4>Satu baris satu kolom dengan border normal:</h4>
            <table border="1">
                <tr>
                    <td>100</td>
                </tr>
            </table>

            <h4>Satu baris dan tiga kolom dengan border tebal:</h4>
            <table border="8">
                <tr>
                    <td>100</td>
                    <td>200</td>
                    <td>300</td>
                </tr>
            </table>

            <h4>Dua baris dan tiga kolom dengan border sangat tebal:</h4>
            <table border="15">
```

```

<tr>
    <td>100</td>
    <td>200</td>
    <td>300</td>
</tr>
<tr>
    <td>400</td>
    <td>500</td>
    <td>600</td>
</tr>
</table>

<h4>Tabel ini tidak memiliki border:</h4>
<table border="0">
    <tr>
        <td>100</td>
        <td>200</td>
    </tr>
    <tr>
        <td>300</td>
        <td>400</td>
    </tr>
</table>
</body>
</html>

```

HASIL :



Setiap tabel dimulai dengan tag table. Setiap baris tabel dimulai dengan tag tr. Setiap data dalam tabel dimulai dengan tag td.

Satu baris satu kolom dengan border normal:

100

Satu baris dan tiga kolom dengan border tebal:

100	200	300
-----	-----	-----

Dua baris dan tiga kolom dengan border sangat tebal:

100	200	300
400	500	600

Tabel ini tidak memiliki border:

100	200
300	400

ANALISIS :

Link dan Gambar

Situs web biasanya tidak hanya terdiri atas 1 halaman saja (*homepage*, halaman

awal/pembuka). Bila suatu situs terdiri atas lebih dari 1 halaman, situs tersebut pasti menggunakan hyperlink.

Link

Start Tag	Kegunaan
<a>	Mendefinisikan sebuah anchor
Atribut Target	Kegunaan
Target=" _blank"	Membuat dokumen baru ke dalam windows baru yang kosong
Target=" self"	Membuat dokumen baru ke dalam windows yang sama dengan anchor(default)
Target=" parent"	Membuat dokumen baru ke dalam parent frame (bila menggunakan frame)
Target=" top"	Membuat dokumen baru ke dalam keseluruhan windows browser (cara yang umum digunakan untuk keluar dari frame)

Gambar

Start Tag	Kegunaan
	Mendefinisikan sebuah gambar

Latihan 5 : Gambar dan link.html

```
<html>
    <head>
        <title>Contoh Gambar dan Link</title>
    </head>
    <body>
        Ini adalah link untuk ke <a href="link2.html" target=" _blank">halaman
2
    </body>
</html>
```

HASIL :



Ini adalah link untuk ke halaman 2

ANALISIS :

Latihan 6 : link2.html

```
<html>
  <head>
    <title>gambar </title>
  </head>
  <body>
    
  </body>
</html>
```

HASIL :



ANALISIS :

Soal

1. Tuliskan sintak untuk membuat halaman web seperti gambar berikut ini



2. Tuliskan sintak untuk membuat halaman web seperti gambar berikut ini



DASAR PEMBUATAN WEBSITE DENGAN HTML

Tujuan Praktikum

1. Memahami konsep form dan frame
2. Mengimplementasikan dengan contoh yang ada

Elemen Form <form>

Form HTML merupakan tag yang paling penting khususnya dalam membuat aplikasi berbasis web. Form menyediakan properti masukan yang dapat berupa textbox, check box, radio button, dan button.

Untuk mendeklarasikan sebuah form digunakan tag `<form> ... </form>`. Di dalam tag ini kita akan mendefinisikan elemen-elemen form seperti yang telah disebutkan di atas. Selain tag elemen form kita juga dapat menuliskan sembarang teks, tag, image.

Atribut Elemen Form

ACTION = URI (dari file yang menangani form)

METHOD = [get | post] (metode HTTP untuk men-submit form)

Properti Masukan Pada Elemen Form <input>, <textarea>, <select>

1. Properti Masukan “Textbox”, “Password”, “Hidden”

Metode Penulisan

a. Text Box `<input type="text">`.

Digunakan untuk memasukkan input berupa text.

b. Password `<input type="password">`

Digunakan untuk memasukkan password.

c. Hidden `<input type="hidden">`

Digunakan untuk mengirim data ke suatu aplikasi yang tidak kita inginkan untuk dilihat oleh browser.

Properti Masukan

SIZE = ukuran textbox dalam karakter, default 20

MAXSIZE = karakter maksimum yang akan diterima

NAME	= nama variabel yang dikirim ke aplikasi
VALUE	= akan menampilkan isinya sebagai nilai default (digunakan pada textbox, tidak digunakan pada password).

Pada masukan bertipe input, properti yang digunakan adalah NAME dan VALUE.

2. Properti masukan “Button” dan “CheckBox”

Metode Penulisan:

a. Check Box <input type=”checkbox”>

Check box digunakan untuk dapat memilih lebih dari satu pilihan.

b. Radio Button <input type=”radio”>

Radio button digunakan untuk dapat memilih hanya salah satu pilihan.

c. Push Button <input type=”button”>

Elemen ini biasanya digunakan dengan JavaScript atau VBScript untuk menghasilkan suatu aksi.

PROPERTI MASUKAN:

NAME = nama variabel yang dikirim ke aplikasi.

VALUE = biasanya diset ke sebuah nilai (pada Push Button, VALUE ber isi label berupa text pada tombol).

CHECKED (beri tanda ‘check’ pada radio button atau checkbox. Pada push button properti ini tidak dipakai).

3. Properti masukan “Submit” dan “Reset”

Metode Penulisan:

a. Submit <input type=”submit”>

Setiap elemen form membutuhkan tombol submit untuk mengirimkan nama dan nilainya ke suatu aplikasi yang ditentukan dalam atribut ACTION dalam elemen FORM

b. Image Submit Button <input type=”image” src=”url”>

Digunakan untuk menggantikan tombol standar submit dengan image.

c. Reset <input type=”reset”>

Digunakan untuk mereset semua masukan dalam form.

PROPERTI MASUKAN:

NAME = nama variabel yang dikirim ke aplikasi.

VALUE = berisi label berupa text pada tombol

4. Properti masukkan “Text Area”

Metode Penulisan:

Text Area <textarea> ... </textarea>

Elemen untuk memasukkan teks secara leluasa seperti notepad.

PROPERTI MASUKAN:

NAME = nama variabel yang dikirim ke aplikasi

ROWS = jumlah baris

COLS = jumlah kolom

5. Properti masukan “Select”

Metode Penulisan:

Select <select> ... </select>

Daftar isi dalam properti select menggunakan tag <option>

PROPERTI MASUKAN:

SIZE = jumlah pilihan yang dapat dilihat

NAME = nama variabel yang dikirim ke aplikasi

Latihan 7: Latihan_Form.html

```
<html>
<head><title>Latihan Form</title></head>
<body>
<form action="#" method="get">
Nama: <input type="text" name="nama"><p>
Alamat :<textarea name ="alamat"></textarea>
<p>Jenis Kelamin
<input type ="radio" name ="jenis_kelamin" value="Perempuan"> Perempuan
<input type ="radio" name ="jenis_kelamin" value="Laki-laki"> Laki-laki
<p>Telepon :<input type="text" name="telepon">
<p>Email :<input type="text" name="email">
<p>Pekerjaan:
<select name="pekerjaan">
<option value="mahasiswa">mahasiswa
<option value="pelajar">pelajar
<option value="peg_negeri">peg_negeri
<option value="presiden">presiden
<option value="menteri">menteri
</select>
<p>Hobi:
<input type="checkbox" name="hobi" value="membaca" >Membaca
```

```

<br><input type="checkbox" name="hobi" value="menulis" >Menulis
<br><input type="checkbox" name="hobi" value="main game" >Main Game
<p>
    <input type="submit" value="kirim data" name=submit>
    <input type="reset" value="ulangi" name=reset>
</form>
</body>
</html>

```

HASIL :

The screenshot shows a web browser window with a pink header bar. The title bar says 'Latihan Form'. Below the header, there's a toolbar with buttons for 'Indonesian', 'Translate', ' Nope', 'Never translate Indonesian', and 'Options'. The main content area contains the following form elements:

- Nama :** [Text input field]
- Alamat :** [Text input field]
- Jenis Kelamin :** [Radio button group] Perempuan (selected) | Laki-laki
- Telepon :** [Text input field]
- Email :** [Text input field]
- Pekerjaan :** [Dropdown menu] mahasiswa
- Hobi :**
 - [checkbox] Membaca
 - [checkbox] Menulis (selected)
 - [checkbox] Main Game
- Buttons:** [Submit] kirim data | [Reset] ulangi

ANALISIS :

Frame

Membuat frame berarti membagi web browser menjadi beberapa bagian dalam satu window, dengan tampilan yang dapat diatur tersendiri. Sebuah halaman web dengan frame merupakan halaman web yang khusus mendefinisikan ukuran dan lokasi tiap kandungan

frame. Halaman dengan frame paling sederhana adalah dengan dua frame, yaitu satu untuk menampilkan daftar hyperlink (atau sering disebut dengan navigasi) dan yang lain untuk menampilkan halaman yang ditunjukkan oleh daftar hyperlink tersebut. Isi dari frame tidak lain adalah suatu halaman yang lain.

Sintaks:

<frameset> = Definisi himpunan frame

<frame> = Definisi suatu sub window (frame)

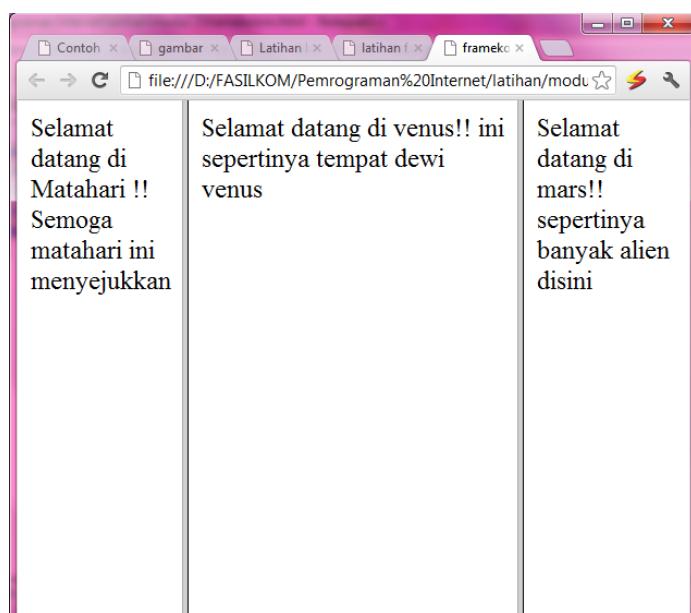
<noframes> = Definisi suatu seksi no frame untuk browser yang tidak dapat menangani frame

1. Frame kolom

Latihan 8: contoh_frame1.html

```
<html>
  <frameset cols="25%,50%,25%">
    <frame src="sun.html">
    <frame src="venus.html">
    <frame src="mars.html">
  </frameset>
</html>
```

HASIL :



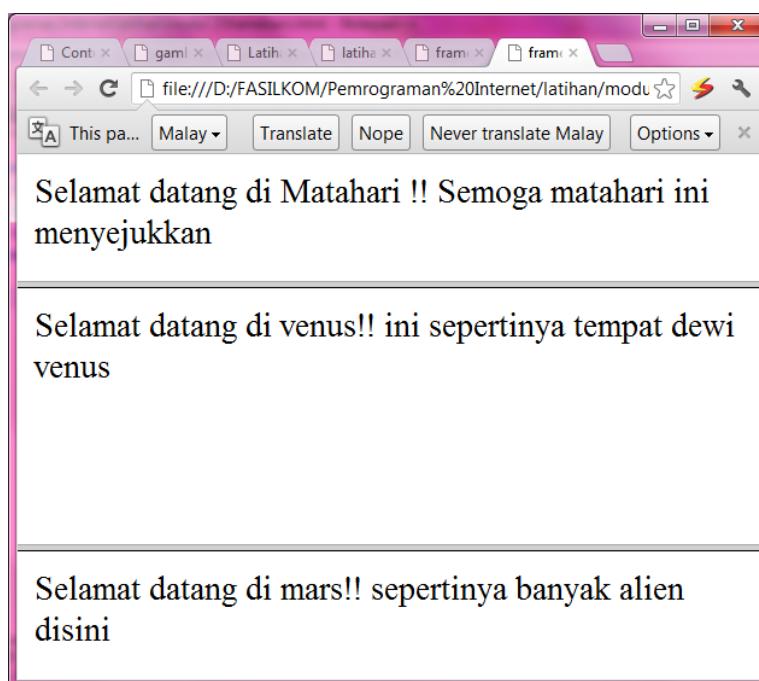
ANALISIS :

2. Frame Baris

Latihan 9: contoh_frame2.html

```
<html>
    <frameset rows="25%,50%,25%">
        <frame src="sun.html">
        <frame src="venus.html">
        <frame src="mars.html">
    </frameset>
</html>
```

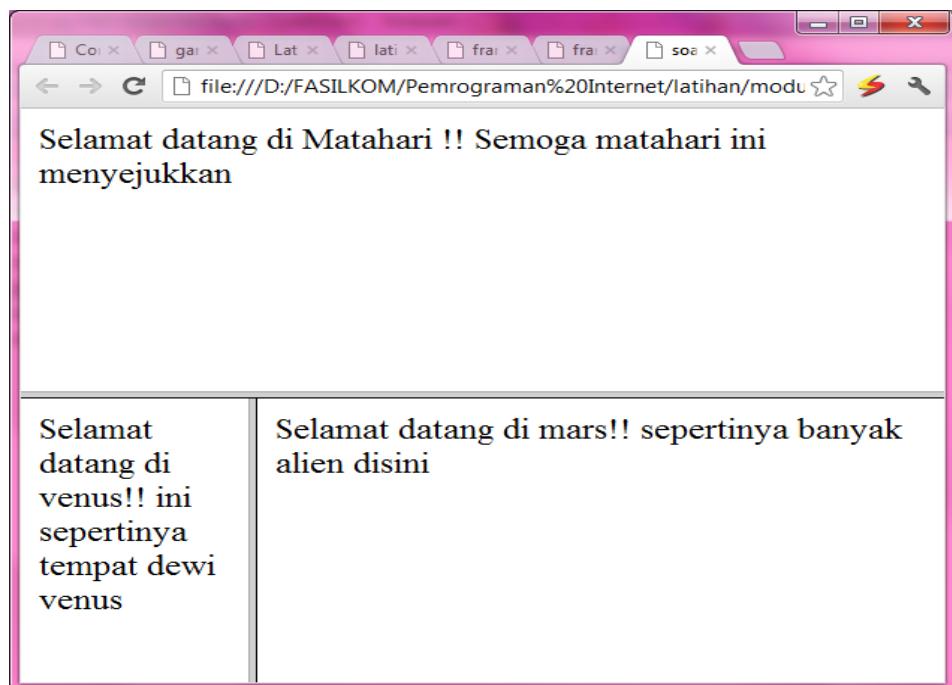
HASIL :



ANALISIS :

Soal:

1. Tuliskan sintak untuk membuat halaman web seperti gambar berikut ini



2. Buatlah sebuah web sederhana mengenai penjualan online memiliki minimal 5 halaman yang saling berhubungan. Salah satu halamannya berisi form.

CSS

Tujuan Praktikum

1. Memahami pembuatan CSS
2. Mampu mengimplementasikan CSS dalam halaman HTML

Pengenalan CSS

CSS adalah kependekan dari Cascading Style Sheet, berfungsi untuk mempercantik penampilan HTML atau menentukan bagaimana elemen HTML ditampilkan, seperti menentukan posisi, merubah warna teks atau background dan lain sebagainya.

Perhatikan contoh kode CSS berikut:



Selector

Selector adalah elemen/tag HTML yang ingin diberi style.

Penamaan Selector:

1. Jika tag HTML yang ingin diberi style memiliki ID, anda dapat menuliskan nama ID tersebut dengan diawali tanda kress (#).
Contoh: #header
2. Dan jika tag yang diberi style memiliki Class, maka penulisan selector bisa dilakukan dengan tanda titik (.) diikuti dengan nama class
Contoh: .artikel
3. Jika hanya menuliskan satu selector, seperti contoh kode CSS di atas, maka seluruh tag h1 yang terdapat dalam file HTML akan memiliki style yang sama. Bagaimana jika hanya ingin memberi style pada tag h1 yang *hanya terdapat di dalam* Class artikel. Maka penulisan selectornya seperti berikut: .artikel h1

Dapat dipilih lebih dari satu tag untuk penghematan kode CSS.

Contoh:

```
h1{background-color: #666666;}  
P {background-color: #666666;}  
a {background-color: #666666;}
```

Kode dapat digabungkan selector dengan menambahkan tanda koma pada nama tag yang ingin diberi style.

```
h1, p, a {background-color: #666666;}
```

Property dan Value

Property adalah sifat-sifat yang ingin diterapkan pada selector, seperti warna text, warna background, jarak antar elemen, garis pinggir dan lain sebagainya.

Untuk memberikan nilai/value pada property kita gunakan tanda titik dua (:). Setiap property diakhiri dengan titik koma (;

```
.artikel h1 {  
    color: red;  
    background-color: blue;  
    font-size: 20px;  
}
```

Penulisan CSS

Ada tiga cara penulisan kode CSS, yaitu

1. inline, contoh: <h1 style="Color: red;"> Judul Situs </h1>
2. internal
3. external

Latihan: ContohInternalCSS.html

```
<html>  
    <head>  
        <title> Tutorial CSS - Cara Merubah Warna Tulisan Teks </title>  
        <style>  
            body {color: blue;}  
            h1 {color: #00ff00;}  
            h2 {color: rgb (255,0,0);}  
        </style>  
    </head>  
    <body>
```

```
<h1> Saya sedang belajar CSS</h1>
<h2>Saya sedang belajar CSS</h2>
    Saya sedang belajar CSS
</body>
</html>
```

HASIL :

ANALISIS :

Latihan: ContohCss1.css

```
h3 {font-family : calibri;
    color : red;
    font-style : italic
}
```

Latihan: LatihanCSSExternal.html

```
<html>
    <head>
        <title> Belajar tutorial CSS</title>
        <link rel="stylesheet" href="ContohCss1.css" type="text/css">
    </head>
    <body>
        <h3>kalimat ini memiliki font jenis calibri, warna merah dan juga di cetak miring</h3>
    </body>
</html>
```

HASIL :

ANALISIS :

SOAL

Buatlah css, sedemikian hingga tampilan menjadi seperti berikut:

Ini adalah Heading 1 (Berwarna Hijau)

Ini adalah paragraf pertama. Perhatikan bahwa warna tulisan ini adalah merah. Warna default untuk text, didefinisikan di selektor body.

Ini adalah tulisan pada paragraf dengan kelas="ex". Warna paragraf ini adalah biru.

Tulisan pada paragraf ini, memiliki indent 50 pixel. Maka definisikanlah pada css untuk tag p. Seorang ulama pernah menyampaikan, bahwa keberhasilan pendidikan / ilmu yang kita miliki, banyak dipengaruhi oleh faktor keimanan dan ketakutan. Sayangnya pada sistem pendidikan dewasa ini, dua hal tersebut diabaikan dan tidak masuk ke dalam kurikulum yang harus diberikan dalam proses pendidikan.

Heading pada Class alain ini diset tengah

Tulisan ini, ditulis dalam tag p dengan kelas 'tenggal', di atur rata kanan. May, 2009

Tulisan ini ditulis pada tag p dengan kelas 'main'. Tulisan ini di set alignment justify. Sehingga pada paragraf ini akan diatur perataan kanan dan kiri. Hal ini dapat dibuktikan dengan melihat perataan kiri dan kanan pada batas kanan dan kiri, di atur rata kanan dan kiri. Untuk memberikan ilustrasi yang mudah dilihat, maka tulisan pada paragraf ini diperpanjang, untuk melihat batas kanan dan kiri pada banyak baris, diparagraf ini. Silahkan atur pada CSS untuk membuat tampilan paragraf ini menjadi rata kanan dan kiri. Sehingga setiap tulisan yang ditulis didalam tag p dengan kelas 'main' akan diatur rata kanan dan rata kiri.

Catatan: Ubah ukuran browser, diperbesar atau diperkecil untuk melihat pengaturan 'justify' bekerja.

Link to: Mat itu aku

Setiap tulisan Heading 1, diberikan garis atas.

Tulisan Heading 2 dicoret

Tulisan Heading 3 diberi garis bawah

TEXT PADA HTML, ADA HURUF UPPERCASE ADA HURUF LOWERCASE. TAPI PADA PARAGRAF DI KELAS INI, DIATUR MENJADI UPPERCASE SEMUA.

text pada html, ada huruf uppercase ada huruf lowercase tapi pada paragraf di kelas ini, diatur menjadi lowercase semua.

Text Pada HTML, Ada Huruf UPPERCASE Ada Huruf Lowercase. Tapi Pada Paragraf Di Kelas Ini, Diatur Menjadi Capitalize.

JAVASCRIPT

Tujuan Praktikum

1. Memahami cara pembuatan Javascript
2. Memahami kegunaan dari Javascript
3. Mampu mengimplementasikannya javascript dalam halaman HTML

Aplikasi Untuk Menjalankan JavaScript

JavaScript merupakan *bahasa script* yang berjalan pada *web browser*, sehingga program yang dibutuhkan untuk menjalankan *JavaScript* hanyalah sebuah aplikasi text editor dan sebuah web browser seperti *Google Chrome* atau *Mozilla Firefox*.

Cara penulisan *JavaScript* mirip dengan penulisan bahasa pemrograman web lainnya seperti PHP dan CSS, yakni dengan menyisipkan kode *JavaScript* di dalam *HTML*.

Cara Memasukkan JavaScript menggunakan tag <script> (internal JavaScript)

Sebagai contoh cara penginputan *JavaScript* dengan tag *<script>*, berikut adalah kode *HTML*nya:

LATIHAN: LatihanJSInternal.html

```
<html>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <head>
        <title>Belajar JavaScript </title>
        <script>
            alert ("Hello World!!");
        </script>
    </head>
    <body>
        <h1>Belajar JavaScript</h1>
        <p>Saya sedang belajar JavaScript </p>
        <p>Belajar Web Programming p>
    </body>
</html>
```

HASIL :

ANALISIS :

Cara Memasukkan JavaScript Menggunakan tag <script src=" " > (external JavaScript)

LATIHAN: kode_javascript.js

```
alert ("Hello World!!")
```

LATIHAN: LatihanJSExternal.html

```
<html>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <head>
    <title>Belajar JavaScript </title>
    <script src="kode_javascript.js"></script>
  </head>
  <body>
    <h1>Belajar JavaScript</h1>
    <p>Saya sedang belajar JavaScript </p>
    <p>Belajar Web Programming</p>
  </body>
</html>
```

HASIL :

ANALISIS :

Cara Memasukkan JavaScript Menggunakan Event Handler (Inline JavaScript)

LATIHAN: LatihanJSEventHandler.html

```
<html>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <head>
    <title>Belajar JavaScript</title>
  </head>
  <body>
    <h1>Belajar JavaScript</h1>
    <p>Saya sedang belajar JavaScript p>
    <p>Belajar Web Programming p>
    <button onclick="alert('Hello World!!')">Klik Saya
  </body>
</html>
```

HASIL :

ANALISIS :

Cara Memasukkan JavaScript Menggunakan URL (href:="javascript:")

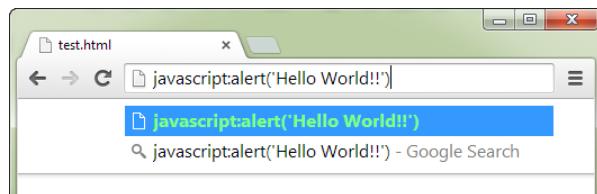
LATIHAN: LatihanJSURL.html

```
<html>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <head>
        <title>Belajar JavaScript </title>
    </head>
    <body>
        <h1>Belajar JavaScript</h1>
        <p>Saya sedang belajar JavaScript </p>
        <p>Belajar Web Programming </p>
        <a href="javascript:alert('Hello World!!!')">Hallo Dunia...</a>
    </body>
</html>
```

HASIL :

ANALISIS :

Anda juga bisa menggunakan **konsep** protocol javascript **ini untuk** menjalankan perintah JavaScript tanpa membuat halaman HTML. Caranya adalah dengan menuliskan perintah JavaScript langsung di bagian address bar dari web browser, seperti contoh berikut:



Cara Menampilkan Hasil Program JavaScript

Untuk menampilkan hasil program ke dalam web browser terdapat beberapa langkah yang agak panjang yaitu:

1. *Pertama*, kita harus membuat sebuah **tag ‘container’**, atau tag penampung untuk hasil program JavaScript. Tag *container* ini bisa berupa tag HTML apapun, seperti tag paragraf **<p>** atau tag **<div>**.
2. *Kedua*, kita harus mencari elemen ‘*container*’ ini dari JavaScript. JavaScript menyediakan beberapa cara untuk mengakses elemen dalam HTML. Salah satu caranya

adalah dengan menggunakan **fungsi** (atau lebih tepatnya: **method**): **document.getElementById("id_continer")**. Fungsi **getElementById** akan mencari elemen HTML yang memiliki atribut **id** yang diinputkan di dalam tanda kurung.

3. Langkah ketiga, adalah *menginputkan* hasil program kedalam tag ‘*container*’ dengan menggunakan properti **innerHTML**.

Untuk mempermudah pemahaman cara menampilkan hasil program JavaScript ke dalam web browser, berikut adalah contoh program untuk menampilkan hasil penjumlahan:

Latihan: ContohMenampilkanHasilJS.html

```
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Belajar JavaScript</title>

    <script>
        window.onload = function ()
        {
            var hasil;
            hasil = 1+3+5+7+9;
            document.getElementById("tempat_hasil").innerHTML=hasil;
        }
    </script>
</head>
<body>
    <h1>Belajar JavaScript</h1>
    <p> Saya sedang belajar JavaScript </p>
    <div id="tempat_hasil"></div>
</body>
</html>
```

HASIL :

ANALISIS :

Fungsi Alert untuk Menampilkan Hasil Program JavaScript

Fungsi alert merupakan cara paling sederhana untuk menampilkan hasil program **JavaScript**. *Fungsi alert* akan menampilkan ‘apapun’ yang diberikan sebagai argumen ke dalam fungsi ini. Setiap output yang ditampilkan akan dikonversi menjadi bentuk text (tipe data **String**). Fungsi alert akan menampilkan hasil **JavaScript** dengan cepat.

Latihan: ContohMenampilkanAlertHasilJS.html

```
<!DOCTYPE html>
<html>
```

```

<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Belajar JavaScript</title>

    <script>
        window.onload = function()
        {
            var hasil;
            hasil = 1+3+5+7+9;
            alert(hasil);
        }
    </script>
</head>

<body>
    <h1>Belajar JavaScript</h1>
    <p> Saya sedang belajar JavaScript </p>
    </div>
</body>
</html>

```

HASIL :

ANALISIS :

Cara lain untuk menampilkan hasil dari **JavaScript** adalah menggunakan fungsi **document.write()**. Berikut adalah cara penggunaannya:

LATIHAN: ContohMenampilkanDocWriteHasilJS.html

```

<!DOCTYPE html>
<html>
    <head>
        <title>Belajar JavaScript</title>

        <script>
            window.onload = function()
            {
                document.write("Hello");
            }
        </script>
    </head>
    <body>
        <h1>Belajar JavaScript</h1>
        <p> Saya sedang belajar JavaScript </p>
        </div>
    </body>
</html>

```

HASIL :

ANALISIS :

Aturan Dasar Penulisan Kode Program JavaScript

Di dalam **JavaScript**, penulisan huruf besar dan huruf kecil dibedakan, atau dalam istilah pemograman bersifat **Case Sensitif**. Hal ini berarti penulisan *variabel*, *keyword*, maupun nama *fungsi* di dalam JavaScript harus konsisten. Variabel **nama**, **Nama**, dan **NAMA** merupakan 3 variabel berbeda. Sedangkan untuk penulisan *keyword while*, harus ditulis dengan ‘**while**’, bukan ‘**While**’ atau ‘**WHILE**’.

Cara Penulisan Komentar dalam JavaScript

JavaScript mendukung 2 jenis cara penulisan komentar, yakni menggunakan karakter // untuk komentar dalam 1 baris, dan karakter pembuka komentar /* dan penutup */ untuk komentar yang mencakup beberapa baris.

