



MODUL PRAKTIKUM WEB

SEKOLAH TINGGI TEKNOLOGI GARUT

ADE SUTEDI, M.KOM

MODUL I

PRAKTIKUM PENGEMBANGAN PLATFORM WEB

TUJUAN

- a. Mahasiswa memahami konsep dasar situs web,
- b. Mahasiswa mengetahui kegunaan Web Server serta contohnya,
- c. Mahasiswa mampu melakukan instalasi Web Server, Basis Data, dan FTP server.

KEBUTUHAN

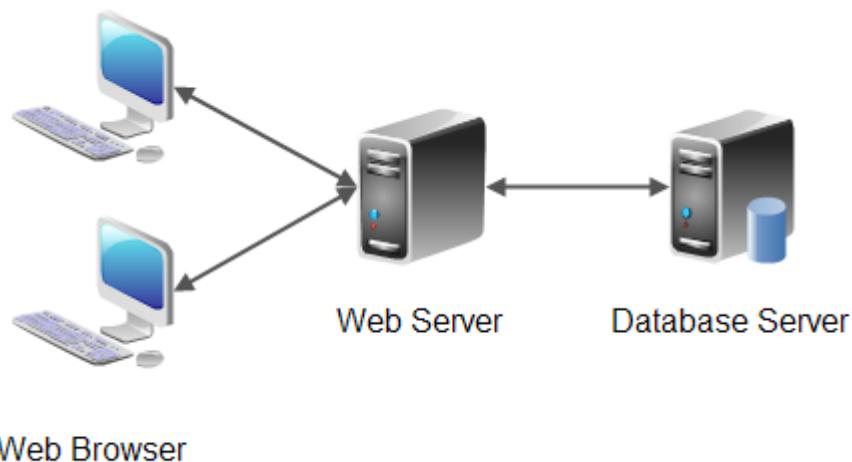
- a. Unit Komputer (Memiliki Sistem Operasi Windows, Linux, dll)
- b. Web server (dalam modul ini digunakan Apache Web Server dari modul XAMPP) yang bisa unduh di laman [ini](#),
- c. Web browser (Mozilla, Chrome, dll)

SEKILAS TENTANG WEB

“Situs web adalah suatu halaman web yang saling berhubungan yang umumnya berada pada peladen yang sama berisikan kumpulan informasi yang disediakan secara perorangan, kelompok, atau organisasi. Sebuah situs web biasanya ditempatkan setidaknya pada sebuah server web yang dapat diakses melalui jaringan seperti Internet, ataupun *Local Area Network* (LAN) melalui alamat Internet yang dikenali sebagai *Uniform Resource Locator* (URL).” (Sumber: Wikipedia)

Situs web biasanya ditempatkan pada server web yang umumnya telah dilengkapi dengan perangkat-perangkat lunak khusus untuk menangani pengaturan nama domain, serta menangani layanan atas protokol Hyper Text Transfer Protokol (HTTP).

Arsitektur Web sederhana memiliki tiga bagian utama diantaranya Client, jaringan penghubung (Intranet/ Internet), dan server yang ditunjukkan pada gambar berikut ini:



Gambar 1.1: Arsitektur Web sederhana

WEB SERVER

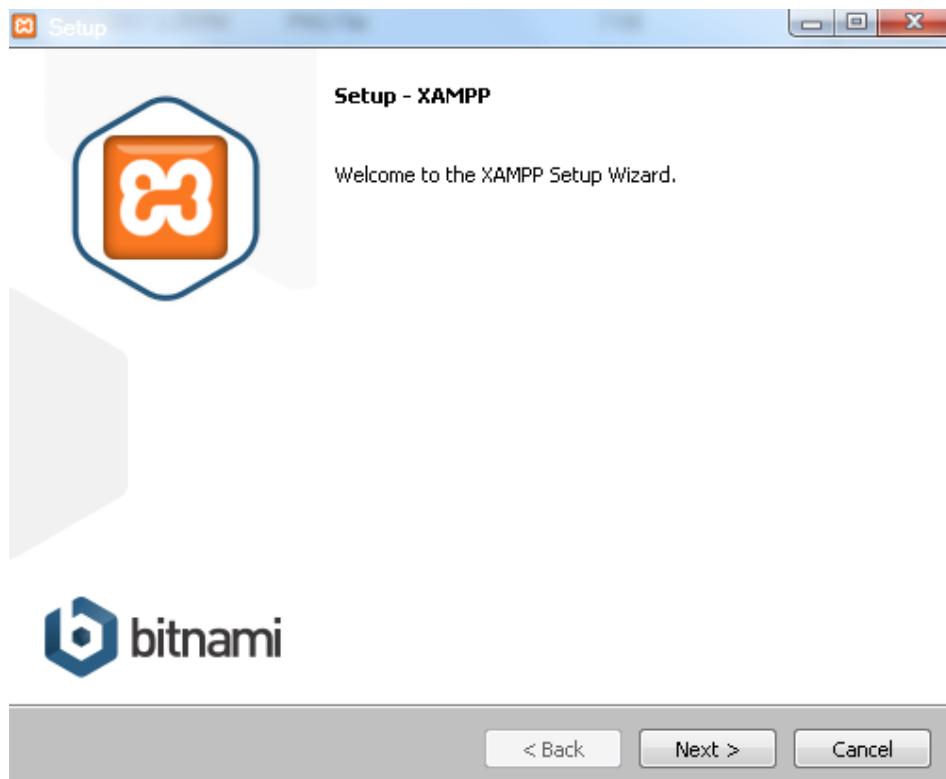
Server atau Web server adalah sebuah software yang memberikan layanan berbasis data dan berfungsi menerima permintaan dari HTTP atau HTTPS melalui web browser (Mozilla Firefox, Google Chrome) kemudian mengirimkannya kembali dalam bentuk halaman web dalam bentuk dokumen HTML atau lainnya. Contoh web server diantaranya: Apache HTTP Server, atau Internet Information Services (IIS).

1. Apa Itu XAMPP?

XAMPP merupakan aplikasi Cross platform: Apache, MySQL, PHP dan Perl. XAMPP juga memberikan solusi sederhana dan cukup ringan dijalankan, memungkinkan Anda membuat web server lokal untuk melakukan pengujian website. XAMPP dapat dijalankan pada Mac dan Linux. Dalam tutorial ini kami akan membahas pada Windows.

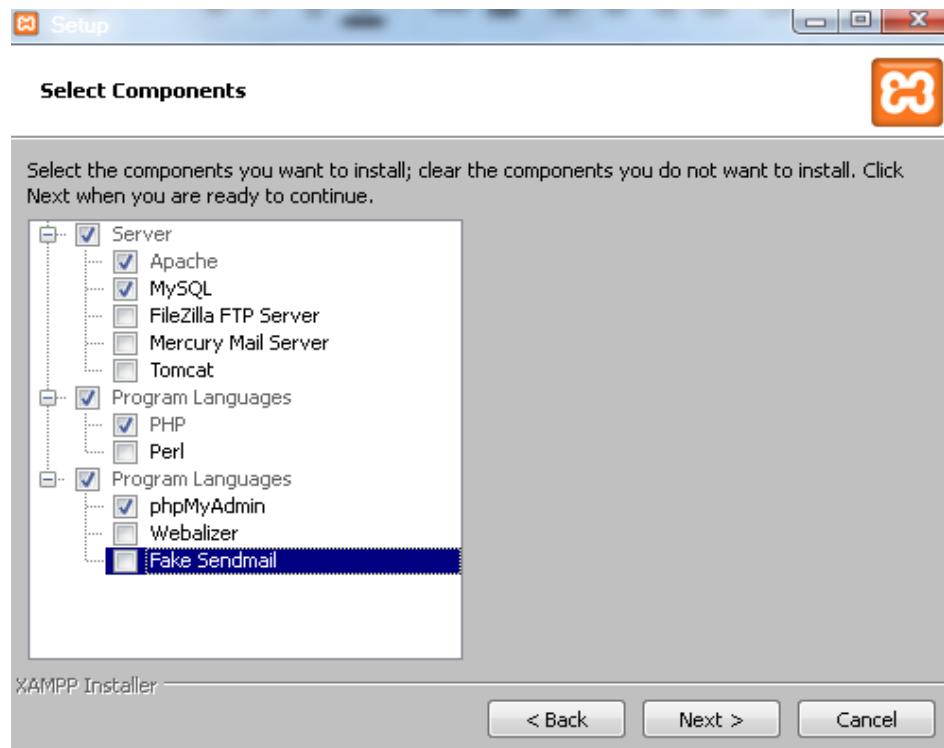
2. Instalasi XAMPP

- a. Lakukan instalasi setelah Anda selesai mengunduh. Selama proses instalasi mungkin Anda akan melihat pesan yang menanyakan apakah Anda yakin akan menginstalnya. Silakan tekan **Yes** untuk melanjutkan instalasi.
- b. Klik tombol **Next**. Perhatikan gambar 1.2



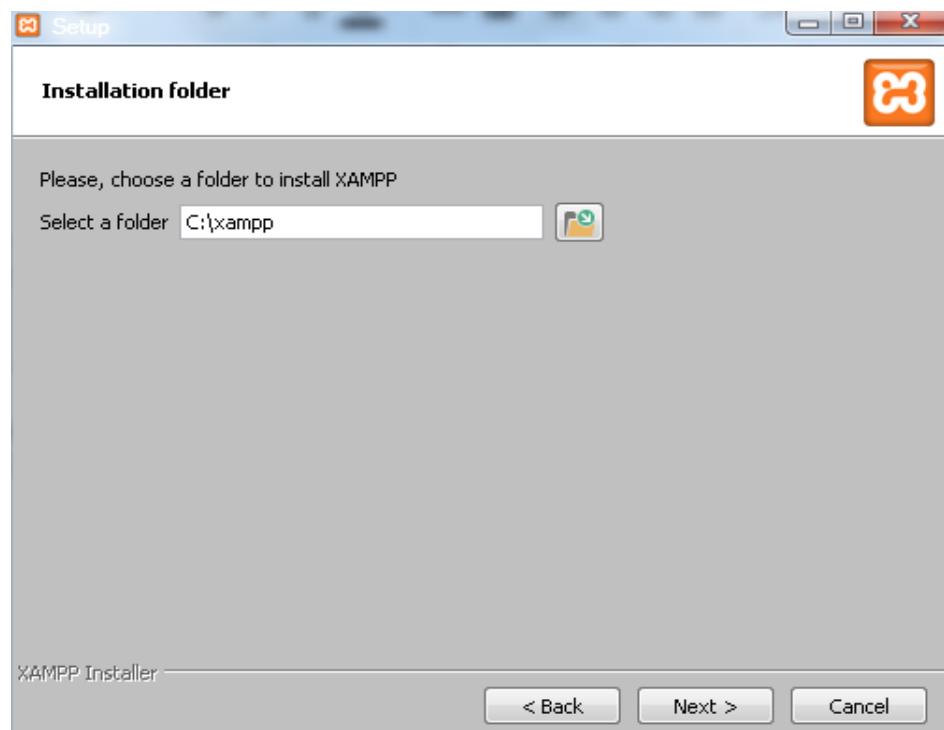
Gambar 1.2: Memulai instalasi XAMPP

- c. Pada tampilan selanjutnya akan muncul pilihan mengenai komponen mana saja yang ingin dan tidak ingin Anda install. Beberapa pilihan seperti Apache dan PHP adalah bagian penting untuk menjalankan website dan akan otomatis diinstal. Silakan centang MySQL dan phpMyAdmin, untuk transaksi basis data pada pilihan lainnya seperti ditunjukkan pada gambar 1.3.