```
<script>
    // ini adalah komentar dalam 1 baris
    /* Baris ini juga merupakan komentar */
        //ini juga komentar

    /* Komentar ini
       mencakup beberapa
       baris
    */

    /*
     * Beberapa programmer
     * menambahkan tanda bintang
     * agar penulisan komentar
     * lebih rapi
     */
</script>
```

Di dalam JavaScript, *identifier* adalah sebutan untuk *nama*. *Nama* ini bisa terdiri dari nama variabel, atau nama dari fungsi. Aturan penulisan *identifier* dalam JavaScript adalah :

- Karakter pertama harus diawali dengan huruf, underscore (_) atau tanda dollar (\$)
- Karakter kedua dan seterusnya bisa ditambahkan dengan huruf, angka, underscore (_) atau tanda dollar (\$).

Dari aturan tersebut dapat dilihat bahwa kita tidak bisa menggunakan angka sebagai karakter pertama dari sebuah variabel atau nama fungsi.

Berikut adalah contoh penulisan nama variabel yang dibolehkan:

```
<script>
    duniaIlkom
    $ilkom
    v12
    _karakter
    b3l4j4r
</script>
```

Namun karakter berikut tidak bisa digunakan sebagai identifier:

```
<script>
    %duniaIlkom // terdapat karakter %
    dunia ilkom // terdapat karakter spasi
    4ngka // diawali dengan angka
    suka#suka // terdapat karakter #
</script>
```

Aturan Penulisan Tanda Semicolon pada Akhir Baris

Didalam **JavaScript** karakter titik koma (bahasa inggris : **semicolon**) sifatnya **opsional** untuk digunakan sebagai penanda akhir dari baris program, dan boleh tidak ditulis.

LATIHAN: ContohJS.html

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Belajar JavaScript</title>
    <script>
        a=13
        b=a+2
        console.log(b)
        alert(b)
    </script>
</head>
<body>
    <h1>Belajar JavaScript</h1>
    <p> Saya sedang belajar JavaScript </p>
    </div>
</body>
</html>
```

HASIL :

ANALISIS :

SOAL

Berikan contoh cara penulisan logika IF-ELSE, SWITCH, FOR, WHILE, DO WHILE, FOR IN dalam javascript.

PENGENALAN PHP

Tujuan Praktikum

1. Memahami konsep kerja PHP
2. Memahami cara menyisipkan script-script PHP dalam tag HTML
3. Memahami variable, struktur, operator dan metode operasi aritmatika PHP

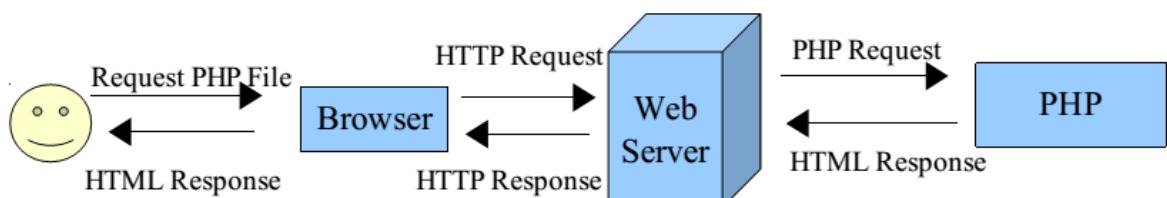
Dasar Teori PHP

PHP (akronim dari PHP: Hypertext Preprocessor) adalah bahasa server-side scripting yang didesain khusus untuk web. Pada halaman HTML anda dapat menempelkan(embed) kode PHP. Kode PHP dieksekusi di sisi server bukan di komputer klien. Dan hasil yang ditampilkan adalah kode HTML murni. PHP adalah salah satu bahasa pemrograman di Internet.

PHP bersifat cepat, gratis dan murah (gratis), selain itu PHP mendukung penggunaan database seperti MySQL, PostgreSQL, mSQL, Oracle etc. Untuk dapat menjalankan PHP melalui browser, maka anda diharuskan terlebih dahulu menginstall web server (misalnya Apache, PWS, IIS) lalu menginstall PHP, sedangkan untuk menjalankan MySQL anda tidak perlu menginstall web server, hanya saja jika ingin dijalankan melalui browser, maka anda harus menginstall web server.

Cara Kerja PHP

Cara kerja PHP yang akan dibahas adalah PHP sebagai bahasa pemrograman untuk mengembangkan aplikasi berbasis web. Karena selain untuk web-programming PHP juga dapat digunakan untuk mengembangkan aplikasi berbasis desktop dan CLI (Command Line Interface).



DASAR PHP

PHP dijalankan dalam file berekstensi. php, .php3 atau .phtml, itu tergantung dengan settingan PHP anda, tetapi secara umum ekstensi file PHP adalah. php.

Kode PHP menyatu dengan tag – tag HTML dalam satu file. Kode PHP diawali dengan tag <? atau <?php dan ditutup dengan ?>.

Ada beberapa cara untuk memulai menuliskan script PHP, yaitu:

<?php script PHP anda ?>	<? script PHP anda ?>
<script language="php"> script PHP anda </script>	<% script PHP anda %>

Cara pertama merupakan format yang dianjurkan tetapi mungkin cara yang kedua lebih sering digunakan karena lebih ringkas. Cara yang ketiga digunakan untuk mengantisipasi editor dan web server yang tidak dapat menerima kedua cara di atas. Selain itu kita juga bisa menggunakan cara penulisan ASP, tetapi tentu saja ada beberapa konfigurasi yang perlu ditambahkan terlebih dahulu.

Latihan: latihanphp1.php

```
<html>
    <head>
        <title>Example</title>
    </head>
    <body>
        <?php
            // Ini adalah contoh penulisan keterangan (komentar)
            // dalam script php.
            echo "Ini script PHP pertamaku!";
        ?>
    </body>
</html>
```

HASIL :



The screenshot shows a Microsoft Internet Explorer window with a pink title bar. The address bar displays "localhost/ngajar/php1.php". The main content area of the browser shows the text "Ini script PHP pertamaku!" in black font.

ANALISIS :

Cobalah salin script di atas dengan text editor (bisa dengan Notepad) kemudian simpan di direktori htdocs atau home directory yang telah ditentukan. Jangan lupa untuk memberi nama file dengan ekstensi php. Misalnya, php01.php. Setelah itu jalankan web browser. Pada address bar, tuliskan hostname komputer diikuti nama file. Sebagai contoh, localhost/direktoriku/php01.php

Variabel

Di setiap bahasa pemrograman pasti akan kita temui konsep variabel. Variabel adalah sebuah tempat di memori untuk menyimpan data yang nilainya dapat berubah-ubah. Namun tidak seperti pada bahasa pemrograman lain yang mengharuskan kita untuk mendeklarasikan variabel terlebih dahulu, variabel dalam PHP tidak harus dideklarasikan sebelum variabel tersebut digunakan. Variabel diwakili oleh kata tertentu dengan aturan penulisan sebagai berikut:

1. Variabel dimulai dengan tanda dollar (\$).
2. Harus dimulai dengan huruf atau underscore (_).
3. Tidak boleh menggunakan tanda baca.
4. Case sensitive atau huruf besar dan huruf kecil dibedakan.
5. Hindari penggunaan kata yang merupakan fungsi built-in PHP.

Contoh:

```
$nama = "Ini tipe String"; // deklarasi tipe string;  
  
$numeric = 123; // deklarasi integer  
  
$numeric = 456.123; // Sekarang $numeric bertipe float  
  
$boolean = true; // deklarasi tipe Boolean
```

Contoh: LatihanPHP2.php

```
<?  
 $nama = "divisi komputer hme UNSRI"; // Huruf kecil semua  
 $NAMA = "DIVISI KOMPUTER HME UNSRI"; // Huruf besar semua  
 print "$nama → variable menggunakan huruf kecil. <br>";  
 print "$NAMA → variable menggunakan huruf besar.";  
?>
```

HASIL :

ANALISIS :

Tipe Data

Berikut ini adalah beberapa tipe data yang didukung oleh PHP.

TipeData	Keterangan
Integer	Digunakan untuk semua angka
String	Digunakan untuk semua huruf, angka, spasi, dan simbol
Double	Digunakan untuk bilangan real
Boolean	Digunakan untuk nilai True atau False
Array	Digunakan untuk menampung beberapa data sekaligus
Object	Digunakan untuk class

Contoh: LatihanPHP3.php

```
<?  
 // Penggunaan tipe data int, float dan string seta type castingnya  
 $var = 2002; // var sebagai integer  
 print "Sekarang tahun $var.<br>";  
 $var = "dua ribu dua"; // var sebagai string  
 print "Sekarang tahun $var.<br>";  
 $var = 2002.5; // var sebagai bilangan bulat / float
```

```

        print "Tahun $var, ada atau nggak?" ;
?>
<?
    // Penggunaan tipe data object
class himp
{
    $divisi = "Divkom HME UNSRI";
    function ubah($str)
    {
        $this->divisi = $str;
    }
}
$hme = new himp;
print $hme->divisi;
print "<br>";
$hme->ubah ("Divisi Komputer HME UNSRI");
print $hme->divisi;
?>

```

HASIL :

ANALISIS :

Konstanta

Cara pendeklarasiannya konstanta digunakan keyword define untuk mendeklarasikan variabel. Konstanta juga tidak diawali dengan tanda \$ (dollar).

```

define ('HARGA', 1500);

define ('NAMA', 'FASILKOM UNSRI');

```

Komentar

Komentar pada script bertujuan untuk memberitahu pembacanya, baik orang lain ataupun anda sendiri. Biasanya komentar digunakan untuk menjelaskan tujuan dituliskannya script tersebut, siapa penulisnya, kapan ditulis dan sebagainya. Komentar juga berguna bagi anda sendiri ketika suatu waktu anda lupa mengapa anda menulis file ini, fungsi-fungsinya untuk apa dan banyak lainnya. PHP akan mengabaikan semua text didalam komentar. Jadi tidak akan berpengaruh pada jalannya suatu script. PHP mengenal tiga jenis gaya komentar. Pertama model bahasa C (banyak baris/multi line), biasanya dapat diletakkan pada bagian atas script. Lihatlah script dibawah ini.

Latihan Penerapan Variabel, Konstanta dan Komentar: variabel.php

```
<html>
  <head>
    <title>Variabel dan Konstanta</title>
  </head>
  <body>
    <?php
/*
  Filename: variabel.php
  Author: Rio Astamal
  Created: 20-02-2016 18:22
  Updated: 22-05-2016 16:11
  Desc: Contoh penerapan komentar
*/
    $menu = 'Menu: Nasi Pecel';
    define ('HARGA', 2500);

    $menu1 = 'Menu: Nasi Rawon';
    define ('HARGA1', 5000);

    echo $menu. '<br />';
    echo HARGA;
    echo $menu1. '<br />';
    echo HARGA1;
  ?>
  </body>
</html>
```

HASIL :

ANALISIS :

Operator

Operator adalah simbol yang digunakan untuk memanipulasi data, seperti penambahan dan pengurangan. Ada operator yang menggunakan satu operand, ada juga yang menggunakan dua operand. Sedangkan operand adalah data yang dioperasikan atau dimanipulasi. Di sini operand dapat digantikan dengan variabel. Operator dapat dikelompokkan dalam 4 kategori, yaitu:

1. Operator aritmatika adalah operator yang berhubungan dengan fungsi matematika.
2. Operator logika adalah operator yang membandingkan TRUE dan FALSE.
3. Operator bitwise adalah operator yang membandingkan binary.
4. Ada juga operator yang sering digunakan namun tidak termasuk dalam kelompok diatas. Kita dapat mempelajarinya dalam kelompok lain.

Operator Aritmatika

Operator aritmetika merupakan operator yang berhubungan dengan fungsi matematika. Operator ini sering digunakan dalam program yang akan kita buat. Tabel di bawah ini menunjukkan operator aritmatika yang didukung oleh PHP. Dua baris terakhir tabel merupakan operator aritmatika yang digunakan dengan satu operand yang berfungsi untuk menambahkan atau mengurangkan nilai satu dari operand yang diikuti operator tersebut.

Operator	Nama	Contoh
+	Penambahan	\$a + \$b
-	Pengurangan	\$a - \$b
*	Perkalian	\$a * \$b
/	Pembagian	\$a / \$b
%	Modulus	\$a % \$b

Contoh

```
<?php
$bil1 = 10;
$bil2 = 5;
$bil3 = 3;
echo $bil1 % $bil2; // output 0
echo $bil1 % $bil3; // output 1
echo $bil2 % $bil3; // output 2
?>
```

HASIL:

ANALISIS:

Operator Kombinasi

Operator	Penggunaan	Sama dengan
+=	\$a += \$b	\$a = \$a + \$b
-=	\$a -= \$b	\$a = \$a - \$b
*=	\$a *= \$b	\$a = \$a * \$b
/=	\$a /= \$b	\$a = \$a / \$b
%=	\$a %= \$b	\$a = \$a % \$b

Operator Perbandingan

Operator perbandingan digunakan untuk membandingkan dua nilai. Penggunaan ekspressi ini juga akan menghasilkan nilai true atau false tergantung dari perbandingan. Operator perbandingan biasanya digunakan pada struktur kontrol seperti percabangan dan perulangan.

Operator	Nama	Penggunaan
<code>==</code>	sama dengan	<code>\$a == \$b</code>
<code>===</code>	identik (harus bertipe sama)	<code>\$a === \$b</code>
<code>!=</code>	tidak sama dengan	<code>\$a != \$b</code>
<code><></code>	tidak sama dengan	<code>\$a <> \$b</code>
<code><</code>	kurang dari	<code>\$a < \$b</code>
<code>></code>	lebih dari	<code>\$a > \$b</code>
<code><=</code>	kurang dari atau sama dengan	<code>\$a <= \$b</code>
<code>>=</code>	lebih dari atau sama dengan	<code>\$a >= \$b</code>

Operator Logika

Operator logika digunakan untuk melakukan tes terhadap sebuah kondisi logika dari dua atau lebih perbandingan.

Operator	Nama	Penggunaan	Keterangan
<code>!</code>	NOT	<code>!\$a</code>	TRUE jika <code>\$a</code> adalah FALSE, berlaku sebaliknya
<code> </code>	OR	<code>\$a \$b</code>	TRUE jika <code>\$a</code> atau <code>\$b</code> bernilai TRUE atau keduanya bernilai TRUE
<code>&&</code>	AND	<code>\$a && \$b</code>	TRUE jika <code>\$a</code> dan <code>\$b</code> bernilai TRUE
<code>xor</code>	XOR	<code>\$a xor \$b</code>	TRUE jika <code>\$a</code> atau <code>\$b</code> bernilai TRUE, tapi tidak keduanya

Operator Increment/Decrement

Operator increment/decrement digunakan untuk menambah/mengurangi nilai dari suatu variabel dengan satu. Increment memiliki simbol `++` (double plus) sedangkan decrement `--` (double minus).

Nama	Contoh	Keterangan
Pre-Increment	<code>++\$a</code>	Tambah <code>\$a</code> dengan satu, lalu kembalikan nilainya
Post-Increment	<code>\$a++</code>	Kembalikan nilai <code>\$a</code> , lalu tambah <code>\$a</code> dengan satu
Pre-Decrement	<code>--\$a</code>	Kurangi <code>\$a</code> dengan satu, lalu kembalikan nilainya
Post-Decrement	<code>\$a--</code>	Kembalikan nilai <code>\$a</code> , lalu kurangi <code>\$a</code> dengan satu

Contoh:

```

<?php
    echo "<h3>Post-increment</h3>";
    $a = 5;
    echo "Seharusnya 5: " . $a++ . "<br />";
    echo "Seharusnya 6: " . $a . "<br />";

    echo "<h3>Pre-increment</h3>";
    $a = 5;
    echo "Seharusnya 6: " . ++$a . "<br />";
    echo "Seharusnya 6: " . $a . "<br />";

    echo "<h3>Post-decrement</h3>";
    $a = 5;
    echo "Seharusnya 5: " . $a-- . "<br />";
    echo "Seharusnya 4: " . $a . "<br />";

    echo "<h3>Pre-decrement</h3>";
    $a = 5;
    echo "Seharusnya 4: " . --$a . "<br />";
    echo "Seharusnya 4: " . $a . "<br />";
?>

```

HASIL:

ANALISIS:

Operator String

Operator string yang dimaksud adalah tanda “.” (titik). Tanda titik ini dapat digunakan untuk menggabung string.

Contoh:

```

<?php
    $a = 'FASILKOM';
    $b = 'UNSRI';
    $c = $a . ' ' . $b; // hasil: FASILKOM UNSRI
    // atau dengan kombinasi
    $a .= ' UNSRI'; // hasil: FASILKOM UNSRI
?>

```

HASIL:

ANALISIS:

Escape Character

Ada beberapa solusi untuk mencetak petik didalam petik itu sendiri:

1. Gunakan petik satu ('') sebagai penutup string jika ingin mencetak petik dua (") dan sebaliknya.
2. Menggunakan Escape Character \ (backslash)

Berikut ini adalah contoh penyelesaian dari kedua solusi diatas.

```
<?php
/*
SOLUSI No. 1
=====
Jika ingin mencetak petik satu gunakan petik dua sebagai penutup string
Jika ingin mencetak petik dua gunakan petik satu sebagai penutup string
*/
$a = 'John Says: "My dear, I Love You."<br />';
$b = "The Girl Says: 'Go to Hell...!!!!<br />";
echo $a . $b . '<br />';
/*
SOLUSI No. 2
=====
Gunakan escape character \ (backslash)
*/
$a = "John Says: \"My dear, I Love You.\"<br />";
$b = 'The Girl Says: \'Go to Hell...!!!!\'<br />';
echo $a . $b;
?>
```

HASIL:

ANALISIS:

STRUKTUR KENDALI DALAM PHP

Tujuan Praktikum

1. Mengenal dan memahami beberapa macam struktur kontrol (kondisi) dan proses perulangan.
2. Mengimplementasikan struktur kontrol dan perulangan dalam berbagai macam kasus.

Struktur Kontrol

Struktur Kontrol merupakan sebuah struktur dalam bahasa pemrograman yang membolehkan kita untuk mengontrol alur dari eksekusi suatu program atau script. Struktur kontrol mencakup struktur kondisi dan struktur pengulangan atau looping. Struktur kondisi terdiri dari beberapa statement yaitu:

1. if..
2. if...else...
3. if...elseif...else...
4. switch...case...break...

Sedangkan struktur perulangan terdiri dari:

1. for...
2. while...
3. do...while...
4. foreach...

Struktur Kondisi

Statement if...

Kita dapat menggunakan statemen if untuk membuat sebuah keputusan. Jika kondisi bernilai true maka blok if akan dieksekusi. Statemen if dikelilingi oleh tanda { } (kurung kurawal). Syntax dasar penulisan statemen if adalah:

```
if (kondisi) {  
  
    // kode yang dieksekusi  
  
}
```

Jika kondisi bernilai TRUE maka perintah yang ada blok pada { } akan dieksekusi. Perhatikan contoh dibawah

```
<?php  
$nama = 'FASILKOM UNSRI';  
if ($nama == 'FASILKOM UNSRI')  
{  
    echo 'OK, anda boleh masuk.';  
}  
?>
```

HASIL:

ANALISIS:

Statement if..else...

Statement ini hampir sama dengan if hanya saja pada if else kita diberikan opsi alternatif untuk menentukan aksi yang dilakukan jika kondisi bernilai *false*. Statement ini akan sangat sering anda gunakan dibanding penggunaan if saja. Syntax penulisan statement if...else... adalah sebagai berikut:

```
if (kondisi) {  
  
    // kode yang dieksekusi jika kondisi true  
  
}  
  
else {  
  
    // kode yang dieksekusi jika kondisi false  
  
}
```

Contoh statement if...else... sederhana dapat anda lihat berikut ini.

```
<?php
    $peghasilan = 5000000;
    if ($penghasilan >= 3000000) {
        echo 'Pak Toyib: "Nak john, kamu boleh menikahi putri saya :)"';
    }
    else {
        echo 'Pak Toyib: "Mau kamu kasih makan apa putri saya nanti!!!!"';
    }
?>
```

HASIL:

ANALISIS:

Statemen if...else if...else...

Jika statemen if...else... hanya dapat menggunakan satu alternatif jika kondisi awal bernilai false. Statemen if...elseif... memberi lebih dari satu aksi alternatif. Hal ini memungkinkan kita untuk memilih aksi dari banyak kemungkinan kondisi. Syntax dasar penulisan if...elseif... adalah sebagai berikut:

```
if (kondisi_1) {
    // kode yang dieksekusi
}
elseif (kondisi_2) {
    // kode yang dieksekusi
}
elseif (kondisi_3) {
    // kode yang dieksekusi
}
else if (kondisi_4) {
    // kode yang dieksekusi
}
else {
```

```
// kode yang dieksekusi  
}
```

Anda dapat menulis statement elseif dengan spasi, yaitu else if. Keduanya sama-sama dianggap valid oleh PHP.

LATIHAN

Pada latihan ini terdapat 2 file yaitu form-toko.html dan form-proses.php

form-toko.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/TR/xhtml1">  
    <head>  
        <title>Pembelian Online</title>  
    </head>  
    <body>  
        <h3>LUG ONLINE STORE</h3><hr />  
        <form action="form-proses.php" method="post">  
            <label>Sticker Linux (Rp7.500,-)</label><br />  
            <label>Jumlah:</label>  
            <input type="text" name="sticker" value="0" size="4" /><br /><br />  
            <label>Kaos Linux (Rp35.000,-)</label><br />  
            <label>Jumlah:</label>  
            <input type="text" name="kaos" value="0" size="4" /><br /><br />  
            <label>Jacket Linux (Rp35.000,-)</label><br />  
            <label>Jumlah:</label>  
            <input type="text" name="jacket" value="0" size="4" /><br /><br />  
            <input type="submit" value="P r o s e s" />  
        </form>  
    </body>  
</html>
```

form-proses.php

```
<?php
    // buat variabel untuk menampung data dari file form-toko.html
    $sticker = $_POST['sticker']; // data sticker
    $kaos = $_POST['kaos']; // data kaos
    $jacket = $_POST['jacket']; // data jacket

    // harga dari masing-masing produk
    define('HARGA_STCIKER', 7500);
    define('HARGA_KAOS', 35000);
    define('HARGA_JACKET', 55000);

    // total harga
    $total = (HARGA_STICKER * $sticker) + (HARGA_KAOS * $kaos)
(HARGA_JACKET * $jacket);
```

```

$diskon = 0;
$pesan_diskon = '0%';

// cek total untuk menentukan diskon
if ($total >= 50000 && $total < 75000)
{
    $diskon = 0.05; // diskon 5%
    $pesan_diskon = '5%';
}
elseif ($total >= 75000 && $total < 100000)
{
    $diskon = 0.1; // diskon 10%
    $pesan_diskon = '10%';
}
elseif ($total >= 100000)
{
    $diskon = 0.15; // diskon 15%
    $pesan_diskon = '15%';
}
$subtotal = $total - ($total * $diskon);

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/TR/xhtml1">
<head>
    <title>Pembelian Online</title>
    <style type="text/css">
        .tanda { background-color: #cfcfcf; }
    </style>
</head>
<body>
    <strong>LUG ONLINE STORE → DATA PEMBELIAN</strong><hr />
    <table border="1">
        <tr class="tanda">
            <th>Barang</th>
            <th>Jumlah</th>
            <th>Total</th>
        </tr>
        <tr>
            <td>Sticker</td>
            <td><?php echo $sticker;?></td>
            <td><?php echo ($sticker * HARGA_STCIKER);?>
        </tr>
        <tr>
            <td>Kaos</td>
            <td><?php echo $kaos;?></td>
            <td><?php echo ($kaos * HARGA_KAOS);?>
        </tr>
        <tr>
            <td>Jacket</td>
            <td><?php echo $jacket;?></td>
            <td><?php echo ($jacket * HARGA_JACKET);?>
        </tr>
        <tr class="tanda">

```

```

        <td colspan="2">TOTAL</td>
        <td><?php echo $total;?></td>
    </tr>
    <tr class="tanda">
        <td colspan="2">Diskon</td>
        <td><?php echo $pesan_diskon;?></td>
    </tr>
    <tr class="tanda">
        <td colspan="2">SUB-TOTAL</td>
        <td><?php echo $subtotal;?></td>
    </tr>
</table>
</body>
</html>

```

HASIL:

ANALISIS:

Barang	Jumlah	Total
Sticker	5	37500
Kaos	2	70000
Jacket	0	0
TOTAL		70000
Diskon		5%
SUB-TOTAL		66500

Statement switch...case...break

Sebenarnya statement switch...case...break sama dengan if...elseif... dimana kita dapat memilih lebih dari dua kondisi atau lebih. Syntax penulisan statement switch adalah sebagai berikut.

```

switch ($kondisi) {

    case kondisi_1:
        // kode yang dieksekusi

```

```
break;

case kondisi_2:
    // kode yang dieksekusi
    break;

case kondisi_3:
    // kode yang dieksekusi
    break;
default:
    // kode yang dieksekusi jika tidak ada yang memenuhi
    break;
}
```

Struktur Perulangan (Looping)

Statement for...

Bentuk perulangan pertama yang akan kita bahas adalah perulangan dengan for. Bentuk penulisan looping for adalah sebagai berikut.

```
for (ekspresi1; kondisi; ekspresi2) {
    // kode yang akan dieksekusi
}
```

LATIHAN:

```
bensin_for.php
```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/TR/xhtml1">
    <head>
        <title>Bensin For</title>
    </head>
    <body>
        <table cellpadding="4">
            <tr style="background-color:#cfcfcf">
                <th>Bensin (Liter)</th>
                <th>Harga (Rp)</th>
            </tr>
            <?php
                $harga = 4500; // semoga tidak naik lagi :p

                // lakukan loop dengan kelipatan 5 sampai dengan 25
                for ($liter = 5; $liter <= 25; $liter += 5)
                {
                    echo "<tr>\n";
                    echo "<td>$liter</td><td>" . ($harga * $liter);
                    echo "</tr>\n";
                }
            ?>
        </table>
    </body>
</html>

```

HASIL:

ANALISIS:

Statement while...

Syntax penulisan while adalah sebagai berikut.

```

while (kondisi) {

    // blok kode yang dieksekusi
}

```

LATIHAN

bensin_while.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/TR/xhtml1">
    <head>
        <title>Bensin While</title>
    </head>
    <body>
        <table cellpadding="4">
            <tr style="background-color:#cfcfcf">
                <th>Bensin (Liter)</th>
                <th>Harga (Rp)</th>
            </tr>
            <?php
                $harga = 4500; // semoga tidak naik lagi :p
                $liter = 5; // inisialisasi $liter diluar loop

                // lakukan loop dengan kelipatan 5 sampai dengan 25
                while ($liter <= 25)
                {
                    echo "<tr>\n";
                    echo "<td>$liter</td><td>" . ($harga * $liter
. "</td>\n";
                    echo "</tr>\n";
                    $liter += 5; // tambah counter
                }
            ?>
        </table>
    </body>
</html>

```

HASIL:

ANALISIS:

Statement do...while

Syntax dasar dari do...while adalah sebagai berikut.

```

do {
    // kode yang dieksekusi
}

while (kondisi)

```

LATIHAN:

bensin_do_while.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/TR/xhtml1">
<head>
    <title>Bensin Do While</title>
</head>
<body>
    <table cellpadding="4">
        <tr style="background-color:#cfcfcf">
            <th>Bensin (Liter)</th>
            <th>Harga (Rp)</th>
        </tr>
        <?php
            $harga = 4500; // semoga tidak naik lagi :p
            $liter = 5; // inisialisasi $liter diluar loop

            // lakukan loop dengan kelipatan 5 sampai dengan 25
            do {
                echo "<tr>\n";
                echo "    <td>$liter</td><td>" . ($harga * $liter)
            "</td>\n";
                echo "</tr>\n";
                $liter += 5; // tambah counter
            } while ($liter <= 25)
        ?>
    </table>
</body>
</html>
```

HASIL:

ANALISIS:

Berinteraksi dengan File

File merupakan salah satu aspek penting dalam sebuah aplikasi. Seringkali sebuah file dibuat untuk berbagai tujuan misalnya, menyimpan konfigurasi, temporary file, cache dan masih banyak lagi. Proses umum yang dilakukan adalah:

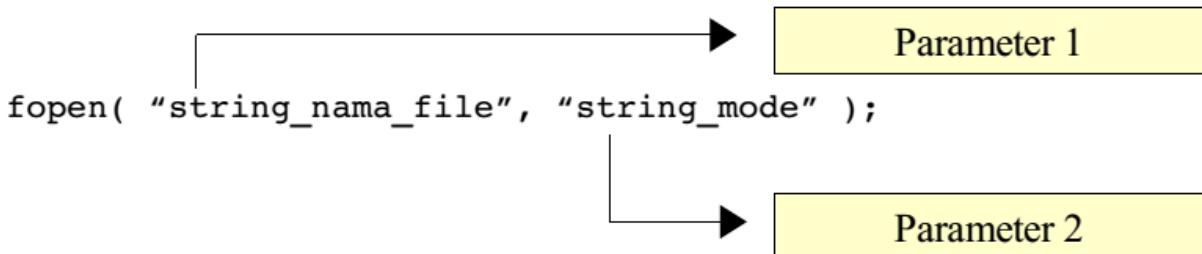
- Membuat resource dengan perintah fopen ()
- Menulis (fwrite ())/membaca file
- Menutup file fclose ()

Menyimpan File

Dibanding dengan penyimpanan pada database penyimpanan pada file relatif sederhana. Namun memiliki banyak kekurangan terutama jika menyangkut masalah keamanan dan kecepatan akses data. Pada PHP langkah-langkah untuk menyimpan sebuah file adalah:

- Membuka file dengan fopen()
- Menulis file dengan fwrite()
- Menutup file dengan fclose()

fopen() minimal memerlukan dua parameter yaitu 1) string nama file dan 2) string mode. Yang perlu diperhatikan disini adalah mode dari file. Syntax yang digunakan adalah:



Mode yang dapat digunakan pada fopen adalah:

Mode	Deskripsi
'r'	Buka file hanya untuk baca saja; pointer file diletakkan diawal file.
'r+'	Buka file untuk baca dan tulis; ponter file diletakkan diawal file.
'w'	Buka file untuk tulis saja; pointer diletakkan diawal file lalu truncate(overwrite); jika file tidak ada, buat file tersebut.
'w+'	Buka file untuk baca dan tulis; pointer diletakkan diawal file lalu truncate(overwrite); jika file tidak ada, buat file tersebut.
'a'	Buka file untuk tulis saja; pointer diletakkan diakhir file append); jika file tidak ada, buat file tersebut.
'a+'	Buka file untuk baca dan tulis; pointer diletakkan diakhir file append); jika file tidak ada, buat file tersebut.

LATIHAN

Pada latihan ini terdapat 3 file yaitu form-toko.html, form-proses.php dan faktur.php

form-toko.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/TR/xhtml1">
  <head>
    <title>Pembelian Online</title>
  </head>
  <body>
    <h3>LUG ONLINE STORE</h3><hr />
    <form action="form-proses.php" method="post">
      <label>Sticker Linux (Rp7.500,-)</label><br />
      <label>Jumlah:</label>
      <input type="text" name="sticker" value="0" size="4" /><br /><br />
      <label>Kaos Linux (Rp35.000,-)</label><br />
      <label>Jumlah:</label>
      <input type="text" name="kaos" value="0" size="4" /><br /><br />
      <label>Jacket Linux (Rp35.000,-)</label><br />
      <label>Jumlah:</label>
      <input type="text" name="jacket" value="0" size="4" /><br /><br />
      <input type="submit" value="P r o s e s" />
    </form>
  </body>
</html>
```

form-proses.php

```

<?php
    // buat variabel untuk menampung data dari file form-toko.html
    $sticker = $_POST['sticker']; // data sticker
    $kaos = $_POST['kaos']; // data kaos
    $jacket = $_POST['jacket']; // data jacket

    // harga dari masing-masing produk
    define('HARGA_STICKER', 7500);
    define('HARGA_KAOS', 35000);
    define('HARGA_JACKET', 55000);

    // total harga
    $total = (HARGA_STICKER * $sticker) + (HARGA_KAOS * $kaos
+ (HARGA_JACKET * $jacket));
    $diskon = 0;
    $pesan_diskon = '0%';

    // cek total untuk menentukan diskon
    if ($total >= 50000 && $total < 75000)
    {
        $diskon = 0.05; // diskon 5%
        $pesan_diskon = '5%';
    }
    elseif ($total >= 75000 && $total < 100000)
    {
        $diskon = 0.1; // diskon 10%
        $pesan_diskon = '10%';
    }
    elseif ($total >= 100000)
    {
        $diskon = 0.15; // diskon 15%
        $pesan_diskon = '15%';
    }
}

```

```

$subtotal = $total - ($total * $diskon);

// tulis ke file, mode yang digunakan adalah 'a' => append
// file faktur.txt akan dibuat otomatis jika belum ada
$fp = fopen('faktur.txt', 'a');
$isi_file = "$sticker Sticker -#- " . (HARGA_STICKER * $sticker) . "
#- ".
"$kaos Kaos -#- " . (HARGA_KAOS * $kaos) . " -#- ".
"$jacket Jacket -#- " . (HARGA_JACKET * $jacket) . " -#- ".
"$total -#- $pesan_diskon -#- $subtotal\n";
fwrite($fp, $isi_file);
fclose($fp);
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/TR/xhtml1">
<head>
    <title>Pembelian Online</title>
    <style type="text/css">
        .tanda { background-color: #cfcfcf; }
    </style>
</head>
<body>
    <strong>LUG ONLINE STORE → DATA PEMBELIAN</strong><hr />
    <table border="1">
        <tr class="tanda">
            <th>Barang</th>
            <th>Jumlah</th>
            <th>Total</th>
        </tr>
        <tr>
            <td>Sticker</td>
            <td><?php echo $sticker;?></td>
            <td><?php echo ($sticker * HARGA_STCIKER);?>
        </tr>
        <tr>
            <td>Kaos</td>
            <td><?php echo $kaos;?></td>
            <td><?php echo ($kaos * HARGA_KAOS);?>
        </tr>
        <tr>
            <td>Jacket</td>
            <td><?php echo $jacket;?></td>
            <td><?php echo ($jacket * HARGA_JACKET);?>
        </tr>
        <tr class="tanda">
            <td colspan="2">TOTAL</td>
            <td><?php echo $total;?></td>
        </tr>
        <tr class="tanda">
            <td colspan="2">Diskon</td>
            <td><?php echo $pesan_diskon;?></td>
        </tr>
        <tr class="tanda">
            <td colspan="2">SUB-TOTAL</td>
            <td><?php echo $subtotal;?></td>
        </tr>
    </table>
    <p><a href="faktur.php">Lihat Faktur</a></p>
</body>
</html>

```

faktur.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/TR/xhtml1">
<head>
    <title>Pembelian Online</title>
</head>
<body>
    <h2>Data Faktur Pembelian</h2>
    <pre>
        <?php
            // baca seluruh isi file
            readfile('faktur.txt');
        ?>
    </pre>
    <a href="form-toko.html">Kembali ke Form</a>
</body>
</html>
```

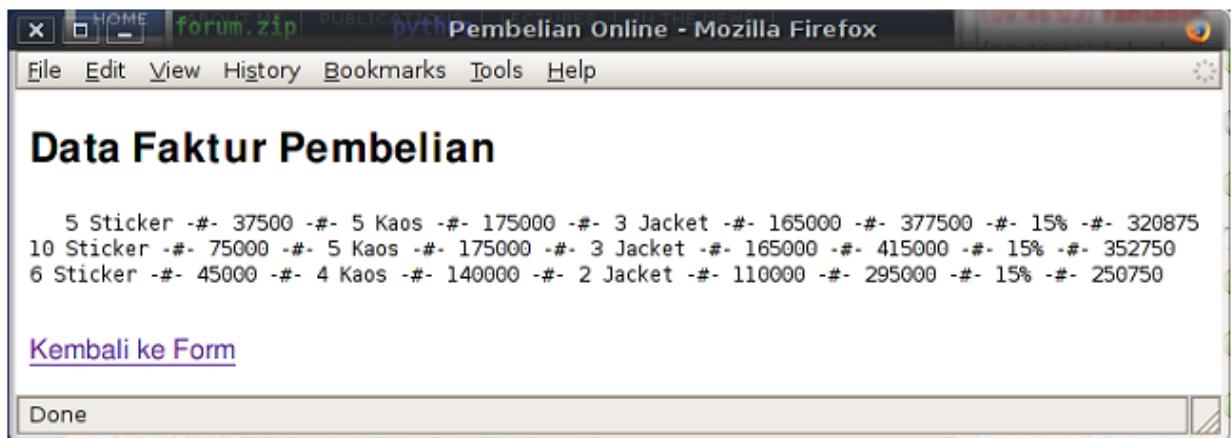
HASIL :

ANALISIS :

The screenshot shows a Mozilla Firefox window with the title "Pembelian Online - Mozilla Firefox". Below the title bar is a menu bar with File, Edit, View, History, Bookmarks, Tools, and Help. The main content area has a heading "LUG ONLINE STORE - DATA PEMBELIAN". Below the heading is a table with the following data:

Barang	Jumlah	Total
Sticker	6	0
Kaos	4	140000
Jacket	2	110000
TOTAL		295000
Diskon		15%
SUB-TOTAL		250750

Below the table is a blue link labeled "Lihat Faktur". At the bottom of the page is a grey footer bar with the word "Done".



Fungsi PHP-MySQL

Tujuan Praktikum

1. Memahami fungsi yang menghubungkan PHP dengan database MySQL
2. Mampu menggunakan query dari MySQL ke dalam sintak PHP

Pengenalan Fungsi PHP-MySQL

Fungsi PHP-MySQL adalah suatu fungsi yang menjembatani antara PHP sebagai Programming web server, dan MySQL sebagai database, sehingga data-data yang terdapat pada database MySQL dapat ditampilkan pada browser.

Fungsi PHP-MySQL antara lain, adalah:

A. **MYSQL_CONNECT ()**

Berfungsi untuk menghubungkan ke server MySQL. Fungsi ini merupakan fungsi awal yang akan dijalankan sebelum melakukan fungsi-fungsi lain. Selain mysql_connect() terdapat juga fungsi penghubung ke server mysql lain , yaitu mysql_pconnect().

Bentuk umum penulisannya:

```
mysql_connect (host, user, password);
```

Atau

```
mysql_pconnect (host, user, password);
```

Contoh:

```
<?
$cek=mysql_connect("localhost"," root","");
If($cek){
    echo "Database berhasil terkoneksi";
}else{
    echo "Database tidak berhasil terkoneksi";
}
?>
```

Untuk password dapat tidak disisipi apabila memang tidak terdapat password pada record yang bersangkutan di tabel user tersebut. Apabila terjadi pesan “Database tidak berhasil terkoneksi” pada browser, maka lakukan pengecekan apakah server mysql sudah diaktifkan atau belum.

B. **MYSQL_CREATE_DB()**

Fungsi yang satu ini hanya merupakan alternatif, namun sangat jarang digunakan, karena pada umumnya setiap programmer web, lebih cenderung dalam membuat database pada console atau command prompt atau phpmyadmin pada browser. Namun tidak ada salahnya jika ingin mengetahui fungsi dari mysql_create_db() itu sendiri. Fungsi dari mysql_create_db() adalah sebagai proses untuk membuat database.

Bentuk umum penulisannya:

```
Mysql_create_db(nama_database);
```

Contoh:

```
<?
mysql_connect ("localhost"," root","");
$cek=mysql_create_db("lat_database");
if($cek) {
echo "Database lat_database berhasil dibuat";
} else {
echo "Database lat_database gagal dibuat";
}
?>
```

C. MYSQL_SELECT_DB ()

Mysql_select_db() berfungsi untuk mengaktifkan database yang telah tersedia pada mysql server.

Bentuk umum penulisannya:

```
mysql_select_db(nama_database);
```

Contoh:

```
<?
mysql_connect ("localhost"," root","");
mysql_select_db("lat_database");
?>
```

D. MYSQL_QUERY ()

Berfungsi untuk memasukkan perintah-perintah SQL, baik yang bersifat DDL atau DML. Dengan demikian perintah-perintah SQL tersebut dapat dijalankan pada browser.

Bentuk umum penulisannya:

```
mysql_query (perintah-perintah SQL);
```

Contoh:

```

<?
    mysql_connect ("localhost","root","");
    mysql_select_db("lat_database");
    $input=mysql_query ("insert      into      tabel_mahasiswa(nama,alamat)
values('Anita','Jl.Probolinggo no.59')");
?>

```

E. MYSQL_DB_QUERY ()

Jika pada mysql_query () hanya berfungsi untuk memasukkan perintah-perintah SQL saja, beda halnya dengan mysql_db_query() yang dapat menyatukan dua fungsi, yaitu mysql_select_db() dan mysql_query(). Jika telah menggunakan fungsi mysql_db_query () tidak perlu lagi menggunakan fungsi mysql_select_db () .

Bentuk umum penulisannya:

```
mysql_db_query(nama_database,perintah-perintah SQL);
```

Contoh:

```

<?
    mysql_connect ("localhost","root","");
    $input=mysql_db_query      ("lat_database","insert      into
tabel_mahasiswa(nama,alamat) values('Anita','Jl.Probolinggo no.59')");
?>

```

F. MYSQL_NUM_ROWS ()

Dengan fungsi mysql_num_rows () dapat mengetahui berapa jumlah record terkini.

Bentuk umum penulisannya:

```
Mysql_num_rows(variable);
```

Contoh:

```

<?
    mysql_connect("localhost","root","");
    mysql_select_db("lat_database");
    $hasil=mysql_query("select * from tabel_mahasiswa");
    $hit=mysql_num_rows($hasil);
?>

```

G. MYSQL_FETCH_ROW()

Berfungsi untuk mengambil data per record atau per baris dalam suatu table. Yang perlu diperhatikan sifat pengambilan record pada fungsi mysql_fetch_row() adalah dengan system urutan nomor index field.

Misalnya terdapat 3 field dalam suatu tabel, yaitu no, nama dan alamat, maka susunan indexnya adalah No berada pada index ke-0, nama berindex 1 dan alamat berindex 2.

Perhatikan contoh berikut:

```
<?
mysql_connect ("localhost"," root","");
mysql_select_db("lat_database");
$hasil=mysql_query ("select * from tabel_mahasiswa");
While($data=mysql_fetch_row($hasil)) {
    echo "$data[0] $data[1]<br>$data[2]<br>";
}
```

Hasil saat di browser:

1.Anita
Jl. Probolinggo no.59
2.Tyas
Jl. Merdeka no.103

H. MYSQL_FETCH_ARRAY()

Mysql_fetch_array () mempunyai fungsi yang sama dengan mysql_fetch_row(), yaitu untuk mengambil data per record. Yang membedakan adalah jika pada fungsi mysql_fetch_row() menggunakan nomor urut index, tetapi pada fungsi mysql_fetch_array() yang digunakan adalah nama fieldnya.