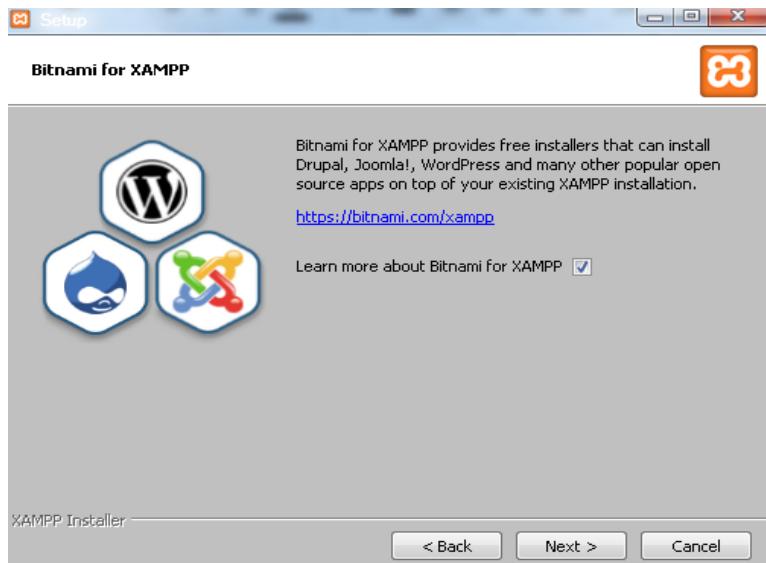
Gambar1.3: Memilih modul pendukung web server

- d. Berikutnya silakan pilih folder tujuan dimana XAMPP ingin Anda instal, pada tutorial ini pada direktori *C:\xampp*.



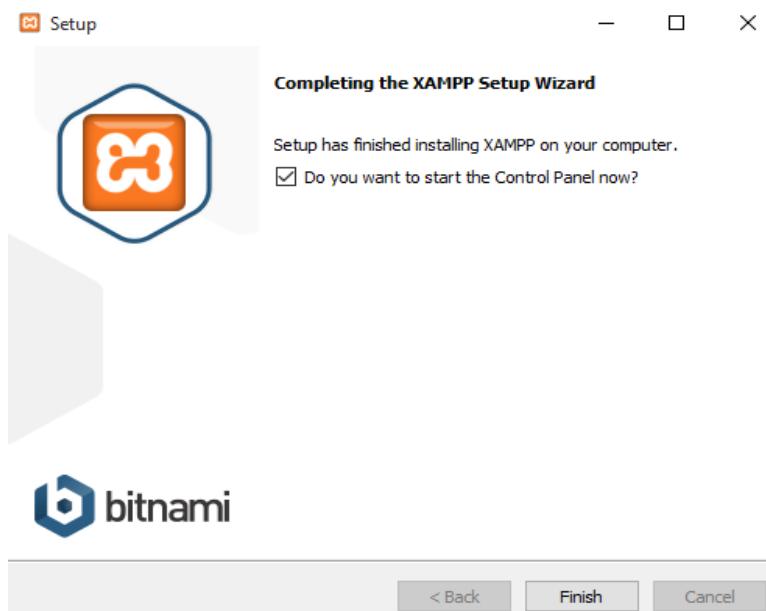
Gambar 1.4: direktori default folder modul program XAMPP

- e. Pada halaman selanjutnya, akan ada pilihan apakah Anda ingin menginstal paket Bitnami untuk XAMPP, dimana nantinya dapat Anda gunakan untuk install WordPress, Drupal, dan Joomla dan produk bitnami lainnya. Pada langkah ini proses instalasi XAMPP akan dimulai. Silakan klik tombol **Next**.



Gambar 1.5: Paket opsional unduh CMS

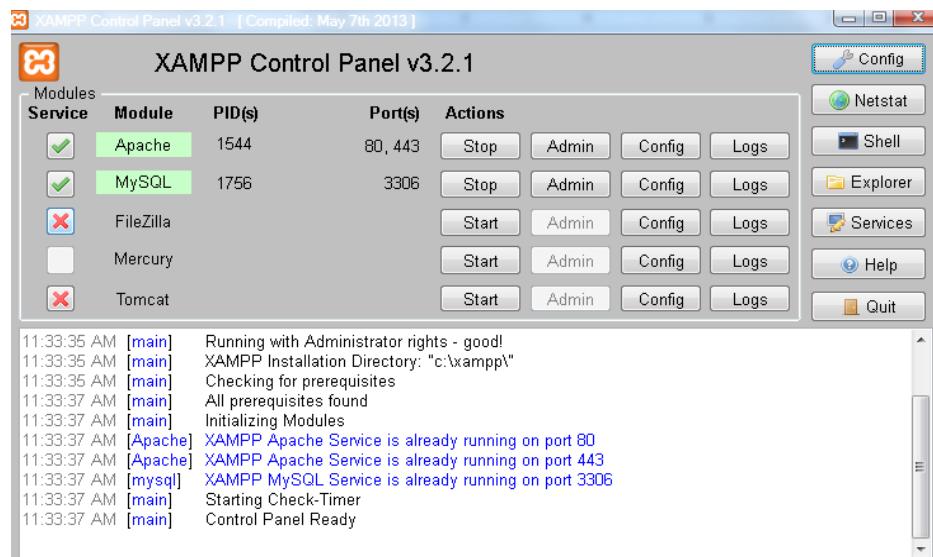
- f. Setelah berhasil diinstal, akan muncul notifikasi untuk langsung menjalankan control panel. Silakan klik **Finish** seperti pada gambar 1.6.