Perhatikan contoh berikut:

```
<?
mysql_connect ("localhost"," root","");
mysql_select_db("lat_database");
$hasil=mysql_query ("select * from tabel_mahasiswa");
While($data=mysql_fetch_array($hasil)) {
    echo "$data[no] $data[nama]<br>$data[alamat]<br>";
}
```

Hasil saat di browser:

1.Anita
Jl. Probolinggo no.59
2.Tyas

Daftar Fungsi-fungsi PHP-MySQL

Berikut ini merupakan table daftar fungsi-fungsi PHP-MySQL, yaitu:

Fungsi	Kegunaan	Penulisan Umum
mysql_affected_rows	Menghitung jumlah record dalam suatu koneksi	mysql_affected_rows(x)
mysql_close	Untuk menutup koneksi database	mysql_close(x)
mysql_connect	Untuk mengaktifkan koneksi ke server mysql	mysql_connect(host, user, password)
mysql_create_db	Berfungsi untuk membuat database	mysql_create_db (nama_database)
mysql_data_seek	Untuk meletakkan posisi pointer	mysql_data_seek (result_ID, nmr_row)
mysql_db_query	Berfungsi untuk menuliskan perintah sql berikut dengan database	mysql_db_query (nama_database, perintah sql)
mysql_drop_db	Untuk menghapus database	mysql_drop_db (nama_database)
mysql_errno	Menampilkan Nomor pesan kesalahan	mysql_errno (link_ID)
mysql_error	Untuk menampilkan pesan kesalahan	mysql_error
mysql_fetch_array	Untuk mengambil nilai per-record dengan sistem nama field table	mysql_fetch_array(x)
mysql_fetch_field	Mengambil informasi field dari hasil query	mysql_fetch_field(x)
mysql_fetch_lengths	Mencari panjang setiap field dari hasil query	mysql_fetch_lengths(x)
mysql_fetch_object	Berfungsi untuk mengambil nilai per-record dengan sistem object	mysql_fetch_object(x)
mysql_fetch_row	Berfungsi untuk mengambil nilai per-record dengan sistem urutan index dari suatu field	mysql_fetch_row(x)
mysql_fetch_name	Berfungsi untuk menentukan nama field dari hasil query	mysql_fetch_name (result_ID, field_index)
mysql_fetch_seek	Untuk meletakkan pointer field offset yang ditunjuk	mysql_fetch_seek (result_ID, field_offset)

mysql_fetch_table	Berfungsi untuk menentukan nama table dari field yang ditunjuk	mysql_fetch_table (result_ID, field_offset)
mysql_fetch_type	Berfungsi untuk menentukan nama field yang ditunjuk	mysql_fetch_type (result_ID, field_offset)
mysql_free_result	Berfungsi untuk menghapus hasil query dari memori	mysql_free_result (result_ID)
mysql_get_client_info	Berfungsi untuk mendapatkan informasi client MySQL	mysql_get_client_info(x)
mysql_get_host_info	Berfungsi untuk mendapatkan informasi host MySQL	mysql_get_host_info(x)
mysql_get_server_info	Berfungsi untuk mendapatkan informasi server MySQL	mysql_get_server_info(x)
mysql_insert_id	Untuk mendapatkan ID dari operasi insert sebelumnya	mysql_insert_id(x)
mysql_num_fields	Berfungsi untuk menentukan jumlah field dari hasil query	mysql_num_fields(x)
mysql_num_rows	Berfungsi untuk menentukan jumlah record atau baris dari hasil query	mysql_num_rows(x)
mysql_pconnect	Berfungsi untuk melakukan koneksi ke server mysql secara persistem	mysql_pconnect (host, user, password)
mysql_query	Berfungsi untuk memasukkan perintah sql	mysql_query(x)
mysql_result	Berfungsi untuk mengambil data dari perintah query	mysql_result (result_ID, row)
mysql_select_db	Berfungsi untuk mengaktifkan database	mysql_select_db (nama_database)
mysql_tablename	Berfungsi untuk mendapatkan nama table dari suatu kolom	mysql_tablename (x)
mysql_unbuffered_query	Untuk mengirim perintah SQL ke server, tanpa menampilkan hasil	mysql_unbuffered_query(x)

MEMBUAT APLIKASI LOGIN DENGAN MENGGUNAKAN BASIS DATA

Membuat Form Login

Buku tamu terdiri dari 2 file yaitu:

- File Index.html
- File PeriksaLogin.php

Menyiapkan Database

Nama Database : Web

Nama Tabel : admin

Struktur Tabel:

#	Name	Type
1	nama	varchar(50)
2	username	varchar(50)
3	password	varchar(50)

Isi Tabel:

nama	username	password
Ani	1	1

Index.php

```
<html>
    <head>
        <title>Form Login</title>
    </head>
    <body>
        <form name="login" id="login" action="periksalogin.php" method="post">
            Username: <input type="text" name="userdariform"> <br>
            Password: <input type="text" name="passdariform"> <br>
            <input type="submit" value="OK">
        </form>
    </body>
</html>
```

PeriksaLogin.php

```
<?php
$host="localhost";
$usernameserver="root";
$passwordserver="";
$database="web";

$koneksi=mysqli_connect($host,$usernameserver,$passwordserver,$database)
or die("tidak bisa nyambung ke mysql");
```

```

$userdariform=$_POST['userdariform'];
$passdariform=$_POST['passdariform'];

if(strlen($userdariform)==0)
{
    echo("username tidak ada");
}
else if(strlen($passdariform)==0)
{
    echo("password tidak ada");
}
else if($koneksi)
{
    $query ="select username, password from admin where username = '" .
$userdariform."' AND password = '". $passdariform ."'";
$result = mysqli_query ($koneksi,$query);
$rowset=mysqli_fetch_row($result);
if($rowset)
{
    echo("Selamat datang");
}
else
{
    echo("username dan password salah");
}
}
?>

```



Username :
 Password :



Username :
 Password :



Selamat datang

SOAL

Tambahkan aplikasi diatas dengan ketentuan

Jika login berhasil dilakukan maka akan lanjut ke halaman pengisian buku tamu

Jika login gagal maka akan kembali lagi ke form login

MEMBUAT APLIKASI BUKU TAMU DENGAN MENGGUNAKAN BASIS DATA

Membuat Buku Tamu

Buku tamu terdiri dari 3 file yaitu :

- o File Index.html
- o File Simpan_bukutamu.php
- o File Tampil_bukutamu.php

Menyiapkan Database

Nama Database : Web

Nama Tabel : bukutamu

Field	Type	Collation	Attributes	Null	Default	Extra
id_btamu	int(3)			No		auto_increment
nama	varchar(35)	latin1_swedish_ci		No		
email	varchar(60)	latin1_swedish_ci		No		
pesan	text	latin1_swedish_ci		Yes	NULL	
tanggal	date			Yes	NULL	

Index.html

```
<html>
<head>
    <title>Form Buku Tamu</title>
</head>
<body>
    <form name = "form1" method="post" action="simpan_bukutamu.php">
        <table>
            <tr>
                <td colspan="2">Form BUKU TAMU</td>
            </tr>
            <tr>
                <td>Nama</td>
                <td><input type="text" name="fm_nama" id="fm_nama"></td>
            </tr>
            <tr>
                <td>Email</td>
                <td><input type="text" name="fm_email" id="fm_email"></td>
            </tr>
            <tr>
                <td>Pesanan</td>
                <td><input type="text" name="fm_pesan" id="fm_pesan"></td>
            </tr>
            <tr>
                <td><input type="submit" name="submit" value="simpan"></td>
            </tr>
        </table>
    </form>
</body>
```

```

</html>
Simpan_bukutamu.php

<?php

$nama=$_POST['fm_nama'];
$email=$_POST['fm_email'];
$pesan=$_POST['fm_pesan'];
$tanggal=date('Y-m-d');

$host="localhost";
$usernameserver="root";
$passwordserver="";
$database="web";

$koneksi=mysqli_connect($host,$usernameserver,$passwordserver,$database)
or die("tidak bisa nyambung ke mysql");

if(strlen($nama)==0)
{
    echo("Nama belum diisi");
}
else if(strlen($email)==0)
{
    echo("Email belum diisi");
}
else if(strlen($pesan)==0)
{
    echo("Pesanan belum diisi");
}
else if($koneksi)
{
    $query="INSERT      INTO      bukutamu      (nama,email,pesan,tanggal)VALUES
('$nama','$email','$pesan','$tanggal')
";
    $result=mysqli_query($koneksi,$query);
    include"tampil_bukutamu.php";
}
?>
```

```

Tampil_bukutamu.php

<html>
<head>
    <title>FORM BUKU TAMU</title>
</head>
<body>
    <table>
        <tr>
            <td cols="2">Daftar pengisian buku tamu</td>
        </tr>
        <?php
        $host="localhost";
        $usernameserver="root";
        $passwordserver="";
        $database="web";

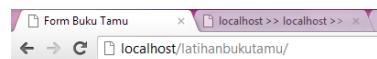
        $koneksi=mysqli_connect($host,$usernameserver,$passwordserver,$database)
        or die("tidak bisa nyambung ke mysql");

        $query="select * from bukutamu";
        $result= mysqli_query($koneksi,$query);
        if($result)
        {
            while($hasil = mysqli_fetch_array($result))
```

```

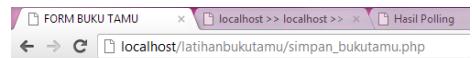
{
    extract($hasil);
?>
<tr>
    <td>Nama</td>
    <td><?php echo "$hasil[nama]";?></td>
</tr>
<tr>
    <td>Email</td>
    <td><?php echo "$hasil[email]";?></td>
</tr>
<tr>
    <td>pesan</td>
    <td><?php echo "$hasil[pesan]";?></td>
</tr>
<tr>
    <td>tanggal</td>
    <td><?php echo "$hasil[tanggal]";?></td>
</tr>
<?php } ?>
</table>
</body>
</html>

```



Form BUKU TAMU

Nama	<input type="text"/>
Email	<input type="text"/>
Pesan	<input type="text"/>
<input type="button" value="simpan"/>	



Daftar pengisian buku tamu	
Nama	Ani
Email	ani@yahoo.com
pesan	rajin rajin belajar
tanggal	2012-10-03

SOAL

Buatlah program yang menerima input berupa nim, nama, alamat, no telp, email, hobby. Kemudian buat output yang menampilkan seluruh data yang telah diinputkan. Gunakan database latihan, buat tabel baru dengan nama mahasiswa, jadikan nim sebagai primary key-nya. Selamat bekerja!

MEMBUAT APLIKASI SEARCHING DENGAN MENGGUNAKAN BASIS DATA

Membuat Form Searching

Program mesin pencari sederhana ini terdiri dari 2 file:

- o Index.html
- o Proses.php

Mempersiapkan Database

Nama Database : Latihan

Nama Tabel : searching

Field	Type	Collation	Attributes	Null	Default	Extra
nama	varchar(25)	latin1_swedish_ci		No		
alamat	varchar(25)	latin1_swedish_ci		No		
jabatan	varchar(25)	latin1_swedish_ci		No		
pendidikan	varchar(25)	latin1_swedish_ci		No		

nama	alamat	jabatan	pendidikan
Muhammad Amin	Tanggul Angin	Manajer Bisnis	Teknik Elektro
Okty Rully	Lumajang	Sekretaris	Teknik Elektro
Dedi Gunawan	Banyumanik Semarang	Manajer NOC	Telekomunikasi Polines
Amin Setiaji	Kediri	Manajer Transmisi	Teknik Listrik
Ryo Handoyo	Malang	Manajer Teknik	Teknik Elektro
Gunawan	Rungkut Suroboyo	Teknisi	Teknik Elektro

Index.php

```
<html>
    <head>
        <title>Pencarian</title>
    </head>
    <body>
        <h1>Pencarian Pegawai</h1>
        <form action="proses.php" method="POST">
            <input type="text" name="nama_pegawai">
            <input type="submit" value="search">
        </form>
    </body>
</html>
```

Proses.php

```
<?php
$namapegawai=$_POST['nama_pegawai'];
```

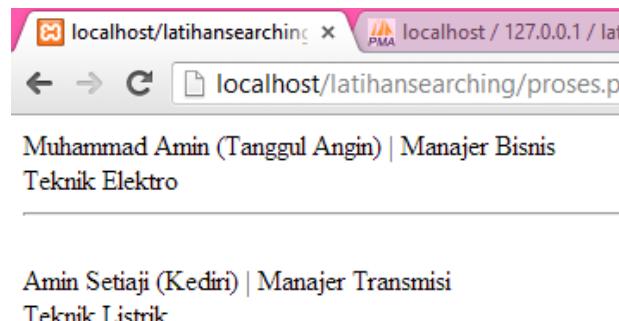
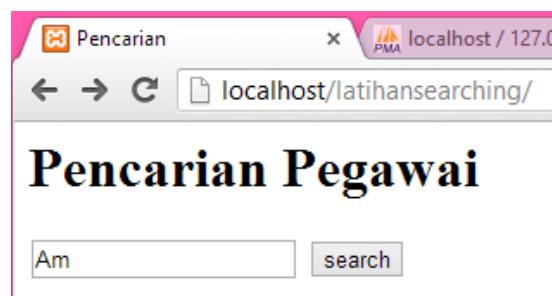
```

$host="localhost";
$usernameserver="root";
$passwordserver="";
$database="web";

$koneksi=mysqli_connect($host,$usernameserver,$passwordserver,$database)
or die("tidak bisa nyambung ke mysql");

$query="SELECT * FROM `searching` WHERE nama LIKE '%$namapegawai%'";
$result=mysqli_query($koneksi,$query);
if($result)
{
    while($hasil=mysqli_fetch_array($result))
    {
        extract ($hasil);
        echo "$hasil[nama] ($hasil[alamat]) | $hasil[jabatan] <br>
$hasil[pendidikan] <br><hr><br>";
    }
}
?>

```



SOAL

Buatlah mesin pencari (search engine) mahasiswa yang menerima input berupa nama mahasiswa dan menampilkan output berupa informasi nim, nama dan alamat mahasiswa yang bersangkutan.

MEMBUAT APLIKASI POLLING DENGAN MENGGUNAKAN BASIS DATA

Membuat Jejak Pendapat

Program Jejak Pendapat terdiri dari 3 file yaitu :

- o Index.html
- o Proses.php
- o Tampil_polling.php

Mempersiapkan Database

Nama Database : Latihan

Nama Tabel : polling

Field	Type	Collation	Attributes	Null	Default	Extra
pilihan	varchar(25)	latin1_swedish_ci		No		
jumlah	tinyint(3)			No	0	

pilihan	jumlah
HarryPotter	0
ResidenceEvil	0
AliceInWonderland	0
TheAvangers	0
SpyKids	0

Index.html

```
<html>
    <head>
        <title>Polling</title>
    </head>
    <body>
        <form action="proses.php" method="post">
        Film Favorit kamu ?
        <br><input type="radio" name="film" value="HarryPotter"> Harry Potter
1-6
        <br><input type="radio" name="film" value="ResidenceEvil"> Residence
        Evil
        <br><input type="radio" name="film" value="AliceInWonderland"> Alice
        in wonderland
        <br><input type="radio" name="film" value="TheAvangers"> The avangers
        <br><input type="radio" name="film" value="SpyKids"> Spy Kids
        <br><input type="submit" value="Proses">
    </form>
    </body>
</html>
```

Proses.php

```
<?php
    $film=$_POST['film'];

    $host="localhost";
    $usernameserver="root";
    $passwordserver="";
    $database="web";

    $koneksi=mysqli_connect($host,$usernameserver,$passwordserver,$database)
        or die("tidak bisa nyambung ke mysql");

    $query="select * from polling where pilihan='$film'";
    $result=mysqli_query($koneksi,$query);
    if($result)
    {
        while($hasil= mysqli_fetch_array($result))
        {
            extract($hasil);
            $jumlah = $hasil[jumlah];
            $tambahkan = $jumlah+1;
            $tambah="UPDATE      polling      set      jumlah='$tambahkan'      where
pilihan='$film'";
            $result=mysqli_query($koneksi,$tambah) or die("error");
        }
    }
    include("tampil_polling.php");
?>
```

Tampil_polling.php

```
<html>
<head>
    <title>Hasil Polling</title>
</head>
<body>
    Film pilihan User : <br>
    <?php
        $host="localhost";
        $usernameserver="root";
        $passwordserver="";
        $database="web";

        $koneksi=mysqli_connect($host,$usernameserver,$passwordserver,$database)
            or die("tidak bisa nyambung ke mysql");

        $query="select * from polling";
        $result=mysqli_query($koneksi,$query) or die("error");
        if($result)
        {
            while ($hasil=mysqli_fetch_array($result))
            {
                extract ($hasil);
                echo "$hasil[pilihan] = $hasil[jumlah] <br>";
            }
        }
    ?>
</body>
</html>
```

FORM BUKU TAMU

localhost >> localhost

localhost/latihanpolling/

Film Favorit kamu ?

Harry Potter 1-6
 Residence Evil
 Alice in wonderland
 The avangers
 Spy Kids

Proses

FORM BUKU TAMU

localhost >> localhost >> localhost

localhost/latihanpolling/proses.php

Film pilihan User Harry Potter 1-6 = 5
Residence Evil = 3
Alice in wonderland = 1
The avangers = 0
Spy Kids = 1

SOAL

Buatlah polling/jajak pendapat yang meminta masukan pengguna mengenai penilaian suatu website. Terdiri dari 4 kategori penilaian yaitu buruk, cukup, baik dan sangat baik.

CODEIGNITER

Tujuan Praktikum

1. Memahami cara penggunaan codeigniter
2. Mampu mengimplementasikannya codeigniter menjadi sebuah halaman web.

Pengenalan CodeIgniter

Secara *default*, URL pada CodeIgniter didesain agar *search-engine* dan *human friendly*, menggunakan pendekatan *segment-based*.

example.com/index.php/class/function/parameter1/parameter2

- a. *Index.php* merupakan segment ke-0.
- b. Segment pertama merepresentasikan *class controller* yang diakses.
- c. Segment kedua merepresentasikan *nama method* yang dipanggil pada class tersebut.
- d. Segment ketiga dan seterusnya bersifat *optional*, merepresentasikan *parameter masukan* untuk fungsi yang dipanggil tersebut.

Index.php dapat dihilangkan dengan .htaccess sederhana. Jika ingin menghilangkan *index.php*, sebaiknya lakukan diawal sebelum *source-code* aplikasi dibuat, karena jika aplikasi sudah jadi dan *index.php* dihilangkan, dapat merubah struktur *link* seluruh aplikasi, contohnya menu, dan *paging*.

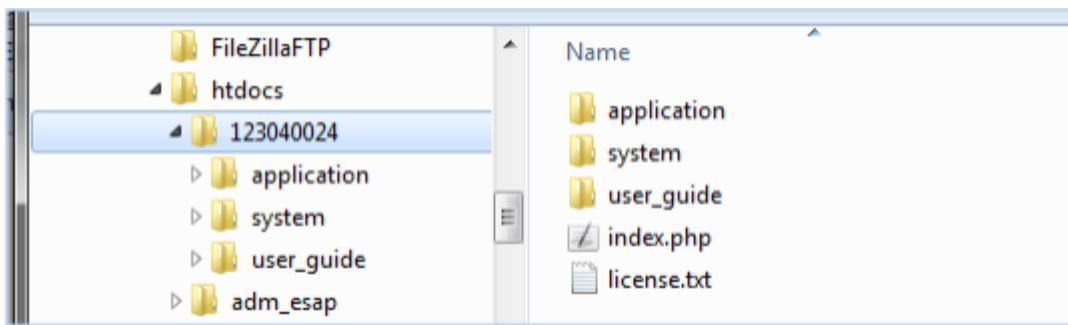
Menggunakan CodeIgniter

1. Download CI

CodeIgniter dapat di-**download** gratis di website resminya yaitu www.codeigniter.com.

2. Instalasi

Extract file CodeIgniter yang sudah di-**download** sebelumnya ke direktori *htdocs*, kemudian ganti nama folder **CodeIgniter_2.x.x** tersebut dengan **NIM Anda** yang selanjutnya akan menjadi direktori kerja Anda, sehingga seperti ini:



Untuk menguji apakah instalasi sudah benar dan membuktikan salah satu keunggulan CI (hampir tidak ada konfigurasi awal), silahkan akses direktori kerja Anda pada browser. Contoh: <http://localhost/123040024>. Jika sukses maka akan tampil halaman pembuka CodeIgniter seperti di bawah ini:

User Guide'. At the bottom right, it says 'Page rendered in 0.0138 seconds'."/>

Welcome to CodeIgniter!

The page you are looking at is being generated dynamically by CodeIgniter.

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

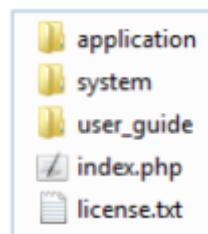
The corresponding controller for this page is found at:
application/controllers/welcome.php

If you are exploring CodeIgniter for the very first time, you should start by reading the [User Guide](#).

Page rendered in 0.0138 seconds

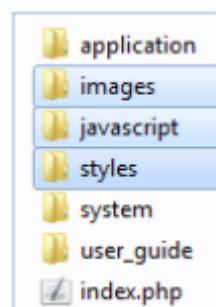
3. Struktur Folder

Berikut akan dibahas fungsi masing-masing folder dan file default dari CodeIgniter. Dimulai dari root direktori kerja Anda.

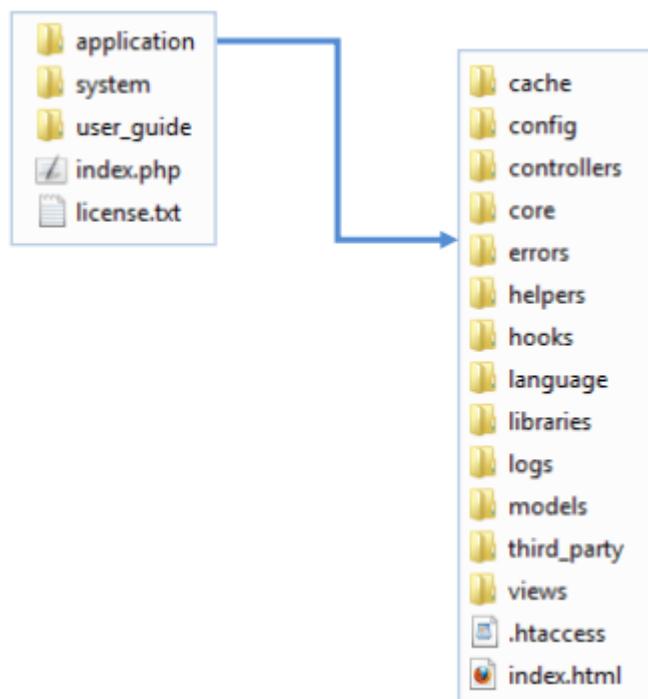


- a. *Application*, berisi folder dan file yang terkait dengan aplikasi yang akan dibuat. Kita akan banyak bekerja pada direktori *application* ini.
- b. *System*, berisi file-file inti framework CodeIgniter. Kita tidak akan mengubah apapun pada direktori *system* ini.
- c. *User_guide*, berisi file-file dokumentasi CI yang dapat diakses melalui browser.
- d. *Index.php*, file konfigurasi awal CodeIgniter

Aplikasi yang akan Anda bangun tentunya menggunakan JavaScript, gambar dan CSS. Anda dapat buat direktori untuk masing-masing sumber daya tersebut pada root aplikasi (setingkat dengan file index.php), sehingga seperti contoh berikut:



3.1. Direktori Application



Berikut penjelasan direktori-direktori utama yang akan sering kita gunakan.

- a. Config, berisi file-file yang terkait konfigurasi dasar aplikasi.

- b. Controllers, berisi file-file controller dari aplikasi yang dibangun. Simpan seluruh file controller aplikasi pada folder ini.
- c. Core, jika ingin memperluas class-class inti CodeIgniter, maka simpan file class yang Anda kembangkan tersebut pada direktori ini.
- d. Helpers, berisi file-file helper yang Anda buat sendiri.
- e. Libraries, berisi file-file library yang Anda buat sendiri.
- f. Logs, jika error log pada config diaktifkan maka file error log akan tersimpan pada direktori ini.
- g. Models, berisi file-file model dari aplikasi yang dibangun. Simpan seluruh file model aplikasi pada folder ini.
- h. Views, berisi file-file view dari aplikasi yang dibangun. Simpan seluruh file view aplikasi pada folder ini.

4. Apa Saja yang Harus Dikonfigurasi untuk Mulai Membangun Aplikasi?

Pada setiap pembangunan aplikasi tentunya diperlukan beberapa konfigurasi tambahan yang mendukung aplikasi tersebut, misalnya konfigurasi error_reporting, database, security, dan controller default aplikasi. Berikut akan dijelaskan beberapa file konfigurasi minimal untuk mulai membangun aplikasi (silahkan buka masing-masing file yang ada di penjelasan berikut ini):

- a. Index.php. Sebenarnya pada tahap development tidak ada konfigurasi yang diubah pada file ini, tapi jika aplikasi akan di-publish / deploy maka error reporting harus dimatikan, silahkan lihat di baris 21 ada konfigurasi ENVIRONMENT. Default-nya terdapat 3 nilai konstanta ENVIRONMENT, yaitu development, testing, dan production. Setiap nilai tersebut berpengaruh pada error_reporting aplikasi. Selain ENVIRONMENT, pada file index.php juga terdapat beberapa konfigurasi lain seperti nama direktori application dan system, ekstensi file aplikasi yang bisa diubah sesuai kebutuhan.

Selanjutnya buka direktori [application/config/](#)

- b. Autoload.php, pada file ini didefinisikan apa saja yang harus dipanggil otomatis oleh CI tanpa harus dideklarasikan disetiap fitur aplikasi. Yang dapat otomatis dipanggil adalah:

- 1). Packages
- 2). Libraries

- 3). Helper
- 4). Custom config, jika Anda membuat file konfigurasi diluar file default CodeIgniter.
- 5). Language
- 6). Model.

Kita dapat mendefinisikan apa saja yang harus dipanggil otomatis oleh CI sesuai kebutuhan, misalnya library **session**, **database**, **security**, helper **form** dan **url**.

- c. **Config.php**, pada file ini didefinisikan macam-macam konfigurasi utama aplikasi. Diantaranya yang biasanya kita ubah pada setiap project adalah:

- 1) **Base url**, berfungsi sebagai URL utama aplikasi yang dibangun. Jika tidak didefinisikan CI akan mendeteksi otomatis URL dan Domain yang dipakai aplikasi.

Pada modul 1 ini silahkan isi dengan URL kerja Anda atau biarkan kosong tidak masalah.

- 2) **Index page**, jika file index diganti namanya maka sesuaikan konfigurasi ini, jika index.php dihilangkan dari URL menggunakan .htaccess maka kosongkan nilai **\$config['index_page']**.

Pada modul 1 ini tetap index.php saja.

- 3) **Url suffix**, jika ingin menambahkan akhiran ekstensi file di URL. Misal example.com/news/show/231.html

Pada modul 1 ini Anda silahkan menentukan sendiri url_suffix yang diinginkan.

- 4) **Language**, bahasa yang akan digunakan pada aplikasi. Kita dapat merubah atau membuat bahasa sendiri untuk sistem CI pada direktori root: system/language/.

Pada modul 1 ini tetap ‘english’ saja

- 5) **Encryption key**, jika aplikasi menggunakan **session**, maka **\$config['encryption_key']**

harus didefinisikan. Pada modul 1 ini kita tidak menggunakan session.

- 6) **Session**, konfigurasi untuk session:

1. [**'sess_cookie_name'**]: nama cookie aplikasi yang dibangun, bisa diubah.
2. [**'sess_expiration'**]: waktu kadaluarsa session, satunya detik.
3. [**'sess_expire_on_close'**]: jika bernilai TRUE, maka session **expire** ketika browser ditutup.
4. [**'sess_encrypt_cookie'**]: beri nilai TRUE jika cookie ingin dienkripsi.

5. [‘sess_use_database’]: beri nilai TRUE jika ingin menggunakan database untuk menyimpan data session.
6. [‘sess_table_name’]: nama table penyimpanan data session, bisa diubah.
7. [‘sess_match_ip’]: beri nilai TRUE jika ingin membanding IP address setiap membaca data session.
8. [‘sess_match_useragent’]: beri nilai TRUE untuk membandingkan browser pengguna setiap membaca data session.
9. [‘sess_time_to_update’]: waktu (dalam detik) CI harus memperbarui data session yang disimpannya.

Pada saat ini tidak ada konfigurasi session yang diubah.

- 7) **XSS Filtering**, mendefinisikan apakah setiap data yang dikirimkan oleh browser / pengguna (POST, GET, COOKIE) diperiksa dulu dari bahaya **hacking** metode XSS

(Cross Site Scripting). Sebaiknya nilai `$config['global_xss_filtering']` bernilai **TRUE**.

- 8) **CSRF Protection**, mendefinisikan apakah setiap data yang dikirimkan melalui form masih masuk waktu SESSION atau tidak dan memastikan bahwa data dikirimkan oleh browser yang mengakses halaman web tersebut, jika tidak maka data akan ditolak. Konfigurasi ini untuk menghindari bahaya **hacking** metode Cross Site Request Forgery. Sebaiknya nilai `$config['csrf_protection']` bernilai **TRUE**.

- d. **Database.php**, pada file ini mendefinisikan parameter-parameter yang dibutuhkan untuk koneksi ke database. CodeIgniter mendukung koneksi ke beberapa produk DBMS, bahkan mendukung koneksi ke beberapa database dan DBMS sekaligus pada satu aplikasi. Berikut penjelasan beberapa konfigurasi:

- 1) **Active group**, isikan nama konfigurasi database default yang digunakan.
- 2) **Active record**, jika bernilai **TRUE**, maka fasilitas Active Records CI dapat digunakan.

Active Record membantu membuat query yang akan **support** multi DBMS.

- 3) **Hostname**, isikan alamat IP atau hostname database server.
- 4) **Username**, isikan username yang dipakai untuk login ke database.
- 5) **Password**, isikan password yang dipakai untuk login ke database.
- 6) **Database**, isikan nama database yang dipakai aplikasi.

- 7) **DB Driver**, isikan tipe driver DBMS yang dipakai aplikasi. Driver DMBS yang terkoneksi dengan CI saat ini : mysql, mysqli, postgre, odbc, mssql, sqllite, oci8.
- 8) **P Connect**, isikan nilai TRUE jika ingin koneksi ke database server tetap terhubung
untuk beberapa waktu sejak eksekusi query terakhir (Persistent Connection).
Cukup membantu agar proses aplikasi lebih cepat, dibandingkan harus menjalin koneksi ke database setiap akan eksekusi query.
- 9) **DB Debug**, isikan nilai TRUE jika ingin error koneksi atau error query ditampilkan.
Sangat membantu pada proses development.

Jika aplikasi akan di-*publish* / *deploy* maka nilai [‘db_debug’] sebaiknya **FALSE**

- e. **Routes.php**, pada file ini didefinisikan beberapa URL yang spesifik mengacu ke controller yang spesifik pula sesuai kebutuhan aplikasi. Pada file ini juga dapat didefinisikan default controller yang akan dipanggil jika tidak didefinisikan pada URL.

Itulah mengapa tanpa konfigurasi awal halaman pembuka CI bisa tampil seperti di Gambar B.2.2, karena \$route[‘default_controller’] diisi controller “welcome”.

Silahkan buka file welcome.php di folder controllers untuk lebih jelasnya.

5. Akses User guide

Buku pedoman (user guide) CodeIgniter dapat diakses pada alamat:

- a. http://www.codeigniter.com/user_guide
- b. Direktori kerja Anda, misalnya http://localhost/123040024/user_guide

Untuk penjelasan lebih detail mengenai seluk-beluk CodeIgniter silahkan akses User Guidenya yang rapi dan mudah dipahami.

LATIHAN:

Diketahui atribut data buku sebagai berikut:

Field	Tipe Data	Ukuran
ID_Buku	Integer	8, auto increment
Judul	Varchar	100
Pengarang	Varchar	150
Kategori	Varchar	10

Pada modul ini, kita akan membuat form untuk input data buku sesuai atribut di atas.
Form input data yang akan kita buat kurang lebih seperti ini :

Judul :

Pengarang :

Kategori :

Selanjutnya ikuti langkah-langkah berikut :

1. Buat controller Buku.

Buatlah sebuah file menggunakan editor PHP, ketikkan script di bawah ini:

```

1 <?php
2
3 class Buku extends CI_Controller
4 {
5     var $data = array();
6
7     function __construct()
8     {
9         parent::__construct();
10
11         $this->load->helper('form');
12
13         $this->data['opt_kategori'] = array('' => '- Pilih salah satu -',
14                                         'novel' => 'Novel',
15                                         'komik' => 'Komik',
16                                         'kamus' => 'Kamus');
17     }
18
19     function index()
20     {
21         $this->add_new();
22     }
23
24     function add_new()
25     {
26         $this->data['is_update'] = 0;
27
28         $this->load->view('buku_form_v', $this->data);
29     }
30 }
31
32 ?>
```

Simpan file tersebut di direktori kerja Anda, pada folder: application/controllers/
Beri nama file : **buku.php**

Penjelasan:

```
class Buku extends CI_Controller
```

- Nama class harus diawali huruf kapital.
- Semua controller yang kita buat harus turunan dari controller CI (**CI_Controller**)
- Lebih detail mengenai controller : [user_guide/general/controllers.html](#)

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

- Membuat konstruktor class Buku yang merupakan overriding dari konstruktor class **CI_Controller**.
- Maksud dibuat konstruktor agar saat class dipanggil, kita dapat menyisipkan beberapa script yang otomatis dieksekusi.

```
$this->load->helper('form');
```

- Memanggil helper form, agar dapat digunakan di view.
- Lebih detail mengenai helper form : [user_guide/helpers/form_helper.html](#)

```
$this->data['opt_kategori'] = array('' => '- Pilih salah satu -',
                                    'novel' => 'Novel',
                                    'komik' => 'Komik',
                                    'kamus' => 'Kamus');
```

- Untuk mengirimkan variabel ke view, harus variabel bertipe array.
- **opt_kategori** dimaksudkan untuk nilai option pada dropdown di view. Lihat point C.2 berikutnya untuk lebih jelas.

```
function index()
{
    $this->add_new();
}
```

- Jika pada class ada method bernama index dan pada URL tidak didefinisikan method apa yang dipanggil, maka method index yang otomatis dipanggil oleh CI.

```
function add_new()
{
    $this->data['is_update'] = 0;
```

- Method add_new untuk memanggil form input data buku.
- **\$this->data['is_update'] = 0;** Dimaksudkan untuk mengirim variabel **is_update** yang bernilai **0** ke view. Berarti aksi yang akan dilakukan bukan update data.

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

- Memanggil view bernama buku_form_v , yang merupakan form input data buku.
- Parameter ke-2, berarti mengirimkan variabel `$this->data` ke view.
- Karena variabel `$this->data` bertipe array, maka index dari array tersebut akan menjadi nama variable di view.

2. Buat view Form Input Data

Buat file baru, kemudian ketikkan script di bawah ini:

```
1 <?php
2
3     echo form_open('buku/save/'.$is_update);
4
5     echo form_hidden('id','');
6
7     echo "Judul : ".form_input('judul','','size='50' maxlength='100');
8     echo "<br /><br />";
9
10    echo "Pengarang : ".form_input('pengarang','','size='50' maxlength='150');
11    echo "<br /><br />";
12
13    echo "Kategori : ".form_dropdown('kategori',$opt_kategori,'');
14    echo "<br /><br />";
15
16    echo form_submit('btn_simpan','Simpan');
17
18    echo form_close();
19
20 ?>
```

Simpan file tersebut di direktori kerja Anda, pada folder: application/views/
Beri nama file : **buku_form_v.php**

Penjelasan:

```
echo form_open('buku/save/'.$is_update);
```

- Membuat elemen form.
- Script tersebut sama dengan:

```
<form action="buku/save/0" method="POST">
```

```
    echo form_hidden('id','');

```

- Membuat elemen hidden bernama id. • Parameter pertama adalah nama elemen, parameter kedua adalah default value.
- Script tersebut sama dengan:

```
    <input type="hidden" name="id" value="" />
```

```
    echo "Judul : ".form_input('judul','','size='50' maxlength='100');
```

- Membuat elemen input textbox bernama judul.

- Parameter pertama adalah nama elemen, kedua adalah default value, ketiga adalah atribut tambahan.
- Script tersebut sama dengan:

```
Judul : <input type="text" name="judul" size="50" maxlength="100" value="" />
echo "Kategori : ".form_dropdown('kategori',$opt_kategori,'');
```

- Membuat elemen combobox / dropdown list.
- Parameter pertama adalah nama elemen, kedua adalah opsi yang tersedia untuk combobox tersebut, ketiga adalah default value, keempat adalah atribut tambahan.
- Script tersebut sama dengan:

```
Kategori : <select name="kategori">
    <option selected="selected">- Pilih salah satu -</option>
    <option value="novel">Novel</option>
    <option value="komik">Komik</option>
    <option value="kamus">Kamus</option>
</select>

echo form_submit('btn_simpan','Simpan');
```

- Membuat button submit bernama btn_simpan dan berteks Simpan.
- Parameter pertama adalah nama elemen, kedua adalah teks button, ketiga adalah atribut tambahan.
- Script tersebut sama dengan:

```
<input type="submit" name="btn_submit" value="Simpan" />

echo form_close();
```

- Membuat tag tutup form.

3. Akses Form Input Data buku pada browser

Akses form input data buku pada browser dengan URL sesuai direktori kerja Anda.
Contoh :

<http://localhost/123040024/index.php/buku/>

atau

http://localhost/123040024/index.php/buku/add_new

4. Rubah konfigurasi Routes

Agar form input data buku otomatis tampil jika pada URL tidak didefinisikan controller apa yang diakses, maka ubah konfigurasi di file routes.php. Lihat point B.4.e. mengenai konfigurasi routes. Ganti default controller ke controller Buku, sehingga seperti ini :

```
$route['default_controller'] = "buku/add_new";
```

Kemudian akses dengan alamat : <http://localhost/123040024/>

CODEIGNITER DAN DATABASE

Tujuan Praktikum

1. Memahami cara menggunakan codeigniter dengan database
2. Mampu mengimplementasikannya codeigniter dengan memanfaatkan data dari database.

Manipulasi Data pada Database

Suatu aplikasi berbasis web, umumnya menggunakan database sebagai media penyimpanan data. CodeIgniter menyediakan banyak fungsi yang mendukung manipulasi data

pada database, baik menggunakan metode tradisional maupun pola Active Record. Active record adalah suatu metode untuk mengakses data pada database relasional, suatu tabel dan view dipetakan menjadi suatu class. Metode ini biasanya digunakan oleh tools ORM (Object Relational Mapping). Beberapa framework PHP sudah menyertakan modul ORM

yang mengimplementasikan pola Active Record yang mendukung relationship dan validasi, contohnya Symfony, CakePHP dan Yii.

Pada CodeIgniter terdapat fitur untuk membangkitkan Query yang bernama “ActiveRecord”, tapi CI tidak menerapkan pola Active Record yang sebenarnya, hanya sebagai

pembangkit query saja agar aplikasi yang dibangun dapat mendukung multi database. Untuk mengimplementasikan fungsi Active Record yang sebenarnya pada CI dapat menggunakan library tersendiri seperti CodeIgniter DataMapper atau CodeIgniter Gas ORM.

1. CodeIgniter Database Class

Agar aplikasi yang dibangun dapat terkoneksi dengan database, harus tentukan dulu parameter koneksi database pada file konfigurasi yang sudah dibahas pada Modul 1. Setelah semua parameter koneksi ditentukan, gunakan script di bawah ini untuk menggunakan Database Class CodeIgniter :

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

Setelah Database Class dipanggil menggunakan script di atas, maka kita dapat mengeksekusi query pada database tersebut.

Jika setiap halaman aplikasi yang dibangun membutukan koneksi ke database, sebaiknya

Database Class dibuat autoload saja dengan menambahkan library “database” pada konfigurasi autoload*. Seperti contoh di bawah ini:

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

2. Eksekusi Query

Terdapat dua cara untuk mengeksekusi query, yaitu dengan metode “tradisional” dan dengan fitur Active Record.

a. Tradisional

Sintaks query dieksekusi oleh method `query()` pada class Database, intansiasi class Database tersebut menjadi object `db`. Berikut contohnya:

```
$query = $this->db->query('SELECT name, title, email FROM my_table');
```

Sintaks query lengkap select data di atas dimasukkan ke method `query()`. Hasil eksekusinya dapat diolah berupa object atau array, sesuai yang kita tentukan

b. Active Record

Dengan menggunakan Active Record, developer dapat memanipulasi data dengan sederhana dan script yang seminimal mungkin. CodeIgniter men-generate sintaks query berdasar method Active Record yang developer panggil, dan juga query yang di-generate dipastikan didukung oleh DBMS yang digunakan. Contohnya query select yang menggunakan limit antara MySQL dan Oracle berbeda, Active Record otomatis menyesuaikan. Berikut contohnya:

```
$query = $this->db->get('table_name');
```

Dengan menggunakan method get() Active Record, akan menghasilkan sintaks query select terhadap tabel yang ditentukan pada parameter method get() tersebut. Berikut beberapa method Active Record yang biasanya dipakai:

- [Select\(\)](#)
- [From\(\)](#)
- [Where\(\)](#)
- [Get_where\(\)](#)
- [Like\(\)](#)
- [Limit\(\)](#)
- [Order_by\(\)](#)
- [Insert\(\)](#)
- [Update\(\)](#)
- [Delete\(\)](#)

Lebih jelas mengenai cara penggunaan method-method Active Record dapat dilihat pada [user_guide](#) menu Database Class.

Hasil eksekusi query (tradisional maupun Active Record) yang pada contoh disimpan pada variabel `$query`, selalu bertipe object.

3. Mengambil Hasil Query

Terdapat dua tipe query result set yang dapat diolah dari hasil eksekusi query, yaitu bertipe object dan bertipe array.

a. Result set tipe Object

Field-field hasil query select menjadi properti pada suatu object. Kita dapat menggunakan method `result()` pada variabel penampung hasil query, pada contoh sebelumnya `$query`. Setiap method `result()` dipanggil maka akan mengembalikan 1 baris record data bertipe object, dengan properti-properti yang merupakan nama field hasil query. Berikut contohnya:

```

$query = $this->db->get('tabel_buku');

foreach($query->result() as $row)
{
    echo $row->Judul;
    echo $row->Pengarang;
}

```

Foreach digunakan untuk menyusuri hasil query, mulai dari record pertama sampai terakhir.

b. Result set tipe Array

Field-field hasil query select menjadi index pada suatu variabel array. Kita dapat menggunakan method **result_array()** pada variabel penampung hasil query, pada contoh sebelumnya **\$query**. Setiap method **result_array()** dipanggil maka akan mengembalikan 1 baris record data bertipe array, dengan index-index yang merupakan nama field hasil query. Berikut contohnya:

```

$query = $this->db->get('tabel_buku');

foreach($query->result_array() as $row)
{
    echo $row['Judul'];
    echo $row['Pengarang'];
}

```

Foreach digunakan untuk menyusuri hasil query, mulai dari record pertama sampai terakhir.

Developer bebas menentukan tipe result set sesuai kebutuhan. Tipe array biasanya cocok jika nama field dinamis tergantung kondisi tertentu yang juga dinamis, karena lebih mudah memanipulasi nama index array dibanding properti suatu objek.

LATIHAN

Latihan di Modul ini akan melanjutkan studi kasus pada modul sebelumnya, yaitu pengelolaan

data Buku. Diketahui atribut data buku sebagai berikut:

Field	Tipe Data	Ukuran
ID_Buku	Integer	8, auto increment
Judul	Varchar	100
Pengarang	Varchar	150
Kategori	Varchar	10

Selanjutnya ikuti langkah-langkah berikut:

1. Buat Database

Bukalah phpmyadmin melalui browser, <http://localhost/phpmyadmin>. Kemudian buatlah database baru dengan nama buku_NIM. Contoh: buku_123040024.

2. Struktur Tabel Buku

	Field	Type	Collation	Attributes	Null	Default	Extra
	ID_Buku	int(20)			No	None	auto_increment
	Judul	varchar(255)	latin1_general_ci		No	None	
	Pengarang	varchar(255)	latin1_general_ci		No	None	
	Kategori	varchar(30)	latin1_general_ci		No	None	

3. Konfigurasi Koneksi Database

Masuklah ke direktori [/application/config](#), lalu buka file [database.php](#). Tentukan hostname, username, password, dan nama database. Contoh:

```
50 $db['default']['hostname'] = 'localhost';
51 $db['default']['username'] = 'root';
52 $db['default']['password'] = '';
53 $db['default']['database'] = 'buku_123040024';
54 $db['default']['dbdriver'] = 'mysql';
```

Pada kuliah ini username yang digunakan untuk koneksi ke DBMS MySQL adalah root. Jika pada pembangunan aplikasi yang sebenarnya, **sangat disarankan** tidak memakai user root untuk koneksi suatu aplikasi. Buatlah user spesifik yang memiliki hak akses terhadap database yang spesifik pula

4. Konfigurasi Autoload

Masuklah ke direktori [/application/config](#), lalu buka file [autoload.php](#). Tambahkan library database pada konfigurasi autoload. Seperti contoh berikut:

```
55 $autoload['libraries'] = array('database');
```

5. Buat model Buku

Buatlah sebuah file menggunakan editor PHP, ketikkan script model Buku di bawah ini:

Simpan file tersebut di direktori kerja Anda, pada folder : application/models/
Beri nama file : [buku_m.php](#).

```

1 <?php
2
3 class Buku_m extends CI_Model
4 {
5     function __construct()
6     {
7         parent::__construct();
8     }
9
10    function get_records($criteria='', $order='', $limit='', $offset=0)
11    {
12        $this->db->select('*');
13
14        $this->db->from('mst_buku');
15
16        if ($criteria != '')
17            $this->db->where($criteria);
18
19        if ($order != '')
20            $this->db->order_by($order);
21
22        if ($limit != '')
23            $this->db->limit($limit, $offset);
24
25        $query = $this->db->get();
26
27        return $query;
28    }
29
30    function insert($data)
31    {
32        $query = $this->db->insert('mst_buku', $data);
33
34        return $query;
35    }
36
37    function update_by_id($data, $id)
38    {
39        $this->db->where("ID_Buku = '$id'");
40
41        $query = $this->db->update('mst_buku', $data);
42
43        return $query;
44    }

```

```

45
46    function delete_by_id($id)
47    {
48        $query = $this->db->delete('mst_buku', "ID_Buku = '$id'");
49
50        return $query;
51    }
52 }
53
54 ?>

```

Penjelasan:

```
class Buku_m extends CI_Model
```

- Nama class harus diawali huruf kapital.
- Semua model yang dibuat harus turunan dari model CI (**CI_Model**)
- Karena tidak ada aturan penamaan model pada CodeIgniter, disarankan beri ciri-ciri di nama class yang menandakan bahwa class tersebut adalah model, untuk memudahkan proses debugging. Contoh di atas **_m**.

- Lebih detail mengenai model : [user_guide/general/models.html](#)

```
function get_records($criteria='', $order='', $limit='', $offset=0)
{
```

- Method yang akan berfungsi melakukan query select data.
- Terdapat 4 parameter optional sebagai input bagi method, yaitu
 - \$criteria**. Berfungsi sebagai kriteria di query, “ where 1=1 and ”
 - \$order**. Berfungsi sebagai ordering di query, “ Judul ASC, Pengarang DESC ... ”
 - \$limit**. Berfungsi sebagai angka limit (batas record) di query.
 - \$offset**. Berfungsi sebagai angka offset (record awal) di query.

```
$this->db->select('*');
$this->db->from('mst_buku');
```

- Active Record untuk mendefinisikan query select data.
- Script tersebut akan men-generate query:

```
select * from mst_buku
```

```
if ($criteria != '')
    $this->db->where($criteria);
```

- Jika kriteria query didefinisikan, maka gunakan Active Record untuk mendefinisikan kriteria query-nya untuk digenerate.
- Misal variabel **\$criteria** berisi “**Judul like ‘Laskar Pelangi%’**”, maka akan di-generate:

```
select * from mst_buku where 'Judul' like 'Laskar Pelangi%'
```

```
if ($order != '')
    $this->db->order_by($order);
```

- Jika ordering ditentukan, maka gunakan Active Record **order_by()**.
- Misal variabel **\$order** berisi “**Judul**” tanpa didefinisikan arah pengurutannya (ASC atau DESC), maka Active Record akan otomatis menambahkan ASC. Script yang di-generate:

```
select * from mst_buku where 'Judul' like 'Laskar Pelangi%'
order by 'Judul' ASC
```

```
$query = $this->db->get();
```

- Query yang sudah didefinisikan pada Active Record, dieksekusi dengan method **get()**
- Hasil eksekusi query disimpan ke variabel **\$query**, untuk diolah kemudian.

6.Modifikasi controller Buku

Bukalah file controller Buku yang sudah Anda buat minggu lalu, kemudian lakukan modifikasi berikut ini:

- Load helper URL dan model Buku_m

Pada constructor, tambahkan script untuk load helper URL dan model Buku_m. Sehingga seperti berikut:

```
-----  
function __construct()  
{  
    parent::__construct();  
  
    $this->load->helper('form');  
    $this->load->helper('url');  
  
    $this->data['opt_kategori'] = array('' => '- Pilih salah satu -',  
                                         'novel' => 'Novel',  
                                         'komik' => 'Komik',  
                                         'kamus' => 'Kamus');  
  
    $this->load->model('Buku_m');  
}
```

Penjelasan:

- Helper URL digunakan di beberapa bagian pada controller **Buku**, contohnya function **redirect()**.
- Model **Buku_m** yang sudah dibuat harus dipanggil (instansiasi) agar dapat digunakan pada controller. Instansiasi model Buku_m, akan menjadi properti di class controller Buku yang namanya sama dengan nama model, yaitu **Buku_m**.

- Modifikasi method index()

Pada Modul 1 method **index** berfungsi sebagai function default yang memanggil method **add_new()** jika pada URL tidak didefinisikan method yang dipanggil. Pada Modul 2 ini modifikasi method **index** yang dimaksudkan berfungsi mengambil data buku dari database kemudian menampilkan di view. Berikut ini script modifikasi method **index()**:

```
function index()  
{  
    $this->data['query'] = $this->Buku_m->get_records();  
  
    $this->load->view('buku_v', $this->data);  
}
```

Penjelasan:

```
$this->data['query'] = $this->Buku_m->get_records();
```

- Panggil method `get_records()` pada model `Buku_m`, untuk mengambil data buku di tabel.
- Return value dari method `get_records()` yang berupa hasil eksekusi query dimasukkan ke properti `data['query']` untuk dikirimkan ke view.

c. Buat method save()

Pada Modul 1 kita telah membuat form input data buku, dimana action form tersebut mengacu ke class Buku method save. Buatlah sebuah method baru bernama `save()` seperti di bawah ini:

```
function save($is_update=0)
{
    $data['Judul']      = $this->input->post('judul', true);
    $data['Pengarang']  = $this->input->post('pengarang', true);
    $data['Kategori']   = $this->input->post('kategori', true);

    if ($is_update==0)
    {
        // Jika tambah data baru
        if ($this->Buku_m->insert($data))
            redirect('buku');
    }
    else
    {
        // Jika update data
        $id = $this->input->post('id');

        if ($this->Buku_m->update_by_id($data, $id))
            redirect('buku');
    }
}
```

Penjelasan:

```
function save($is_update=0)
{
```

- Method yang akan berfungsi melakukan aksi insert atau update data, tergantung dari parameter `$is_update` yang diterimanya dari view (action dari form).
`$is_update == 0`, maka aksi insert.
`$is_update != 0`, maka aksi update.

```
$data['Judul']      = $this->input->post('judul', true);
```

- Mengambil data dari form yang dikirimkan menggunakan method post, dimasukkan ke variabel `$data['Judul']`. Index dari variabel `$data` tersebut merupakan nama field di tabel mst_buku, yang akan disimpan datanya.
- Parameter pertama `$this->input->post()` adalah nama elemen di form, parameter kedua bernilai `TRUE` mendefinisikan bahwa data harus difilter dahulu oleh class Security.
- Script tersebut sama dengan `$_POST['judul']`, namun cara konvensional seperti itu tanpa filter security dahulu.

```
if ($is_update==0)
{
    // Jika tambah data baru
    if ($this->Buku_m->insert($data))
        redirect('buku');
```

- Jika `$is_update` bernilai 0, maka tambah data baru (insert).
- Variabel `$data` yang merupakan penampung data dari form, dikirimkan sebagai parameter method `insert()` di model `Buku_m`.
- Jika proses insert data sukses, maka redirect browser ke controller `Buku`.

```
else
{
    // Jika update data
    $id = $this->input->post('id');

    if ($this->Buku_m->update_by_id($data, $id))
        redirect('buku');
```

- Jika `$is_update` tidak bernilai 0, maka update data lama.
- Ambil ID_Buku dari form, simpan ke variabel `$id`.
- Variabel `$data` yang merupakan penampung data dari form, dikirimkan sebagai parameter method `update_by_id()` di model `Buku_m`.
- Jika proses update data sukses, maka redirect browser ke controller `Buku`.