Gambar 1.6: Mengakhiri pemasangan paket XAMPP

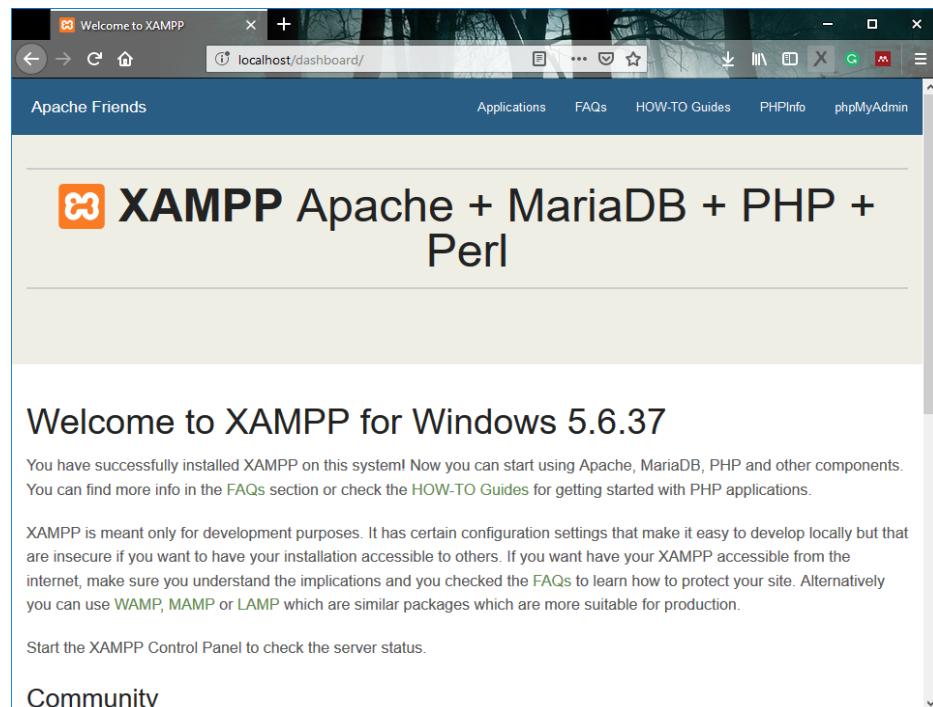
3. Jalankan XAMPP

Silakan buka aplikasi XAMPP kemudian klik tombol Start pada Apache dan MySQL. Jika berhasil dijalankan, Apache dan MySQL akan berwarna hijau seperti gambar di bawah ini.



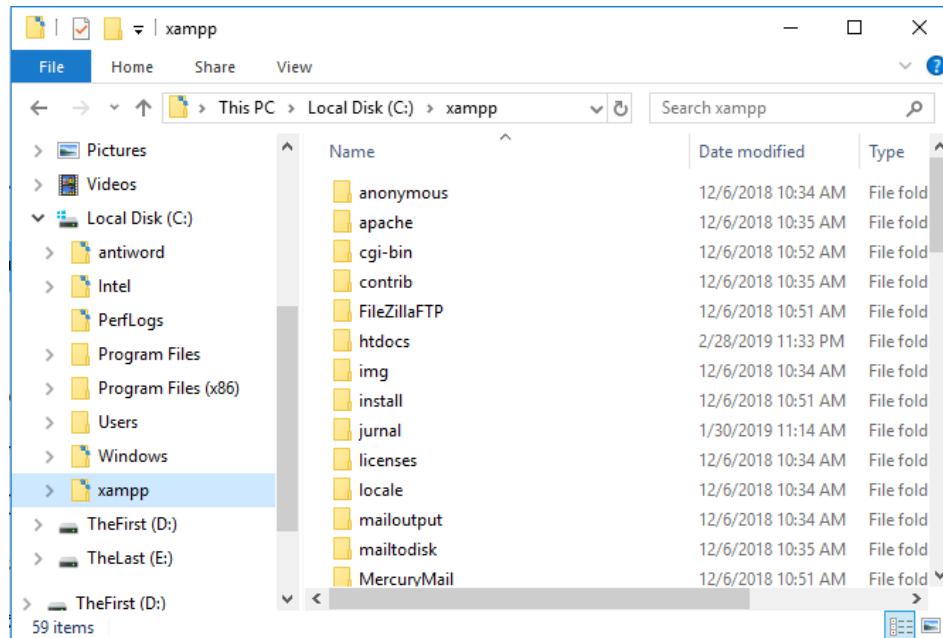
Gambar 1.7: Jendela menu XAMPP control panel

Untuk melakukan pengecekan, silakan akses link <http://localhost> melalui browser Anda. Jika berhasil, maka akan tampil seperti pada gambar 1.8.



Gambar 1.8: Tampilan halaman web default dari XAMPP

Jika kita perhatikan direktori XAMPP yang sudah dipasang, akan terlihat beberapa folder yang berisi modul-modul aplikasi yang dimiliki XAMPP. Pada praktikum ini, halaman web yang akan kita buat akan ditempatkan pada folder ./htdocs disertai folder baru untuk membedakan akses halaman web yang akan dibuat dengan halaman <http://localhost> tadi. Perhatikan gambar 1.9:



Gambar 1.9: Folder-folder aplikasi dalam paket XAMPP

TUGAS PRAKTIKUM 1

1. Sebutkan jenis-jenis aplikasi yang dapat dijadikan sebagai web server!
2. Coba pasang aplikasi XAMPP pada computer/ laptop masing-masing!
3. Jelasakan bagaimana agar aplikasi web server yang telah kita pasang berjalan otomatis tanpa harus dibuka secara manual?

MODUL II

PENGANTAR HYPER TEXT MARKUP LANGUAGE (HTML)

TUJUAN

- a. Mahasiswa mengetahui dasar-dasar HTML
- b. Mahasiswa mengetahui HTML Editor
- c. Mahasiswa mampu membuat dokumen HTML dan melakukan formatting terhadap dokumen tersebut
- d. Mahasiswa mampu melakukan text formatting
- e. Mahasiswa mampu melakukan listing pada dokumen HTML
- f. Mahasiswa mampu membuat listing bersarang (nested) pada dokumen HTML

KEBUTUHAN

- a. Unit Komputer (Memiliki Sistem Operasi Windows, Linux, dll)
- b. Text editor (Notepad, Sublime Text, dll)
- c. Web server (dalam modul ini digunakan Apache dari modul XAMPP)
- d. Web browser (Mozilla, Chrome, dll)

SEKILAS TENTANG HTML

HTML adalah singkatan dari Hyper Text Markup Language, yang merupakan suatu bahasa yang digunakan mengembangkan halaman web. HTML dibuat dan diperkenalkan oleh Berners-Lee pada akhir 1991 tetapi spesifikasi HTML standar pertama yang diterbitkan pada 1995 adalah "HTML 2.0". HTML dibuat dengan tujuan agar pengguna dapat bekerja sama dengan menggabungkan pengetahuan mereka dalam suatu dokumen hypertext melalui laman Web. Kemudian pada akhir 1999, diterbitkan HTML 4.01 yang menjadi versi utama dari HTML. Meskipun versi HTML 4.01 banyak digunakan, tetapi saat ini telah diperkenalkan versi

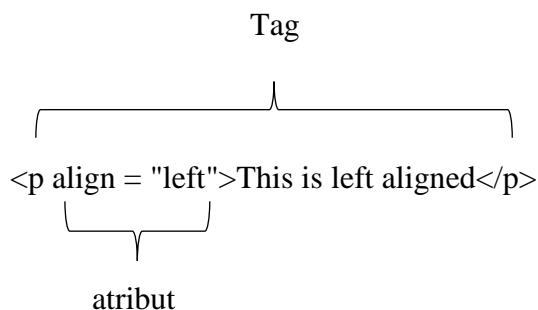
HTML-5 yang merupakan perluasan ke HTML 4.01 yang diterbitkan pada tahun 2012.

MEMULAI MENULIS HTML

1. Format penulisan HTML

Secara umum, penulisan HTML dibagi menjadi tiga bagian yaitu bagian deklarasi tipe dokumen, bagian kepala dokumen, dan bagian badan dokumen. Bagian-bagian ini dibuat dengan menambahkan komponen berupa tag (markup) serta elemen standar yang menunjukkan isi dari bagian dokumen HTML.

Komponen tersebut terdiri dari *start tag* yang berfungsi sebagai awal deklarasi dari isi dokumen HTML, kemudian diakhiri dengan *end tag* sebagai tanda akhir dari bagian dokument tersebut. Misalnya <head> ... </head>, <h1> ... </h1>, dan tag-tag lainnya. Kemudian, setiap tag memungkinkan memiliki atribut didalamnya tergantung kebutuhan dari isi dokumen yang akan kita buat. Misalnya:



Berikut contoh sederhana penulisan teks HTML:

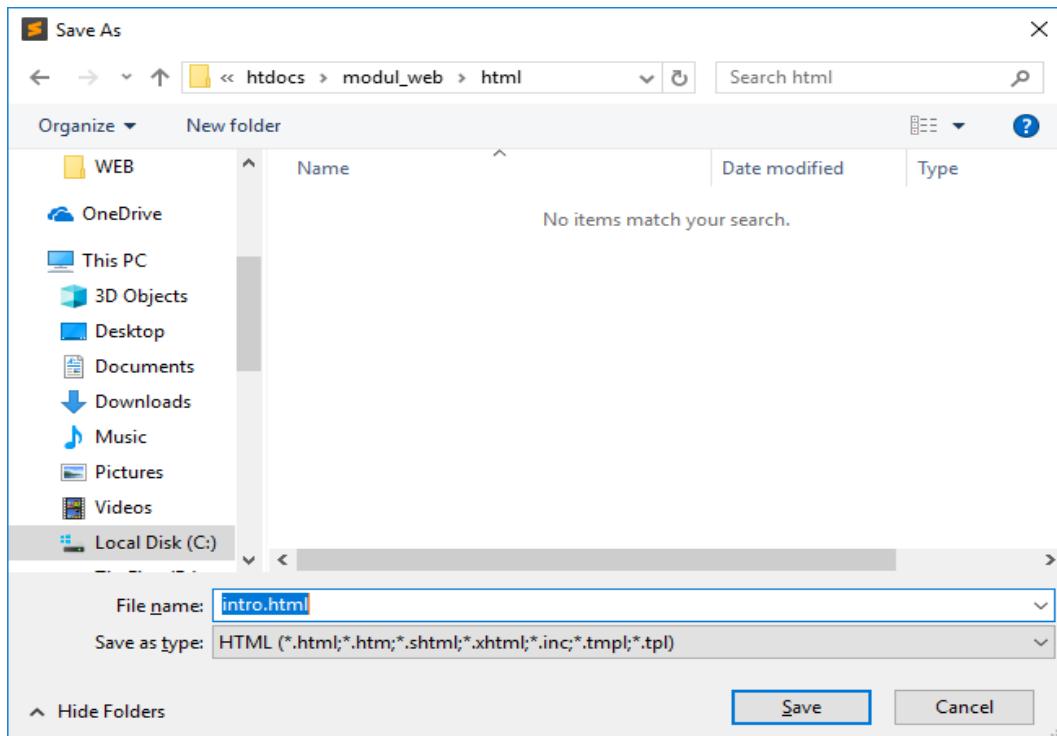
```
<!DOCTYPE html>
<html>

    <head>
        <title>Judul dokumen</title>
    </head>

    <body>
        <p>Membuat halaman HTML</p>
    </body>

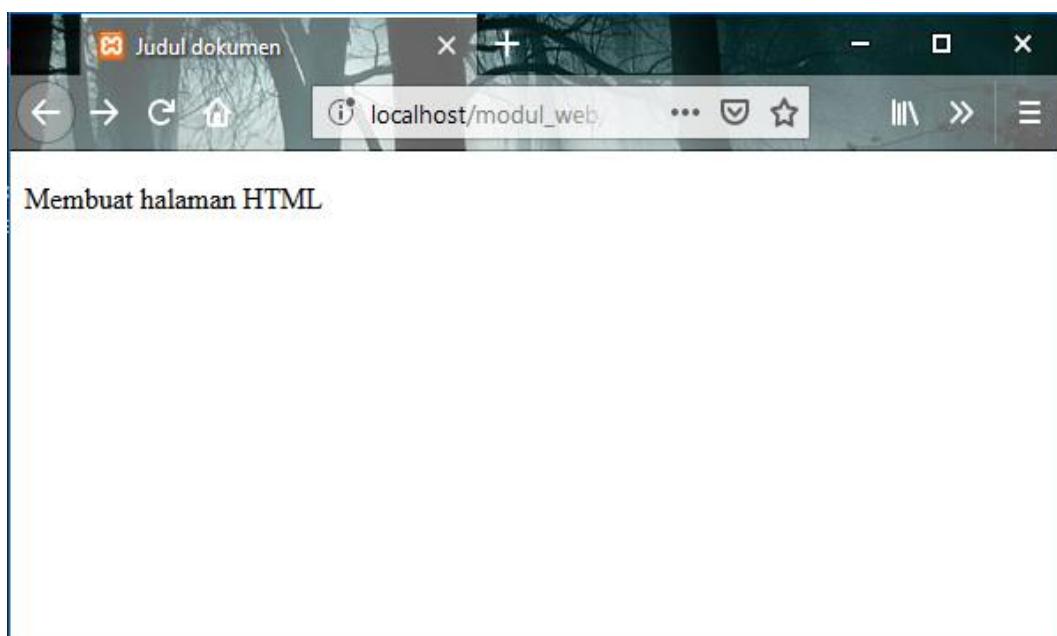
</html>
```