7. Buat view Daftar Buku

Buatlah file baru yang akan berfungsi sebagai view daftar buku, antarmukanya kurang lebih seperti ini :

Tambah Buku					
No	Judul	Pengarang	Kategori	Aksi	
1	Modul Kuliah Rekasaya Web - CodeIgniter Framework	Candra Utama	Novel	Edit Hapus	

Scriptnya seperti di bawah ini:

```
<?php

    echo anchor('buku/add_new', 'Tambah Buku');
    echo "<br /><br />";

    echo "<table border='1'>
        <tr>
            <th>No</th>
            <th>Judul</th>
            <th>Pengarang</th>
            <th>Kategori</th>
            <th>Aksi</th>
        </tr>";

    $no = 0;
    foreach($query->result_array() as $row)
    {
        $no++;
        $kategori = $row['Kategori'];

        $link_edit = anchor('buku/edit/'.$row['ID_Buku'], 'Edit');
        $link_delete = anchor('buku/delete/'.$row['ID_Buku'], 'Hapus');

        echo "<tr>
            <td>".$no."</td>
            <td>".$row['Judul']."</td>
            <td>".$row['Pengarang']."</td>
            <td>".$opt_kategori[$kategori]."</td>
            <td>".$link_edit.' '.$link_delete."</td>
        </tr>";
    }
    echo "</table>";

?>
```

Simpan file tersebut di direktori kerja Anda, pada folder : application/views/
Beri nama file : **buku_v.php**

Penjelasan:

```
echo anchor('buku/add_new', 'Tambah Buku');
```

- Anchor() merupakan fungsi dari helper URL untuk men-generate hiperlink.
- Parameter pertama adalah alamat (url) yang akan diakses.
- Parameter kedua adalah teks dari hiperlink tersebut.
- Script tersebut sama dengan:

```
<a href='http://localhost/123040024/index.php/buku/add_new'>Tambah Buku</a>
```

```
foreach($query->result_array() as $row)
{
```

- Controller Buku mengirimkan hasil query melalui variabel `data['query']`. Oleh CI, index variabel data tersebut diubah menjadi variabel pada view. (Ingin kembali materi di Modul 1).
- Ulangi sebanyak record hasil query, resultset query di-fetch ke variabel `$row` dengan tipe result array. Lihat point A.3.b.
- Script tersebut sama dengan:

```
while ($row = mysql_fetch_array($query))
{
}
```

8. Uji Coba Insert Data

Lakukan penambahan beberapa data buku. Jika sudah sukses lanjutkan langkah 9.

9. Buat function Edit

Buka kembali file controller `Buku`, kemudian buat method baru bernama `edit`, yang akan berfungsi sebagai pengendali aksi edit data. Berikut script method `edit`:

```
function edit($id)
{
    $this->data['query'] = $this->Buku_m->get_records("ID_Buku = '$id'");
    $this->data['is_update'] = 1;
    $this->load->view('buku_form_v', $this->data);
}
```

Penjelasan:

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

- Method yang akan berfungsi mengambil data buku berdasarkan parameter `$id` yang diterima dari view
- Method ini dipanggil melalui hyperlink Edit pada view daftar buku.

```
$this->data['query'] = $this->Buku_m->get_records("ID_Buku = '$id');
```

- Panggil method `get_records()` pada model `Buku_m`, untuk mengambil data buku di tabel, berdasarkan kriteria `ID_Buku = '$id'`.

```
$this->data['is_update'] = 1;
```

- Dimaksudkan untuk mengirim variabel `is_update` yang bernilai `1` ke view. Berarti aksi yang akan dilakukan adalah update data.

10.Modifikasi view Form Input Data

Untuk fitur edit data, tentunya harus ditampilkan dulu data sebelumnya yang akan diedit. Method `edit` pada controller `Buku` memberikan data yang akan diedit tersebut ke view, maka tugas view adalah menampilkannya pada kolom-kolom isian data. Buka kembali file view `buku_form_v.php`, kemudian modifikasi beberapa baris seperti berikut:

```
<?php  
  
if (!empty($query))  
{  
    $row = $query->row_array();  
}  
else  
{  
    $row['ID_Buku'] = '';  
    $row['Judul'] = '';  
    $row['Pengarang'] = '';  
    $row['Kategori'] = '';  
}  
  
echo form_open('buku/save/'. $is_update);  
  
echo form_hidden('id', $row['ID_Buku']);  
  
echo "Judul : ".form_input('judul', $row['Judul'], "size='50' maxlength='100'");  
echo "<br /><br />";  
  
echo "Pengarang : ".form_input('pengarang', $row['Pengarang'], "size='50' maxlength='150'");  
echo "<br /><br />";  
  
echo "Kategori : ".form_dropdown('kategori', $opt_kategori, $row['Kategori']);  
echo "<br /><br />";  
  
echo form_submit('btn_simpan', 'Simpan');  
  
echo form_close();  
?  

```

Penjelasan:

```
if (!empty($query))  
{  
    $row = $query->row_array();  
}
```

- Jika variabel `$query` yang dikirimkan dari controller tidak kosong, maka ambil record pertama hasil eksekusi query di variabel `$query`.
- `row_array()` mengembalikan result satu record.

```
echo form_hidden('id', $row['ID_Buku']);
```

- Membuat elemen hidden bernama id, value diisi ID_Buku dari tabel mst_buku.
- Script tersebut sama dengan:

```
<input type="hidden" name="id" value="php echo $row['ID_Buku']?" />
```

```
echo "Kategori : ".form_dropdown('kategori',$opt_kategori,$row['Kategori']);
```

- Membuat elemen combobox / dropdown list bernama kategori.
- Parameter pertama adalah nama elemen, kedua adalah opsi yang tersedia untuk combobox tersebut, ketiga adalah default value, keempat adalah atribut tambahan.
- Default value diambil dari data pada tabel mst_buku.
- Script tersebut kurang lebih sama dengan:

```
$selected_kosong = "";
if ($row['Kategori'] == '')
    $selected_kosong = "selected = 'selected'";

$selected_novel = "";
if ($row['Kategori'] == 'novel')
    $selected_novel = "selected = 'selected'";

$selected_komik = "";
if ($row['Kategori'] == 'komik')
    $selected_komik = "selected = 'selected'";

$selected_kamus = "";
if ($row['Kategori'] == 'kamus')
    $selected_kamus = "selected = 'selected'";

<select name="kategori">
    <option value="" $selected_kosong>- Pilih salah satu -</option>
    <option value="novel" $selected_novel> Novel </option>
    <option value="komik" $selected_komik> Komik </option>
    <option value="kamus" $selected_kamus> Kamus </option>
</select>
```

11.Uji coba edit

Lakukan perubahan beberapa data buku. Jika sudah sukses lanjutkan langkah 12.

12.Buat function delete

Buka kembali file controller **Buku**, kemudian buat method baru bernama **delete**, yang akan berfungsi sebagai pendelasi aksi hapus data. Berikut script method **delete**:

```
function delete($id)
{
    if($this->Buku_m->delete_by_id($id))
    {
        redirect('buku');
    }
}
```

Penjelasan:

```
function delete($id)
{
```

- Method yang akan berfungsi menghapus data buku berdasarkan parameter `$id` yang diterima dari view.
- Method ini dipanggil melalui hyperlink Hapus pada view daftar buku.

```
if($this->Buku_m->delete_by_id($id))
{
    redirect('buku');
}
```

- Hapus data buku berdasarkan ID_Buku yang dikirimkan dari view.
- Jika hapus sukses, maka redirect browser ke controller Buku.

13.Uji coba delete

Lakukan hapus beberapa data buku. Sebelum aksi hapus data benar-benar dilakukan program, sangat disarankan ada konfirmasi terlebih dahulu. Bagaimana caranya...? Tambahkan javascript berikut pada view `buku_v.php` pada hyperlink Hapus:

```
$link_delete = anchor('buku/delete/'.$row['ID_Buku'], 'Hapus', "onclick='return confirm(\"Yakin?\\\")'");
```

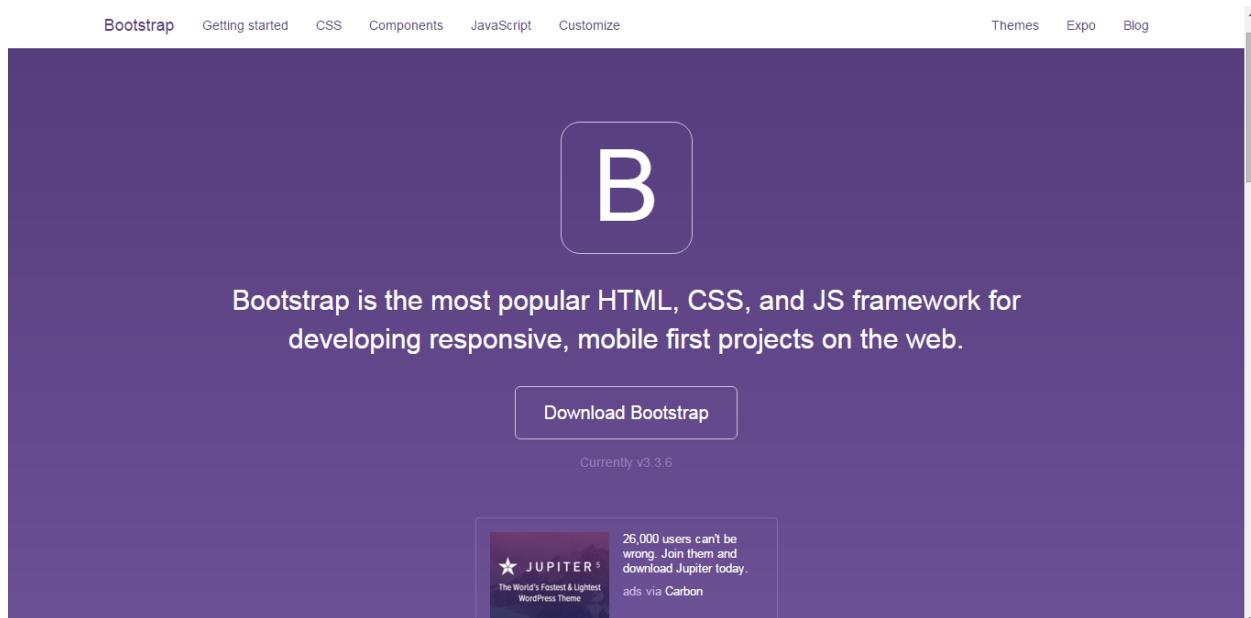
TEMPLATE BOOTSTRAP

Tujuan Praktikum

1. Memahami beberapa class yang ada pada template bootstrap
2. Mampu mengimplementasikannya template bootstrap dalam halaman HTML

Pengenalan Bootstrap

Untuk cara menggunakan bootstrap yang pertama kali yang harus anda lakukan adalah melakukan instalasi bootstrap. anda dapat mendownload langsung bootstrap pada website resminya di <http://getbootstrap.com/getting-started/>.



Berikut ini adalah struktur isi dari bootstrap setelah anda mendownloadnya.

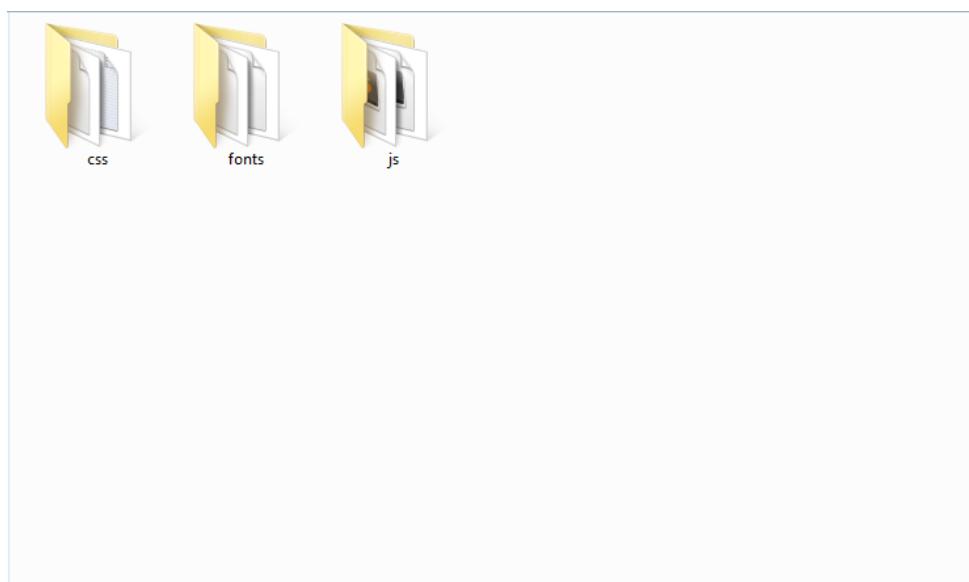
```
1  bootstrap/
2    --css/
3      --bootstrap.css
4      --bootstrap.css.map
5      --bootstrap.min.css
6      --bootstrap.min.css.map
7      --bootstrap-theme.css
8      --bootstrap-theme.css.map
9      --bootstrap-theme.min.css
10     --bootstrap-theme.min.css.map
11     --js/
12       --bootstrap.js
13       --bootstrap.min.js
```

```
14 └── fonts/
15     ├── glyphicons-halflings-regular.eot
16     ├── glyphicons-halflings-regular.svg
17     ├── glyphicons-halflings-regular.ttf
18     ├── glyphicons-halflings-regular.woff
19     └── glyphicons-halflings-regular.woff2
```

sedikit informasi tambahan sebelum anda mendownload **bootstrap** anda dapat menentukan sendiri atau meng-costumize bootstrap sesuai dengan keinginan anda.

Cara Menginstal Bootstrap

Cara menginstall bootstrap ada beberapa cara. anda bisa menginstall bootstrap dengan composer, bower, menggunakan npm, secara offline dan online. berikut ini contoh perintah untuk menginstall bootstrap menggunakan bower. Cara menginstall bootstrap secara offline ini sangat banyak di gunakan oleh pada developer, karena anda dapat bekerja menggunakan bootstrap tanpa harus terkoneksi dengan internet lagi, karena file framework bootstrap sudah terletak pada pc kita sendiri. lain dengan penginstalan bootstrap secara online yang file-file bootstrap nya terletak pada server luar. Setelah mendownload framwork css bootstrap kemudian ekstrak pada project anda. di sini saya meletakkannya pada localhost saya dengan alamat <http://localhost/IF/bootstrap/>



cara menginstall bootstrap

kemudian untuk menghubungkan atau menginstall bootstrap buat sebuah file HTML atau PHP yang akan kita hubungkan dengan bootstrap. di sini saya membuat sebuah file dengan nama index.php.

Bootstrap juga memerlukan jquery karena untuk bagian javascript nya bootstrap menggunakan jquery, jadi jangan lupa untuk menambahkan juga file jquery pada folder js bootstrap atau letakkan sesuai keinginan anda asalkan direktori saat menghubungkan file css dan js nanti sesuai.

Latihan: index.php

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Bootstrap Part 1 : Pengertian dan Cara Menggunakan Bootstrap</title>
5     <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
6     <script type="text/javascript" src="js/jquery.js"></script>
7     <script type="text/javascript" src="js/bootstrap.js"></script>
8 </head>
9 <body>
10    <h1>Cara Menggunakan Bootstrap </h1>
11    <button class="btn btn-danger">TOMBOL MERAH</button>
12    <button class="btn btn-primary">TOMBOL BIRU</button>
13 </body>
14 </html>
```

dapat di perhatikan pada contoh cara menghubungkan file **bootstrap** di atas. untuk menghubungkan file html atau php anda dengan bootstrap.css

```
1 <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
```

kemudian hubungkan file html atau php anda dengan bootstrap.js dan jquery.js. ingat letakkan jquery.js pada bagian atas karena bootstrap bekerja menggunakan bantuan jquery bukan jquery menggunakan bantuan bootstrap.

```
1 <script type="text/javascript" src="js/jquery.js"></script>
2 <script type="text/javascript" src="js/bootstrap.js"></script>
```

kemudian class “btn” pada element <button> di atas merupakan **class bootstrap** yang bisa di gunakan untuk membuat tombol. dan “btn-danger” dan “btn-primary” untuk menentukan warna dari tombol yang ingin kita buat.

dan jika di jalankan pada browser maka hasilnya

Cara Menggunakan Bootstrap

TOMBOL MERAH TOMBOL BIRU

contoh penggunaan bootstrap

NOTE : Perhatikan letak direktori file css dan js bootstrap dan jquery pada saat anda menghubungkannya. anda juga tidak perlu menghubungkan seluruh isi bootstrap, yang hanya perlu di hubungkan adalah file bootstrap.css dan bootstrap.js.

Membuat Table Dengan Bootstrap

Untuk membuat table dengan bootstrap caranya sangat mudah. anda hanya perlu memasukkan class-class bootstrap yang di gunakan untuk mendesign table. Adapun beberapa class bootstrap yang di gunakan untuk design atau membuat table adalah sebagai berikut.

- .table
class ini di gunakan untuk mendefinisikan atau membuat table standar/ table biasa.
- .table-striped
table-striped merupakan class tambahan untuk membuat table dengan menggunakan bootstrap. class ini berfungsi untuk membuat baris table yang bergaya belang-beling (strip).
- .table-bordered
table-bordered di gunakan untuk membuat table yang memiliki garis dan termasuk class tambahan pada bootstrap untuk mendesign table.
- .table-hover
table-hover merupakan class tambahan untuk mendesign table menggunakan bootstrap.

table hover di gunakan untuk membuat efek hover pada row table pada saat cursor mouse di letakkan di atas row table.

dan untuk membuat pewarnaan pada row atau table data anda dapat menggunakan class-class berikut ini yang dapat anda tambahkan pada tag <tr> untuk memberi warna pada table row, atau tambahkan pada tag <td> untuk memberi warna pada table data.

- .success

class ini di gunakan untuk memberi tanda sukses atau warna hijau pada table row atau table data.

- .danger

class ini di gunakan untuk memberi warna merah pada table row atau table data.

- .info

class ini di gunakan untuk memberi warna biru pada table row atau table data.

- .warning

class ini di gunakan untuk memberi warna kuning pada table row atau table data.

Untuk membuat atau mendesign table dengan bootstrap buat sebuah file php atau html. di sini saya membuat file index.php.

Latihan: index.php

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Bootstrap Part 2 : Membuat table dengan Bootstrap</title>
5     <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
6     <script type="text/javascript" src="js/jquery.js"></script>
7     <script type="text/javascript" src="js/bootstrap.js"></script>
8 </head>
9 <body>
10    <h1>Cara Membuat table dengan Bootstrap </h1>
11    <table class="table">
12        <thead>
13            <tr>
14                <th>No</th>
15                <th>Makanan</th>
16                <th>Harga</th>
17            </tr>
18        </thead>
19        <tbody>
20            <tr>
21                <td>1</td>
22                <td>Bakso</td>
23                <td>12.000</td>
24            </tr>
25            <tr>
26                <td>2</td>
27                <td>Mie Goreng</td>
28                <td>7.000</td>
29            </tr>
30            <tr>
31                <td>3</td>
32                <td>Nasi Goreng</td>
33                <td>15.000</td>
34            </tr>
35            <tr>
36                <td>4</td>
37                <td>Sate Padang</td>
38                <td>17.000</td>
39            </tr>
40            <tr>
41                <td>5</td>
42                <td>Nasi Soto</td>
43                <td>20.000</td>
44            </tr>
45        </tbody>
46    </table>
47 </body>
48 </html>

```

HASIL

No	Makanan	Harga
1	Bakso	12.000
2	Mie Goreng	7.000
3	Nasi Goreng	15.000
4	Sate Padang	17.000
5	Nasi Soto	20.000

ANALISIS :

Membuat table border dengan bootstrap

untuk membuat table border atau table yang memiliki garis anda hanya perlu menambahkan class “**table_bordered**” bersamaan dengan class “table”. Contohnya

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Bootstrap Part 2 : Membuat table dengan Bootstrap</title>
5     <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
6     <script type="text/javascript" src="js/jquery.js"></script>
7     <script type="text/javascript" src="js/bootstrap.js"></script>
```

```

8 </head>
9 <body>
10    <h1>Cara Membuat table dengan Bootstrap </h1>
11    <table class="table table-bordered">
12        <thead>
13            <tr>
14                <th>No</th>
15                <th>Makanan</th>
16                <th>Harga</th>
17            </tr>
18        </thead>
19        <tbody>
20            <tr>
21                <td>1</td>
22                <td>Bakso</td>
23                <td>12.000</td>
24            </tr>
25            <tr>
26                <td>2</td>
27                <td>Mie Goreng</td>
28                <td>7.000</td>
29            </tr>
30            <tr>
31                <td>3</td>
32                <td>Nasi Goreng</td>
33                <td>15.000</td>
34            </tr>
35            <tr>
36                <td>4</td>
37                <td>Sate Padang</td>
38                <td>17.000</td>
39            </tr>
40            <tr>
41                <td>5</td>
42                <td>Nasi Soto</td>
43                <td>20.000</td>
44            </tr>
45        </tbody>
46    </table>
47 </body>
48 </html>

```

HASIL

:

The screenshot shows a web browser window with the title "Bootstrap Part 2 : Membuat table dengan Bootstrap". The address bar indicates the URL is "localhost/boot/lat2.php". The main content area displays a table with the following data:

No	Makanan	Harga
1	Bakso	12.000
2	Mie Goreng	7.000
3	Nasi Goreng	15.000
4	Sate Padang	17.000
5	Nasi Soto	20.000

ANALISIS :

Membuat Table strip dengan bootstrap

table dengan gaya strip atau belang-belang dapat di buat dengan menggunakan class “table-striped” pada bootstrap

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Bootstrap Part 2 : Membuat table dengan Bootstrap</title>
5     <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
6     <script type="text/javascript" src="js/jquery.js"></script>
7     <script type="text/javascript" src="js/bootstrap.js"></script>
8 </head>
9 <body>
10    <h1>Cara Membuat table dengan Bootstrap </h1>
11    <table class="table table-bordered table-striped">
12        <thead>
13            <tr>
14                <th>No</th>
15                <th>Makanan</th>
16                <th>Harga</th>
17            </tr>
18        </thead>
19        <tbody>
```

```
20      <tr>
21          <td>1</td>
22          <td>Bakso</td>
23          <td>12.000</td>
24      </tr>
25      <tr>
26          <td>2</td>
27          <td>Mie Goreng</td>
28          <td>7.000</td>
29      </tr>
30      <tr>
31          <td>3</td>
32          <td>Nasi Goreng</td>
33          <td>15.000</td>
34      </tr>
35      <tr>
36          <td>4</td>
37          <td>Sate Padang</td>
38          <td>17.000</td>
39      </tr>
40      <tr>
41          <td>5</td>
42          <td>Nasi Soto</td>
43          <td>20.000</td>
44      </tr>
45  </tbody>
46 </table>
47 </body>
48 </html>
```

HASIL :

No	Makanan	Harga
1	Bakso	12.000
2	Mie Goreng	7.000
3	Nasi Goreng	15.000
4	Sate Padang	17.000
5	Nasi Soto	20.000

ANALISIS :

Membuat table hover dengan bootstrap

Seperti pada **contoh-contoh membuat table dengan bootstrap** di atas, untuk membuat efek hover pada table anda hanya perlu menambahkan class “table-hover” pada tag <table> pembuka.

```
<table class="table table-hover">
```

Membuat table responsive dengan bootstrap

Untuk membuat table yang responsive dengan bootstrap anda hanya tinggal meletakkan element table di dalam element <div> yang di beri class “table-responsive”. <div class='table-responsive'>. kelebihan dari class table-responsive ini akan membuat table anda tidak akan berantakan walaupun di buka melalui smartphone yang memiliki resolusi layar yang lebih kecil. akan muncul scrol secara otomatis untuk membuat menjaga table tetap utuh. berikut ini adalah contoh **membuat table responsive dengan bootstrap**.

```
<table class=" table table-bordered table-striped table-hover ">
```

Tampilan Gambar Dengan Bootstrap

Membuat tampilan gambar yang menarik dengan bootstrap. yaitu membuat gambar berbentuk melengkung pada sisi sudut, gambar dengan bentuk lingkaran, gambar thumbnail dan gambar responsive dengan bootstrap.

Bootstrap menyediakan class untuk **mengatur gambar** sesuai dengan keinginan, misalnya membuat gambar dengan bentuk rounder, **circle**(lingkaran) dan gambar yang responsive. berikut ini di jelaskan tentang beberapa class **bootstrap** yang bisa di gunakan untuk membuat memanipulasi tampilan gambar pada bootstrap

- .img-responsive

class img-responsive berguna untuk membuat gambar menjadi responsive saat di jalankan pada semua bentuk dan ukuran resolusi.

- img-rounded

class img-rounded di gunakan untuk membuat gambar berbentuk round atau pada sisi sudut gambar memiliki bentuk melengkung.

- img-circle

class img-circle di gunakan untuk membuat gambar dengan bentuk lingakaran.

- img-thumbnail

class img-thumbnail di gunakan untuk membuat gambar thumbnail dengan bootstrap.

berikut ini beberapa contoh yang dapat di lihat untuk membuat tampilan gambar dengan bootstrap.