Simpan file tersebut (lihat gambar) dengan ekstensi file *.html yang menunjukkan bahwa file tersebut merupakan dokumen HTML. Perhatikan gambar 2.1:



Gambar 2.1: Menyimpan file HTML

Kemudian, jika file dibuka melalui web browser tampilannya akan seperti gambar 2.2 berikut:



Gambar 2.2: Tampilan halaman Web HTML

2. Tag dasar HTML

Untuk menulis dokumen HTML diperlukan deklarasi tag dari setiap bagian yang akan dibuat. Biasanya tag ini disebut juga sebagai element pembangun dasar dari suatu halaman web baik itu web statis maupun dinamis. Pada tabel berikut disajikan beberapa contoh tag yang sering digunakan dalam membuat dokumen HTML pada halaman web.

Start	Content	End Tag
<html>	Deklarasi halaman HTML	</html>
<h1>	Menuliskan heading jenis kesatu	</h1>
<title>	Memberi judul dokumen	</title>
<body>	Bagian badan dokumen	</body>
<div>	Blok untuk membedakan kelompok elemen tertentu	</div>
	Membedakan kelompok elemen dalam satu baris	
<class>	Membuat kelas	</class>
<id>	Identitas suatu blok	</id>
<style>	Memberikan ragam tampilan	</style>
<p>	Membuat paragraph	</p>
 	Membuat garis baru	</br>
<hr>	Membuat garis horizontal	</hr>
	Menyisipkan gambar	
<table>	Membuat tabel	</table>
<tr>	Membuat baris pada table	</tr>
<td>	Membuat kolom data pada table	</td>
<th>	Membuat baris kolom horizontal pada table	</th>
	Membuat tulisan tebal	
<i>	Membuat tulisan miring	</i>
<u>	Membuat tulisan bergaris bawah	</u>
^{	Membuat superscript (tulisan menjorok ke atas)	}
_{	Membuat superscript (tulisan menjorok ke atas)	}
<big>	Membuat tulisan ukuran besar	</big>
<small>	Membuat tulisan ukuran kecil	</small>
	Memberikan penomoran symbol (Unorder list)	
	Nomor item (List item)	
	Memberikan penomoran angka (Order List)	
<button>	Membuat tombol	</button>

a. Heading

Heading biasa digunakan untuk membuat judul suatu teks pada dokumen HML. Heading memiliki enam ukuran yaitu <h1>, <h2>, <h3>, <h4>, <h5>, dan <h6>. Saat menampilkan heading, web browser akan menambahkan satu baris sebelum dan satu baris setelah heading tersebut.

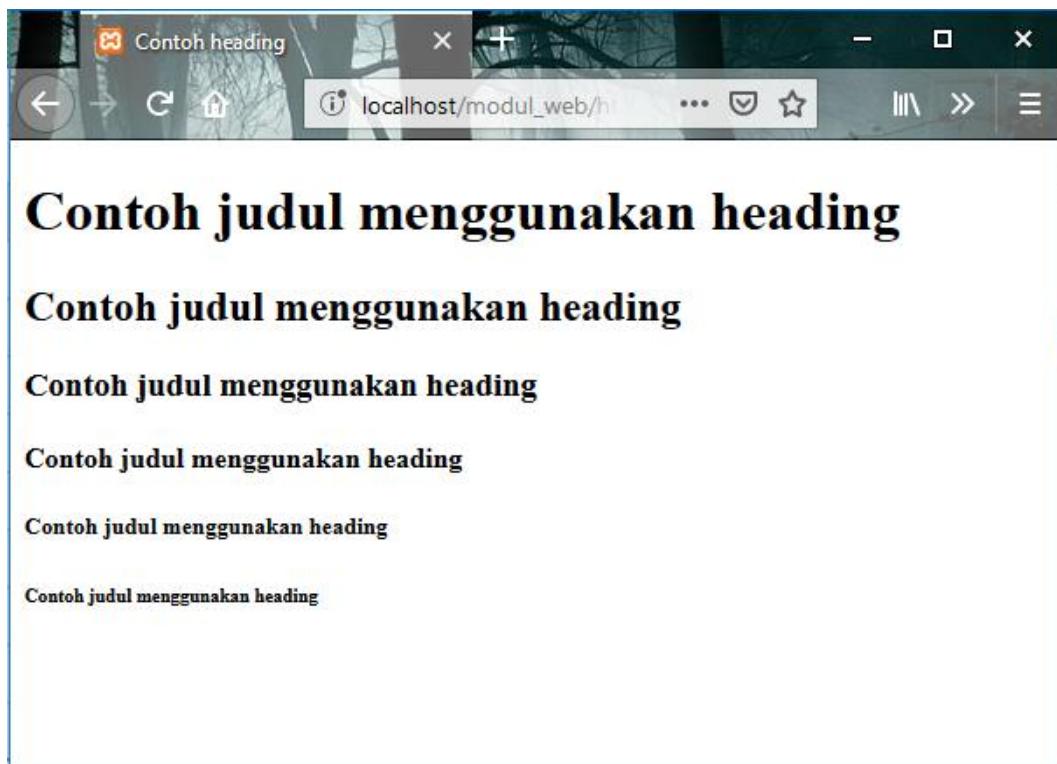
```
<!DOCTYPE html>
<html>

    <head>
        <title>Contoh heading</title>
    </head>

    <body>
        <h3>Contoh judul menggunakan heading</h3>
    </body>

</html>
```

Tampilan heading pada web browser ditunjukkan pada gambar2.3 berikut.