Membuat gambar thumbnail dengan bootstrap

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Bootstrap Part 3 : Tampilan gambar dengan Bootstrap</title>
5     <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
6     <script type="text/javascript" src="js/jquery.js"></script>
7     <script type="text/javascript" src="js/bootstrap.js"></script>
8   </head>
9   <body>
10    <h1>Cara Membuat tampilan gambar dengan Bootstrap</h1>
11    
13 </body>
</html>
```

HASIL :

Cara Membuat tampilan gambar dengan Bootstrap



ANALISIS :

Membuat gambar circle dengan bootstrap

Sama dengan cara membuat gambar **thumbnail** pada contoh di atas. untuk membuat gambar dengan bentuk lingkaran anda dapat menggunakan class “img-circle”.

```
< img src="boot.png" class="img-circle" alt="Gambar" width="304"
height="236">
```

Membuat gambar round dengan bootstrap

```

```

Membuat gambar responsive dengan bootstrap

untuk membuat gambar responsive dengan menggunakan bootstrap anda dapat melakukannya dengan menambahkan class “img-responsive” pada elemeng gambar.

Membuat Tombol Dengan Bootstrap

Selain mampu mendesign tampilan table, menu navigasi dan lain-lainnya, bootstrap juga memiliki kemampuan untuk mendesign tombol atau hyperlink. dan cara membuat tombol dengan bootstrap juga sangat mudah. anda hanya tinggal menambahkan class “btn” untuk mendefinisikan element menjadi tombol, dan kemudian mengatur warna tombol yang anda inginkan, serta mengatur ukuran tombol yang anda inginkan.

Bootstrap menyediakan beberapa class yang bisa kita gunakan untuk mendesign atau menciptakan bentuk tombol yang menarik. berikut ini adalah beberapa class bootstrap yang bisa di gunakan untuk membuat tombol, mengatur ukuran dan mengatur warna dari tombol tersebut.

- .btn

btn merupakan class bootstrap yang bisa di gunakan untuk membuat tombol.

- .btn-xs

btn-xs merupakan class bootstrap untuk membuat tombol dengan ukuran sangat kecil atau extra kecil.

- .btn-sm

btn-sm merupakan class bootstrap untuk membuat tombol dengan ukuran yang kecil.

- .btn-md

btn-md merupakan class bootstrap untuk membuat tombol dengan ukuran sedang.

- .btn-lg

btn-lg merupakan class bootstrap untuk membuat tombol dengan ukuran besar.

- .btn-danger
btn-danger merupakan class bootstrap untuk membuat tombol berwarna merah.
- .btn-default
btn-default merupakan class bootstrap untuk membuat tombol dengan warna standar bootstrap.
- .btn-warning
btn-warning merupakan class bootstrap untuk membuat tombol berwarna kuning.
- .btn-primary
btn-primary merupakan class bootstrap untuk membuat tombol berwarna biru.
- .btn-info
btn-danger merupakan class bootstrap untuk membuat tombol berwarna biru langit.
- .btn-success
btn-success merupakan class bootstrap untuk membuat tombol berwarna hijau.

Untuk membuat tombol dengan bootstrap anda bisa menggunakan tag <button> atau bisa juga menggunakan tag hyperlink yaitu <a>. Silahkan perhatikan contoh membuat tombol dengan bootstrap di bawah ini. yang menggunakan class-class tombol bootstrap yang telah di jelaskan di atas.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Bootstrap Part 4 : Membuat tombol dengan Bootstrap</title>
5     <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
6     <script type="text/javascript" src="js/jquery.js"></script>
7     <script type="text/javascript" src="js/bootstrap.js"></script>
8 </head>
9 <body>
10    <h1>Cara Membuat tombol dengan Bootstrap </h1>
11    <a href="#" class="btn btn-success btn-lg">Tombol</a>
12    <a href="#" class="btn btn-success btn-md">Tombol</a>
13    <a href="#" class="btn btn-success btn-sm">Tombol</a>
14    <a href="#" class="btn btn-success btn-xs">Tombol</a>
```

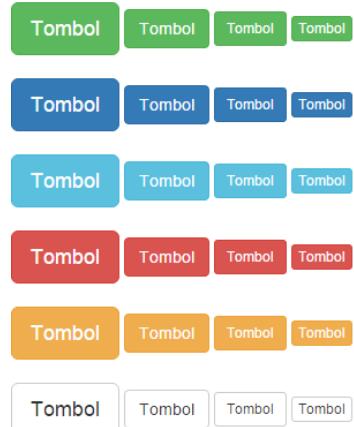
```

15   <br/><br/>
16   <a href="#" class="btn btn-primary btn-lg">Tombol</a>
17   <a href="#" class="btn btn-primary btn-md">Tombol</a>
18   <a href="#" class="btn btn-primary btn-sm">Tombol</a>
19   <a href="#" class="btn btn-primary btn-xs">Tombol</a>
20   <br/><br/>
21   <a href="#" class="btn btn-info btn-lg">Tombol</a>
22   <a href="#" class="btn btn-info btn-md">Tombol</a>
23   <a href="#" class="btn btn-info btn-sm">Tombol</a>
24   <a href="#" class="btn btn-info btn-xs">Tombol</a>
25   <br/><br/>
26   <a href="#" class="btn btn-danger btn-lg">Tombol</a>
27   <a href="#" class="btn btn-danger btn-md">Tombol</a>
28   <a href="#" class="btn btn-danger btn-sm">Tombol</a>
29   <a href="#" class="btn btn-danger btn-xs">Tombol</a>
30   <br/><br/>
31   <a href="#" class="btn btn-warning btn-lg">Tombol</a>
32   <a href="#" class="btn btn-warning btn-md">Tombol</a>
33   <a href="#" class="btn btn-warning btn-sm">Tombol</a>
34   <a href="#" class="btn btn-warning btn-xs">Tombol</a>
35   <br/><br/>
36   <a href="#" class="btn btn-default btn-lg">Tombol</a>
37   <a href="#" class="btn btn-default btn-md">Tombol</a>
38   <a href="#" class="btn btn-default btn-sm">Tombol</a>
39   <a href="#" class="btn btn-default btn-xs">Tombol</a>
40 </body>
41 </html>

```

HASIL

Cara Membuat tombol dengan Bootstrap



ANALISIS

Membuat Pesan Alert Dengan Bootstrap

Bootstrap menyediakan class khusus untuk membuat pesan alert dengan sangat mudah. di buktikan dengan kemudahan penggunaan bootstrap yang hanya tinggal menambahkan nama-nama class yang sudah di sediakan oleh bootstrap. untuk cara membuat pesan alert dengan bootstrap caranya cukup mudah. anda hanya perlu menambahkan class “alert” untuk mendefinisikan penggunaan alert bootstrap. selanjutnya tambahkan satu class lagi untuk menentukan jenis alert apa yang di gunakan. alert-success, alert-danger, alert-info dan alert-warning. berikut ini di jelaskan tentang penggunaan class alert bootstrap.

- **alert-success**

class alert-success di gunakan untuk membuat pesan alert yang berwarna hijau, biasanya pesan alert ini di gunakan untuk membuat pesan sukses.

- **alert-info**

class alert-info di gunakan untuk membuat pesan alert yang berwarna biru, biasanya pesan alert ini di gunakan untuk membuat pesan yang berupa informasi.

- **alert-warning**

class alert-warning di gunakan untuk membuat pesan alert yang berwarna kuning, biasanya pesan alert ini di gunakan untuk membuat pesan yang berupa peringatan.

- **alert-danger**

class alert-danger di gunakan untuk membuat pesan alert yang berwarna merah, biasanya pesan alert ini di gunakan untuk membuat pesan gagal.

Untuk membuat pesan alert dengan bootstrap caranya sangat mudah, buat sebuah element <div> yang berisi pesan peringatan alert yang ingin di buat. kemudian pada tag pembuka <div> berikan class “alert” dan satu lagi gunakan class alert seperti penjelasan di atas. bisa menggunakan alert-success untuk membuat pesan alert sukses, alert-info untuk membuat pesan alert informasi, alert-warning untuk membuat pesan alert berupa peringatan dan alert-danger untuk membuat pesan alert gagal. perhatikan contoh berikut ini.

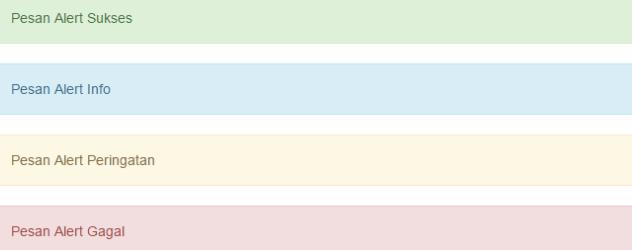
```

<!DOCTYPE html>
<html>
<head>
    <title>Bootstrap Part 6 : Membuat Pesan Alert Dengan Bootstrap</title>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
    <script type="text/javascript" src="js/jquery.js"></script>
    <script type="text/javascript" src="js/bootstrap.js"></script>
</head>
<body>
    <div class="container">
        <h1>Membuat Pesan Alert Dengan Bootstrap </h1>
        <div class="alert alert-success">Pesan Alert Sukses</div>
        <div class="alert alert-info">Pesan Alert Info</div>
        <div class="alert alert-warning">Pesan Alert Peringatan</div>
        <div class="alert alert-danger">Pesan Alert Gagal</div>
    </div>
</div>
</body>
</html>

```

HASIL

Membuat Pesan Alert Dengan Bootstrap



ANALISIS :

Dapat di lihat pada contoh di atas, pesan alert dengan bootstrap sudah jadi. tetapi seperti nya masih ada yang kurang. akan lebih bagus jika kita tambahkan tombol silang (x) pada pesan alert sehingga jika tombol silang di klik maka pesan alert akan hilang.

Membuat Tombol Close Pada Pesan Alert Bootstrap

perhatikan contoh berikut untuk membuat tombol close pada alert bootstrap.

```

<!DOCTYPE html>
<html>
<head>

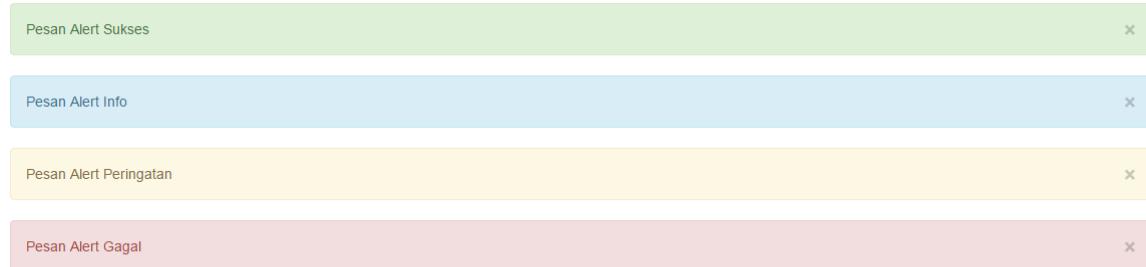
```

```

<title>Bootstrap Part 6 : Membuat Pesan Alert Dengan Bootstrap</title>
<link rel="stylesheet" type="text/css" href="css/bootstrap.css">
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript" src="js/bootstrap.js"></script>
</head>
<body>
    <div class="container">
        <h1>Membuat Pesan Alert Dengan Bootstrap </h1>
        <div class="alert alert-success fade in">
            <a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a>
            Pesan Alert Sukses
        </div>
        <div class="alert alert-info">
            <a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a>
            Pesan Alert Info
        </div>
        <div class="alert alert-warning">
            <a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a>
            Pesan Alert Peringatan
        </div>
        <div class="alert alert-danger fade in">
            <a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a>
            Pesan Alert Gagal
        </div>
    </div>
</body>
</html>

```

HASIL :



ANALISIS :

Perhatikan pada contoh di atas. sudah terdapat tombol close untuk menghapus atau menutup pesan alert. untuk membuat tombol close pada pesan alert tambahkan

```
<a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a>
```

sedikit tambahan lagi yang harus di perhatikan. kita bisa memberikan efek pada saat pesan alert di tutup. yaitu dengan menambahkan class “fade in” pada element div. efek fade merupakan efek fade in merupakan efek menutup pesan alert dengan efek memudar.

```
<div class="alert alert-success fade in">
```

TEMPLATE DAN CODEIGNITER

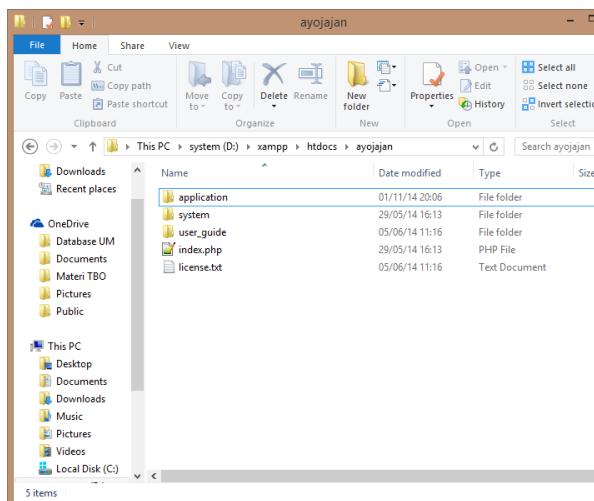
Tujuan Praktikum

1. Mampu menggunakan template yang sudah ada kedalam framework CI

Pendahuluan

Cara untuk memasukan template ke dalam Code Igniter, tempalate pada codeigniter tersebut bertujuan untuk mempercantik web, dengan adanya template ini maka pengunjung website kita nantinya akan betah untuk berlama-lama didalam website kita. berikut ini adalah langkah-langkah untuk memasukan template kedalam code ignaiter:

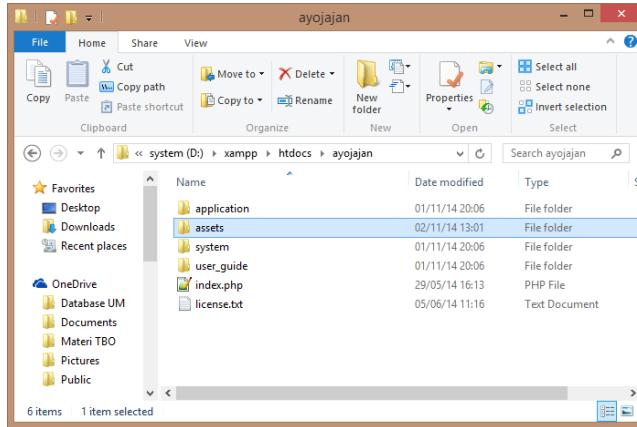
- kita persiapkan dahulu bahan-bahannya, seperti xampp, code igniter dan template
- Jika sudah didownload codeigniternya lalu di ekstrak codeigniternya, copy file yang didalam folder tersebut yang terdiri dari application, system dan user_guides kedalam folder htdocs pada xampp



Isi CodeIgniter

- Pada direktori xampp/htdocs yang telah di ekstrak file Code Igniter, pada tutorial kali ini saya menggunakan Code Igniter 2.2.0 , folder Code Igniter saya rename menjadi ayojajan
- setelah itu persiapkan template, pada laporan kali ini saya menggunakan template yang saya dapatkan dari mesin pencari google. berhubung filenya dalam bentuk zip jadi harus di ekstrak terlebih dahulu.

- pada direktori ayojajan buat direktori baru, beri nama folder tersebut ‘assets’



- setelah itu kita copy template yang telah diekstrak kedalam direktori assets
- sebelumnya kita harus menkonfigurasi dahulu konfigurasi pada codeignaiter, Buka file config.php (ayojajan\application\config\config.php), isi base url

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

- Selanjutnya buka file autoload.php (ayojajan\application\config\autoload.php), tambahkan ('url') didalam helper

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

- lalu setelah itu kita kemabli lagi pada folder asset, lalu nanti file yang akan dijadikan template kita bagi menjadi 3 yaitu header, footer dan konten. Header dan footer adalah bagian dari website yang bersifat statis atau tidak berubah, untuk itu kita akan membuat secara terpisah. Buat folder main (nama foldernya terserah) di dalam views, kemudian di dalam folder main kita buat file header.php dan footer.php.
- buka file index.html. lalu copy dari baris awal sampai ada tag penutup header, kemudian paste ke header.php

```
87
88
89
90


</div>
</nav>
</div>
</header>


```

- lalu kita akan membuat footernya, copy dari tag footer sampai habis, lalu paste ke footer.php

```
238
239
240
241
242
<footer>
<div class="wrapper">
<ul class="widget-cols clearfix">
<li class="first-col">
```

- Lalu sisanya kita copy lalu kita simpan dengan nama home.php
- setelah header dan footernya terbuat, pada file header.php dan footer.php kita cari file yang mengarah untuk memanggil file secara lokal seperti img, js dan css, tambahkan

kata <?php echo base_url(); ?>assets/ pada setiap yang merujuk untuk link membuka file, seperti ini

```
<link rel="stylesheet" media="all" href="<?php echo base_url(); ?>assets/css/style.css"/>
<meta name="viewport" content="width=device-width, initial-scale=1"/>
<!-- Adding "maximum-scale=1" fixes the Mobile Safari auto-zoom bug: http://filamentgroup.com/examples/iosScaleBug/ -->

<!-- JS -->
<script src="<?php echo base_url(); ?>assets/js/jquery-1.6.4.min.js"></script>
<script src="<?php echo base_url(); ?>assets/js/css3-mediaqueries.js"></script>
<script src="<?php echo base_url(); ?>assets/js/custom.js"></script>
<script src="<?php echo base_url(); ?>assets/js/tabs.js"></script>
```

- selanjutnya kita akan mengatur controller pada ci, copy welcome.php (ayojajan\application\controllers\welcome.php), lalu rename menjadi main.php, lalu diubah menjadi seperti ini

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

class Main extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
    }
    public function index()
    {
        $this->load->view('main/header');
        $this->load->view('main/home', );
        $this->load->view('main/footer');
    }
}
```

- lalu untuk membuat main.php sebagai default website, bisa diubah pada file routes.php (ayojajan\application\config\routes.php), ubah menjadi seperti ini.

```
$route['default_controller'] = "main";
```

jadi nanti ketika website dibuka, maka controller yang pertama kali dicari oleh ci adalah controller template.

- untuk mengetest website yang telah dibuatkan templatnya, bisa diakses pada browser ketik url <http://localhost/ayojajan> tapi sebelumnya jangan lupa untuk mengaktifkan terlebih dahulu xamppnya



- nah kini website pada codeigniter kita telah memiliki template. untuk memanggil konten yang lainnya yang kita perlukan hanya mengganti file yang home saja, lalu kita panggil melalui controller.

TUTORIAL MEMBUAT CRUD CODEIGNITER OTOMATIS DENGAN GROCERY

Tujuan Praktikum

1. Mampu menggunakan grocery crud untuk membuat crud otomatis pada yang sudah ada kedalam framework CI

Pendahuluan Grocery Crud

GROCERY CRUD adalah library open source yang digunakan untuk melakukan generate fitur CRUD (Create Read Update Delete) secara otomatis pada aplikasi berbasis web yang dibuat menggunakan framework codeigniter atau sering juga disebut dengan sebutan codeigniter CRUD Generator, karena hanya cukup dengan 5 baris script saja anda sudah mempunyai fitur CRUD yang powerfull. tentunya ini akan sangat bermanfaat jika anda sedang mengerjakan sebuah proyek yang memiliki banyak tabel master, dari pada membuat CRUD nya satu persatu lebih baik pakai GROCERY CRUD saja untuk menghemat waktu penggerjaan proyek.

Integrasi Grocery CRUD Dengan Codeigniter

Untuk bisa menggunakan library ini tentunya kita harus melakukan integrasi GROCERY CRUD dengan proyek kita terlebih dahulu, silahkan ikuti intruksi berikut:

1. Download library GROCERY CRUD pada situs resminya grocerycrud.com.
2. extract grocery-crud-1.5.7.zip kedalam project codeigniter.

Konfigurasi Dasar Codeigniter

untuk codeigniter yang baru di install/ fresh install maka ada beberapa konfigurasi sederhana yang harus anda dilakukan, salah satunya adalah setting base url nya, settingan ini bisa anda temukan pada application/config/config.php dan cari settingan berikut:

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

diganti menjadi :

	\$config['base_url'] = 'http://localhost/ci3/' ;
selanjutnya pengaturan koneksi database yang bisa anda temukan pada	application/config/database.php

1	\$db['default'] = array(
2	'dsn' => '',
3	'hostname' => 'localhost',
4	'username' => 'root',
5	'password' => '',
6	'database' => 'example',
7	'dbdriver' => 'mysqli',
8	'dbprefix' => '',
9	'pconnect' => FALSE,
10	'db_debug' => (ENVIRONMENT !== 'production'),
11	'cache_on' => FALSE,
12	'cachedir' => '',
13	'char_set' => 'utf8',
14	'dbcollat' => 'utf8_general_ci',
15	'swap_pre' => '',
16	'encrypt' => FALSE,
17	'compress' => FALSE,
18	'stricton' => FALSE,
19	'failover' => array(),
20	'save_queries' => TRUE
21) ;

setelah itu kita perlu melakukan load otomatis untuk library dan helper yang akan sering kita gunakan, anda bisa melakukan konfigurasi ini pada file application/config/autoload.php :

	\$autoload['libraries'] = array('grocery_CRUD','database');
	\$autoload['helper'] = array('url');

sampai pada tahap ini anda sudah selesai melakukan konfigurasi codeigniter dan GROCERY CRUD.

Membuat Database Dan Tabel

GROCERY CRUD menyediakan database bawaan yang bisa kita gunakan untuk praktik, anda bisa menemukan file examples_database.sql.zip didalam source code GROCERY

CRUD yang sudah anda download tadi, silahkan extract file tersebut lalu buatlah sebuah database baru dengan nama example dan lakukan import database melalui PhpMyadmin.

Table	Action	Rows	Type	Collation	Size	Overhead
actor		288	InnoDB	utf8_general_ci	16 Kib	-
category		16	MyISAM	utf8_general_ci	2.3 Kib	-
customers		122	MyISAM	latin1_swedish_ci	17.7 Kib	-
employees		23	MyISAM	latin1_swedish_ci	3.7 Kib	-
film		1,000	MyISAM	utf8_general_ci	141.5 Kib	-
film_actor		5,462	InnoDB	utf8_general_ci	224 Kib	-
film_category		2,016	MyISAM	utf8_general_ci	14.8 Kib	-
offices		7	MyISAM	latin1_swedish_ci	2.5 Kib	-
orderdetails		2,996	MyISAM	latin1_swedish_ci	132.6 Kib	-
orders		326	MyISAM	latin1_swedish_ci	22.7 Kib	-
payments		273	MyISAM	latin1_swedish_ci	16.5 Kib	-
productlines		7	MyISAM	latin1_swedish_ci	5.3 Kib	-
products		110	MyISAM	latin1_swedish_ci	33.1 Kib	-
13 tables	Sum	12,558	InnoDB	latin1_swedish_ci	632.7 Kib	0 B

Cara Menggunakan GROCERY CRUD

Salah satu kelebihan grocery crud yang powerfull adalah hanya dengan satu function dalam sebuah controller sudah bisa menghandle sebuah proses CRUD untuk sebuah tabel, misalnya anda punya sebuah function employees dalam controller Data maka anda sudah mempunyai fitur CRUD untuk tabel employees, dan jika anda ingin tambahkan fitur CRUD lain maka anda bisa membuat function baru pada controller yang sama sehingga lebih hemat file dan coding yang ditulis, sekarang silahkan buat sebuah file baru dengan nama Data.php pada folder controller dan tulis script berikut :

```

1 <?php
2
3     Class Data extends CI_Controller {
4
5         function __construct() {
6             parent::__construct();
7         }
8
9         function index() {
10
11     }
12
13     public function employees() {
14         // instance object
15         $crud = new grocery_CRUD();
16         // pilih tabel yang akan digunakan
17         $crud->set_table('employees');
18         // simpan hasilnya kedalam variabel output
19         $output = $crud->render();
20         // tampilkan di view
21         // $this->example_output($output);
22         $this->load->view('template.php', $output);
23     }
24 }
```

22	}
23	}
24	}
25	

langkah selanjutnya kita akan membuat sebuah view untuk menampilkan hasil generate dari GROCERY CRUD, sebenarnya library ini sudah menyediakan sebuah view dengan nama example.php tapi terkadang kita punya template sendiri yang sudah kita buat sedemikian rupa untuk di integrasikan dengan proyek yang sedang kita kerjakan, oleh karena itu silahkan buat sebuah file baru dengan nama template.php pada folder view dan tulislah script berikut:

1	<!DOCTYPE html>
2	<html>
3	<head>
4	<title>GROCERY GRUD - Belajarphp.net</title>
5	<meta charset="utf-8" />
6	<meta name="viewport" content="width=device-width, initial-
7	scale=1.0">
8	<?php foreach (\$css_files as \$file): ?>
9	<link type="text/css" rel="stylesheet" href="<?php echo \$file; ?>" />
10	<?php endforeach; ?>
11	<?php foreach (\$js_files as \$file): ?>
12	<script src="<?php echo \$file; ?>"></script>
13	<?php endforeach; ?>
14	</head>
15	<body>
16	<div style='height:20px;'></div>
17	<div>
18	<?php echo \$output; ?>
19	</div>
20	</body>
	</html>

setelah itu jangan lupa melakukan save untuk menyimpan perubahan lalu untuk melihat hasilnya silahkan buka web browser dan akses url <http://localhost/ci3/index.php/data/employees>

Employee Data							Actions
Last Name	First Name	Extension	Email	Office Code	File URL	Job Title	
Murphy	Diane	x5800	dmurphy@classicmodelcars.com	1		President	
Patterson	Mary	x4611	mpatterso@classicmodelcars.com	1		VP Sales	
Firrelli	Jeff	x9273	jfirrelli@classicmodelcars.com	1		VP Marketing	
Patterson	William	x4871	wpatterson@classicmodelcars.com	6		Sales Manager (APAC)	
Bondur	Gerard	x5408	gbondur@classicmodelcars.com	4	pdf-test.pdf	Sale Manager (EMEA)	
Bow	Anthony	x5428	abow@classicmodelcars.com	1		Sales Manager (NA)	
Jennings	Leslie	x3291	ljennings@classicmodelcars.com	1		Sales Rep	
Thompson	Leslie	x4065	lthompson@classicmodelcars.com	1		Sales Rep	
Firrelli	Julie	x2173	jfirrelli@classicmodelcars.com	2	test-2.pdf	Sales Rep	
Patterson	Steve	x4334	spatterson@classicmodelcars.com	2		Sales Rep	

Search: Search all Clear filtering

Show 10 entries Page 1 of 3 Displaying 1 to 10 of 23 items

Menampilkan Custom Field

jika pada script di atas kita menampilkan semua data dalam tabel employees maka kali ini kita akan belajar cara menampilkan beberapa kolom saja, GROCERY CRUD sudah menyediakan fungsi **columns** yang digunakan untuk menampilkan custom Field pada sebuah tabel, sebagai contoh misalnya kita hanya ingin menampilkan field firstName, lastName dan email dari tabel employees maka script nya menjadi seperti ini:

```

1  public function employees() {
2      // instance object
3      $crud = new grocery_CRUD();
4      // pilih tabel yang akan digunakan
5      $crud->set_table('employees');
6      // custom field yang ingin ditampilkan
7      $crud->columns('lastName', 'firstName', 'email');
8      // simpan hasilnya kedalam variabel output
9      $output = $crud->render();
10     // tampilkan di view
11     $this->load->view('template.php', $output);
12 }
```

silahkan save dan refresh web browser untuk melihat perubahan, sekarang hasilnya menjadi seperti ini

Employee Data			
Last Name	First Name	Email	Actions
Murphy	Diane	dmurphy@classicmodelcars.com	
Patterson	Mary	mpatterso@classicmodelcars.com	
Firrelli	Jeff	jfirrelli@classicmodelcars.com	
Patterson	William	wpatterson@classicmodelcars.com	
Bondur	Gerard	gbondura@classicmodelcars.com	
Bow	Anthony	abow@classicmodelcars.com	
Jennings	Leslie	ljennings@classicmodelcars.com	
Thompson	Leslie	lthompson@classicmodelcars.com	
Firrelli	Julie	jfirrelli@classicmodelcars.com	
Patterson	Steve	spatterson@classicmodelcars.com	

Search: Search all Clear filtering

Show 10 entries Page 1 of 3 Displaying 1 to 10 of 23 items

Menggunakan Validation

jika anda sudah terbiasa menggunakan library validation maka pada GROCERY CRUD juga anda bisa melakukan hal yang sama untuk melakukan validasi data sebelum diproses, sebagai contoh misalnya kita ingin membuat validasi inputan email yang dimasukan oleh users harus berformat email maka kita bisa menggunakan fungsi set_rule seperti dibawah ini

```

1  public function employees() {
2      // instance object
3      $crud = new grocery_CRUD();
4      // pilih tabel yang akan digunakan
5      $crud->set_table('employees');
6      // custom field yang ingin ditampilkan
7      $crud->columns('firstName','lastName','email');
8      // set validation rule
9      $crud->set_rules('email','email','valid_email');
10     $crud->set_rules('firstName','firstName','required');
11     // simpan hasilnya kedalam variabel output
12     $output = $crud->render();
13     // tampilkan di view
14     $this->load->view('template.php', $output);
15 }
```

jangan lupa lakukan save dan selanjutnya buka form untuk add data baru, silahkan kosongkan kolom firstName dan berikan format email yang salah pada kolom email kemudian klik save maka akan muncul warning seperti dibawah ini

The screenshot shows a web application window titled "GROCERY GRUD - Belajar". The URL in the address bar is "localhost/ci3/index.php/data/employees/add". The main content is a form titled "Add Record" with the following fields:

- Last Name : Akbar
- First Name : (highlighted in red)
- Extension : SST
- Email : nurisakbar.com (highlighted in yellow)
- Office Code : 1
- File URL :
- Job Title : Programmer (highlighted in yellow)

Below the form, there are two error messages in a red-bordered box:

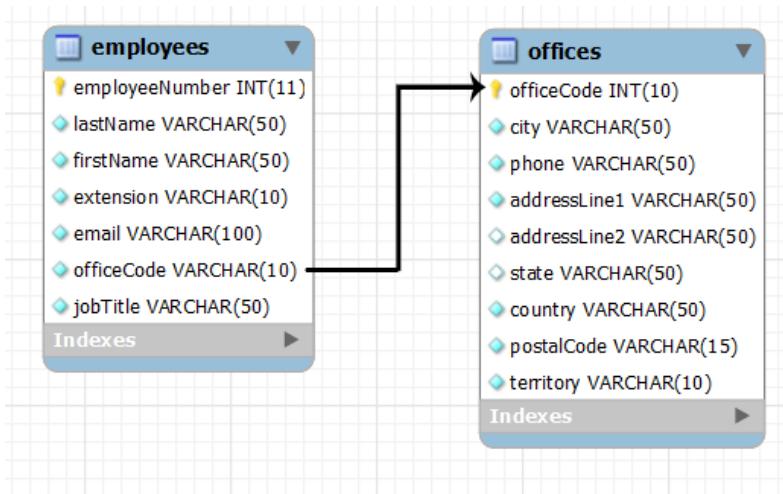
- ! The firstName field is required.
- ! The email field must contain a valid email address.

At the bottom of the form are three buttons: "Save" (highlighted in blue), "Save and go back to list", and "Cancel".

untuk referensi validation rule yang bisa anda gunakan silahkan mengacu pada dokumentasi validation rule codeigniter 3.

Menampilkan Data Dari tabel Yang Berelasi

Mungkin ini adalah salah satu bagian yang paling anda tunggu yaitu bagaimana cara nya menampilkan data dari tabel yang berelasi, jika anda mempelajari struktur database bawaan GROCERY CRU GENERATOR maka anda akan menemukan tabel employees dan offices, kedua tabel ini berelasi dengan officeCode sebagai PK pada tabel offices dan menjadi FK pada tabel employees seperti gambar dibawah ini



lalu anda ingin menampilkan field city sebagai pengganti pengganti field officeCode yang ada pada tabel employees sehingga scriptnya menjadi seperti ini

```

1  public function employees() {
2      // instance object
3      $crud = new grocery_CRUD();
4      // pilih tabel yang akan digunakan
5      $crud->set_table('employees');
6      // memberikan alias untuk kolom officeCode menjadi Office City
7      $crud->display_as('officeCode', 'Office City');
8      $crud->set_subject('Employee');
9      // mengambil data city dari tabel offices berdasarkan relasi dari
10     tabel officeCode
11     $crud->set_relation('officeCode', 'offices', 'city');
12     // simpan hasilnya kedalam variabel output
13     $output = $crud->render();
14     // tampilkan di view
15     $this->load->view('template.php', $output);
    }
  
```

silahkan save dan lakukan refresh pada web browser anda untuk melihat perubahan

The screenshot shows a web application titled "GROCERY GRUD - Belajar" running on "localhost/ci3/index.php/data/employees/". The page displays a table of employee records with the following columns: LastName, FirstName, Extension, Email, Office City, File url, Job Title, and Actions. The "Actions" column contains icons for search, edit, and delete. An additional "Office City" column is present in the table, indicating success in adding it via code. The table has 10 items displayed on page 1 of 3.

Last Name	First Name	Extension	Email	Office City	File url	Job Title	Actions
Murphy	Diane	x5800	dmurphy@classicmodelcars.com	San Francisco		President	
Patterson	Mary	x4611	mpatterso@classicmodelcars.com	San Francisco		VP Sales	
Firrelli	Jeff	x9273	jfirrelli@classicmodelcars.com	San Francisco		VP Marketing	
Patterson	William	x4871	wpatterson@classicmodelcars.com	Sydney		Sales Manager (APAC)	
Bondur	Gerard	x5408	gbondur@classicmodelcars.com	Paris	pdftest.pdf	Sale Manager (EMEA)	
Bow	Anthony	x5428	abow@classicmodelcars.com	San Francisco		Sales Manager (NA)	
Jennings	Leslie	x3291	ljennings@classicmodelcars.com	San Francisco		Sales Rep	
Thompson	Leslie	x4065	lthompson@classicmodelcars.com	San Francisco		Sales Rep	
Firrelli	Julie	x2173	jfirrelli@classicmodelcars.com	Boston	test-2.pdf	Sales Rep	
Patterson	Steve	x4334	spatterson@classicmodelcars.com	Boston		Sales Rep	

sekarang field office City sudah muncul, itu tandanya kita sudah berhasil.

Upload File/ Upload Field

Kasus selanjutnya adalah bagaimana caranya untuk membuat field yang akan kita gunakan sebagai interface untuk mengupload file ke server. dalam hal ini GROCERY CRUD Generator sudah menyediakan function set_field_upload dengan format seperti dibawah ini:

```
$crud->set_field_upload('NamaField','lokasiTujuan');
```

sehingga jika kita implementasikan secara menyeluruh maka script CRUD nya menjadi seperti ini

```

1  public function employees() {
2      // instance object
3      $crud = new grocery_CRUD();
4      // pilih tabel yang akan digunakan
5      $crud->set_table('employees');
6      // memberikan alias untuk kolom officeCode menjadi Office City
7      $crud->display_as('officeCode', 'Office City');
8      $crud->set_subject('Employee');
9      // mengambil data city dari tabel offices berdasarkan relasi dari
10     tabel officeCode
11     $crud->set_relation('officeCode', 'offices', 'city');
12     // menjadikan field file_url sebagai elemen upload pada form input
13     data
14     $crud->set_field_upload('file_url','assets/uploads/files');
15     // simpan hasilnya kedalam variabel output
16     $output = $crud->render();
17     // tampilkan di view
      $this->load->view('template.php', $output);

```

```
}
```

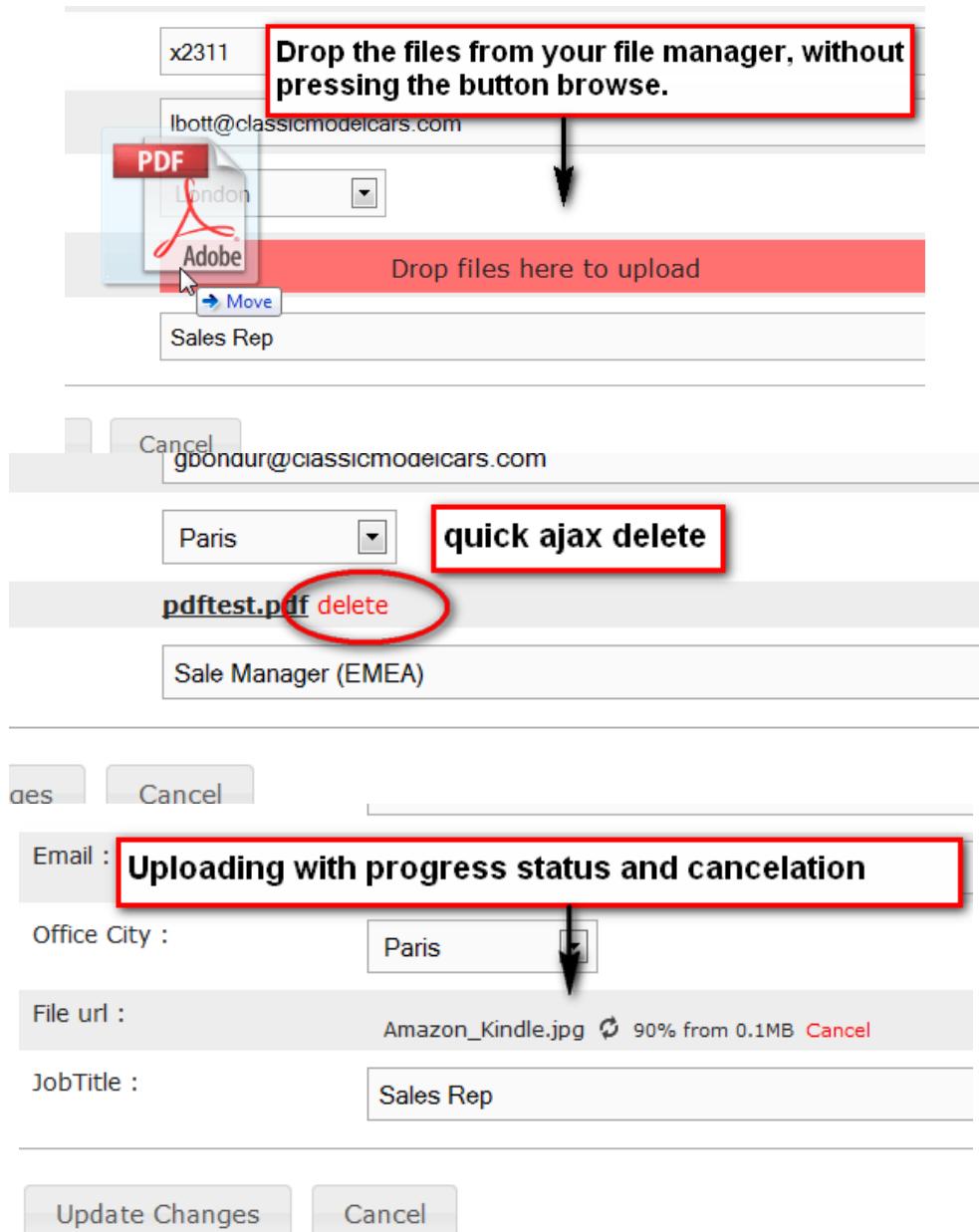
silahkan save dan refresh untuk melihat perubahannya, seperti yang anda lihat sekarang anda bisa langsung mengdownload file yang sudah anda upload melalui halaman baca data seperti dibawah ini

Office City	File url	JobTitle
Paris	pdftest.pdf	Sale Manager (EMEA)
Paris		Sales Rep
London		Sales Rep
San Francisco		Sales Manager (NA)
Paris		Sales Rep
San Francisco		VP Marketing

proses upload file menggunakan teknik ajax

The screenshot shows a user interface for file upload. At the top left, a red box highlights the text "ajax uploading". An arrow points from this box to a file input field labeled "Upload a file". Below the input field is another red box containing the text "same button interface for all the browsers". To the right of the input field are two buttons: "Update changes" and "Cancel". Above the input field is a dropdown menu labeled "Select Office City".

anda juga bisa melakukan drop file dan arahkan ke dalam form untuk melakukan proses upload



Menambahkan Button Action Pada Halaman Utama

secara default library GROCERY CRUD hanya menampilkan button untuk melihat detail data, button edit dan button delete, lalu jika anda ingin menambahkan custom button maka anda bisa menggunakan function add_action seperti dibawah ini

```
$crud->add_action('More', '', 'demo/action_more','ui-icon-plus');
```

sehingga jika kita implementasikan jadi seperti ini :

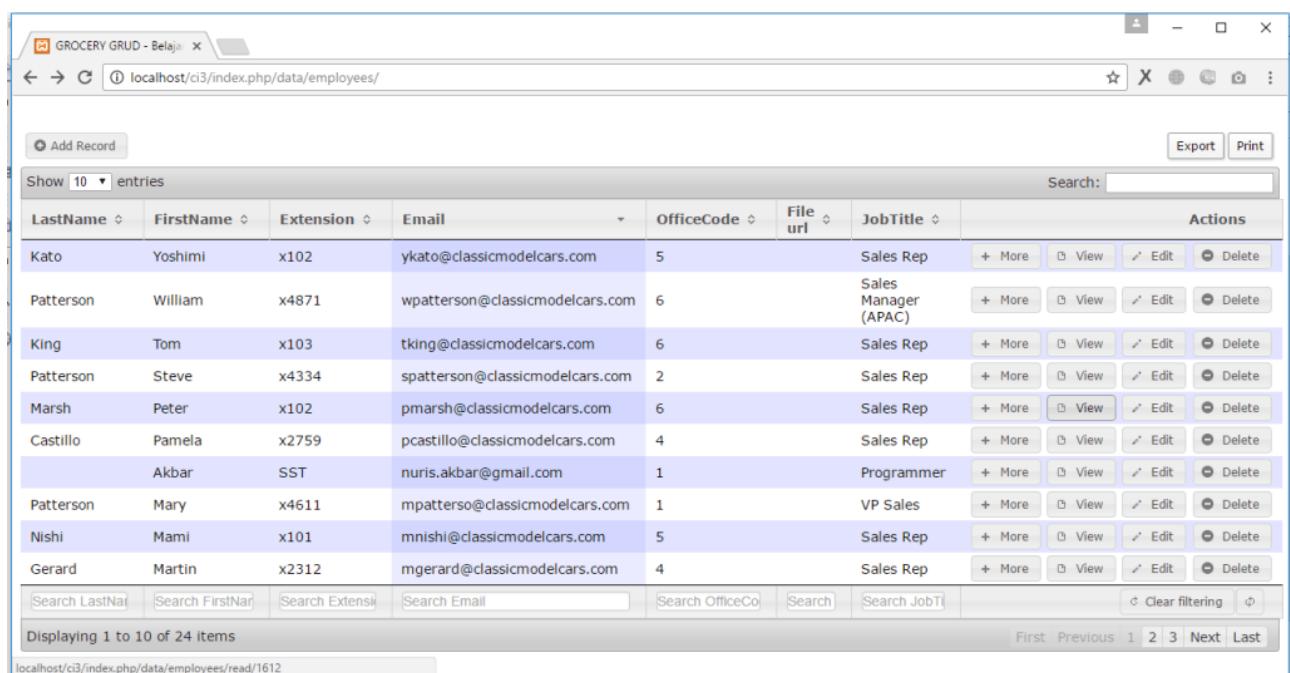
1	public function employees() {
---	-------------------------------

```

2          // instance object
3      $crud = new grocery_CRUD();
4      // ganti themes
5      $crud->set_theme('datatables');
6      // pilih tabel yang akan digunakan
7      $crud->set_table('employees');
8      // memberikan custom button
9      $crud->add_action('More', '', 'demo/action_more', 'ui-icon-plus');
10     // simpan hasilnya kedalam variabel output
11     $output = $crud->render();
12     // tampilkan di view
13     $this->load->view('template.php', $output);
14 }

```

jika anda lihat pada script line 5 maka anda akan menemukan fungsi set_theme, sesuai dengan nama nya bahwa fungsi ini digunakan untuk mengganti tema yang digunakan, GROCERY CRUD menyediakan 2 jenis theme yang bisa anda gunakan yaitu datatables dan flexigrid



Last Name	First Name	Extension	Email	Office Code	File URL	Job Title	Actions
Kato	Yoshimi	x102	ykato@classicmodelcars.com	5		Sales Rep	+ More View Edit Delete
Patterson	William	x4871	wpatterson@classicmodelcars.com	6		Sales Manager (APAC)	+ More View Edit Delete
King	Tom	x103	tking@classicmodelcars.com	6		Sales Rep	+ More View Edit Delete
Patterson	Steve	x4334	spatterson@classicmodelcars.com	2		Sales Rep	+ More View Edit Delete
Marsh	Peter	x102	pmarsh@classicmodelcars.com	6		Sales Rep	+ More View Edit Delete
Castillo	Pamela	x2759	pcastillo@classicmodelcars.com	4		Sales Rep	+ More View Edit Delete
	Akbar	SST	nuris.akbar@gmail.com	1		Programmer	+ More View Edit Delete
Patterson	Mary	x4611	mpatterso@classicmodelcars.com	1		VP Sales	+ More View Edit Delete
Nishi	Mami	x101	mnishi@classicmodelcars.com	5		Sales Rep	+ More View Edit Delete
Gerard	Martin	x2312	mgerard@classicmodelcars.com	4		Sales Rep	+ More View Edit Delete

Fitur Callback GROCERY

Untuk memudahkan developer dalam melakukan manipulasi bisniss proses maupun modifikasi tampilan, maka GROCERY CRUD menyediakan sebuah fitur yang disebut callback, dengan fitur ini kita bisa memberikan modifikasi pada textbox form atau membuat custom function yang akan dipanggil ketika sebuah proses di eksekusi.

contoh pertama adalah callback_add_field, fitur ini digunakan untuk memodifikasi kolom inputan, misalnya saya ingin memberikan kode area +62 sebelum textbox phone maka scriptnya jadi seperti ini

```
1 function employees(){
2     $crud = new grocery_CRUD();
3
4     $crud->set_table('offices');
5     $crud->set_subject('Office');
6     $crud->required_fields('city');
7     $crud->columns('city','country','phone','addressLine1','postalCode');
8
9     $crud->callback_add_field('phone',array($this,'add_field_callback_1'));
10
11    $output = $crud->render();
12
13    $this->load->view('template.php', $output);
14 }
15
16 function add_field_callback_1()
17 {
18     return '+62 <input type="text" maxlength="50" value="" name="phone" style="width:462px">';
19 }
```

The screenshot shows a web browser window with the URL <http://localhost/ci3/index.php/data/employees/add>. The page title is "Add Office". The form fields are as follows:

- City*: [empty input]
- Phone: +62 [input field]
- AddressLine1: [empty input]
- AddressLine2: [empty input]
- State: [empty input] (for U.S. only)
- Country: [empty input]
- PostalCode: [empty input]
- Territory: [empty input]

At the bottom of the form are three buttons: "Save", "Save and go back to list", and "Cancel". Above the form, there is a "Switch theme:" dropdown set to "Flexigrid".

contoh kedua adalah callback_edit_field, fitur ini digunakan untuk memodifikasi kolom pada form edit, beda nya pada function ini kita bisa mendapatkan value dari kolom yang di edit, contoh script nya seperti ini :

```
1     function offices() {
2         $crud = new grocery_CRUD();
3
4             $crud->set_table('offices');
5             $crud->set_subject('Office');
6             $crud->required_fields('city');
7             $crud->columns('city',      'country',      'phone',      'addressLine1',
8 'postalCode');
9             $crud->callback_edit_field('phone',           array($this,
10 'edit_field_callback_1'));
11             $output = $crud->render();
12             $this->load->view('template.php', $output);
13     }
14
15     function edit_field_callback_1($value, $primary_key) {
16         return '+62 <input type="text" maxlength="50" value="' . $value .
17 ' name="phone" style="width:462px">';
18 }
```

silahkan save dan refresh web browser yang anda gunakan untuk melihat perubahan

The screenshot shows a web browser window with the title "GROCERY GRUD - Belajar". The address bar displays the URL "localhost/ci3/index.php/data/offices/edit/1". The main content is a form titled "Edit Office" with the following fields:

Field	Value
City*	San Francisco
Phone	+62 +1 650 219 4782
AddressLine1	100 Market Street
AddressLine2	Suite 300
State	CA
Country	USA
PostalCode	94080
Territory	NA

At the bottom of the form are three buttons: "Update changes", "Update and go back to list", and "Cancel".

selanjutnya ada function callback_before_insert yang digunakan untuk memanggil function lain yang akan di eksekusi sebelum proses insert dilakukan, sebagai contoh kita akan melakukan pengecekan data kode POS yang di input oleh pengguna, jika kode pos nya dibiarkan kosong maka sistem akan menyimpan string “kosong” kedalam field, ini hanya sekedar contoh saja, tentu pada implementasinya bisa dilakukan hal lain yang lebih luas, berikut scriptnya seperti dibawah ini:

```
1  function offices() {
2      $crud = new grocery_CRUD();
3      $crud->set_table('offices');
4      $crud->set_subject('Office');
5      $crud->required_fields('city');
6      $crud->columns('city',      'country',      'phone',      'addressLine1',
7      'postalCode');
8      $crud->callback_before_insert(array($this, 'checking_post_code'));
9      $output = $crud->render();
10     $this->load->view('template.php', $output);
11 }
12
13     function checking_post_code($post_array) {
14         if (empty($post_array['postalCode'])) {
15             $post_array['postalCode'] = 'Not U.S.';
16         }
17         return $post_array;
}
```

selain callback_before_insert, masih ada juga fungsi lain yang bisa anda gunakan seperti :

1. callback_after_insert
2. callback_before_update dan callback_after_update
3. callback_after_delete dan callback_before_delete
4. callback_before_upload dan callback_after_upload

untuk keterangan lebih lanjut tentang fitur yang belum dibahas silahkan baca dokumentasi dari GROCERY CRUD.

Kesimpulan

dengan adanya library ini mempermudah pekerjaan developer dalam mengerjakan fitur CRUD dalam proyek yang sedang dikerjakan, performance nya pun terbilang cukup bagus karena sudah mengimplementasikan teknik ajax dalam memproses data.

REFERENSI