Gambar 2.3: Tampilan heading pada halaman Web

b. Paragraf

Paragraf merupakan salah satu bagian penting dalam suatu dokumen HTML. Biasanya paragraf digunakan untuk membuat artikel-artikel berita atau buku yang isinya menjelaskan suatu topik/ judul tertentu. Pada dokumen HTML, paragraf ditulis menggunakan tag `<p> ... </p>`. Berikut contoh penulisan tag untuk paragraf:

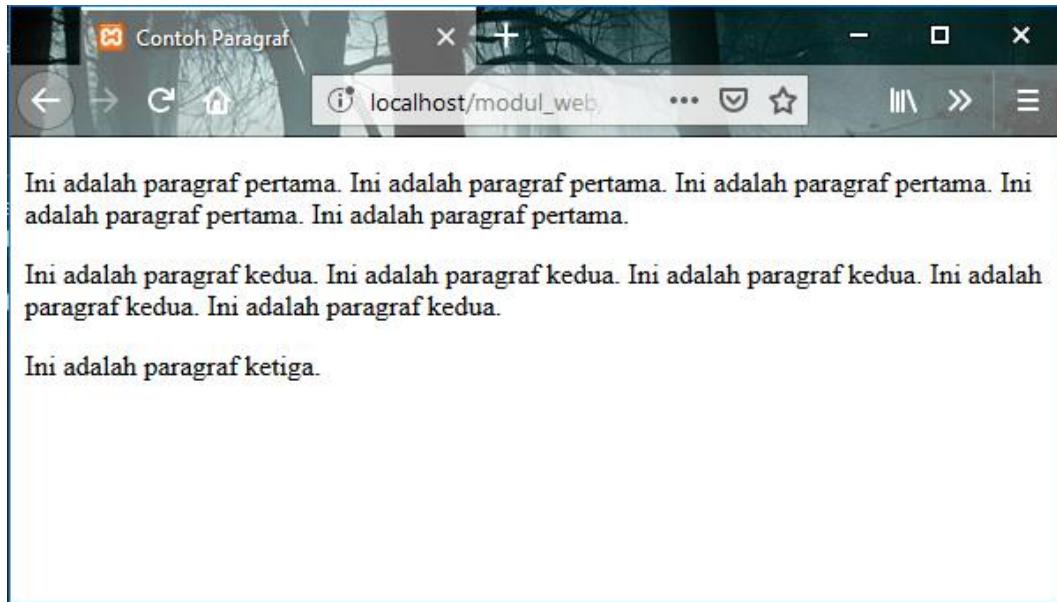
```
<!DOCTYPE html>
<html>

    <head>
        <title>Contoh Paragraf</title>
    </head>

    <body>
        <p>Ini adalah paragraf pertama.</p>
        <p> Ini adalah paragraf kedua.</p>
        <p> Ini adalah paragraf ketiga.p>
    </body>

</html>
```

Jika ditampilkan pada web browser, maka paragraf akan terlihat seperti pada gambar 2.4 berikut:



Gambar 2.4: Tampilan paragraf pada halaman Web

c. Garis baru

Pada dokumen HTML, garis baru dibuat menggunakan tag
. Terdapat perbedaan antara halaman yang menggunakan tag dengan dan tanpa penutup tag. Contoh penulisan teks disertai dengan tag
 yaitu:

```
<!DOCTYPE html>
<html>

    <head>
        <title>Contoh Tag Break</title>
    </head>

    <body>
        <p>Ini adalah kalimat pertama. Ini adalah kalimat kedua.  

        Ini adalah kalimat ketiga. Ini adalah kalimat keempat. Ini  

        adalah kalimat kelima.  

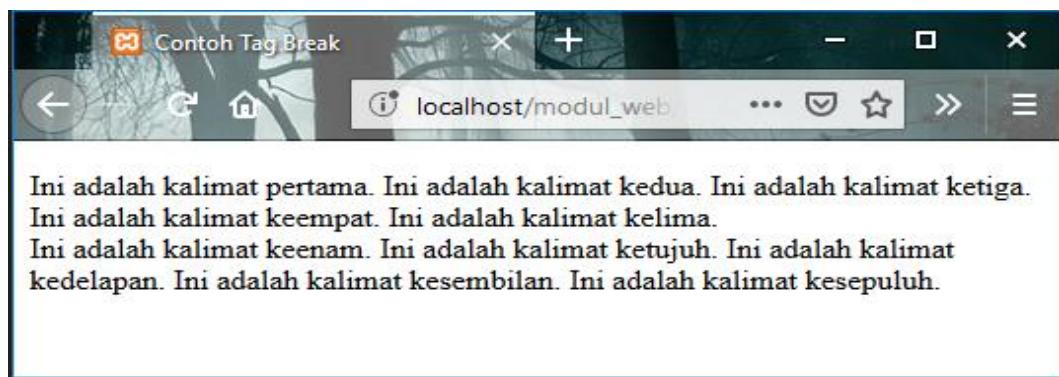
        <br> Ini adalah kalimat keenam. Ini adalah kalimat ketujuh.  

        Ini adalah kalimat kedelapan. Ini adalah kalimat  

        kesembilan. Ini adalah kalimat kesepuluh.</p>
    </body>

</html>
```

Jika dibuka pada halaman web, maka akan tampil seperti pada gambar 2.5 berikut:



Gambar 2.5: Tampilan garis baru (break) dengan dan tanpa penutup tag

d. Garis horizontal

Garis horizontal merupakan garis yang digunakan untuk memisahkan dua paragraf atau untuk memisahkan konten dalam halaman web. Untuk membuat garis horizontal gunakan tag <hr> seperti contoh berikut:

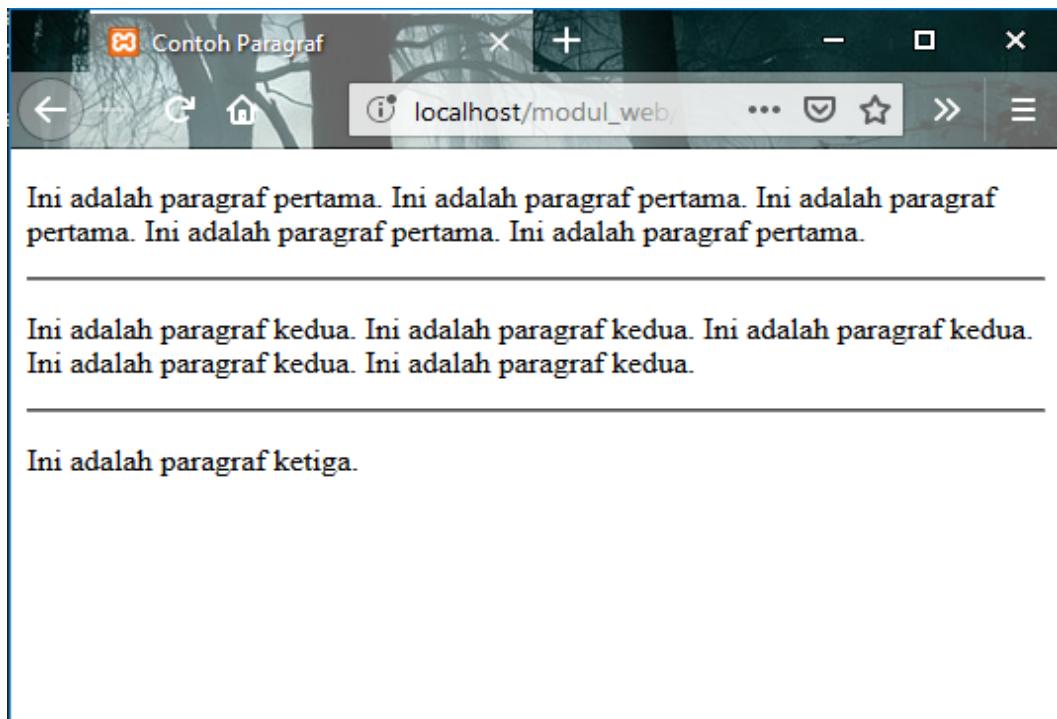
```
<!DOCTYPE html>
<html>

    <head>
        <title>Contoh Paragraf</title>
    </head>

    <body>
        <p>Ini adalah paragraf pertama. Ini adalah paragraf pertama.
        Ini adalah paragraf pertama. Ini adalah paragraf pertama.
        Ini adalah paragraf pertama.</p>
        <hr>
        <p> Ini adalah paragraf kedua. Ini adalah paragraf kedua.
        Ini adalah paragraf kedua. Ini adalah paragraf kedua. Ini
        adalah paragraf kedua.</p>
        <hr>
        <p> Ini adalah paragraf ketiga.</p>
    </body>

</html>
```

Jika dibuka dengan web browser maka tampilannya seperti pada gambar 2.6:



Gambar 2.6: Memisahkan paragraf dengan garis horizontal

3. Atribut generik HTML dan XML

Terdapat atribut generik (umum) yang digunakan oleh dokumen HTML yang kemudian diadopsi oleh dokumen XML. Atribut tersebut diantaranya:

Attribute	Options	Function
align	right, left, center	Tag yang mengatur posisi secara horizontal
valign	top, middle, bottom	Tag yang mengatur posisi secara vertikal
bgcolor	numeric, hexidecimal, RGB values	Menempatkan warna di belakang elemen
background	URL	Menempatkan gambar background di belakang elemen
id	User Defined	Menamai elemen untuk dipadukan dengan CSS
class	User Defined	Mengklasifikasikan elemen untuk dipadukan dengan CSS
width	Numeric Value	Menentukan lebar tabel, gambar, atau sel tabel
height	Numeric Value	Menentukan tinggi tabel, gambar, atau sel tabel
title	User Defined	Judul dari suatu elemen

4. Halaman Web pertama

Berikut ini akan disajikan contoh skrip dokumen HTML yang memuat beberapa tag dasar yang sering digunakan pada halaman Web.

```
<!DOCTYPE html>
<html>
<head>
    <title>Artikel Homepage</title>
</head>

<body>

    <h1 id="top">Halaman Web HTML</h1>
    <p>Membuat situs web merupakan salah satu pekerjaan yang cukup populer dan menjanjikan saat ini. Syarat minimal untuk
```

membuat situs web adalah kita memahami dasar-dasar HTML. Selanjutnya, HTML bisa dipadukan dengan script PHP, JavaScript, dan script lainnya yang mendukung pemrograman sisi server.</p>

Berikut disajikan beberapa link contoh membuat halaman web dengan HTML:


```
<ul>
<li><a href="break.html">Break</a></li>
<li><a href="garis.html">Horizontal Line</a> </li>
<li><a href="paragraf.html">Paragraf</a></li>
</ul>
```

<hr>

<p align="center">Selanjutnya, kita dapat membuat posisi teks/ paragraf sesuai dengan layout yang kita inginkan. Bisa berada di posisi tengah seperti paragraf ini.</p>

<p align="center">Posisi Tengah</p>

<hr>

<p align="left">Selanjutnya, kita dapat membuat posisi teks/ paragraf sesuai dengan layout yang kita inginkan. Bisa berada di posisi tengah seperti paragraf ini.</p>

<p align="center">Posisi Kiri</p>

<hr>

<p align="right">Selanjutnya, kita dapat membuat posisi teks/ paragraf sesuai dengan layout yang kita inginkan. Bisa berada di posisi tengah seperti paragraf ini.</p>

<p align="center">Posisi Kanan</p>

<hr>

<p><i>Selanjutnya, kita dapat membuat posisi teks/ paragraf sesuai dengan layout yang kita inginkan. Bisa berada di posisi sebelah kanan seperti paragraf ini.</i></p>

<p align="center">Posisi Teks Miring</p>

<hr>

<p><u>Selanjutnya, kita dapat membuat posisi teks/ paragraf sesuai dengan layout yang kita inginkan. Bisa berada di posisi sebelah kanan seperti paragraf ini.</u></p>

<p align="center">Posisi Teks Bergaris Bawah</p>

<hr>

<h2 align="center">Link Kembali Top</h2>

</body>

</html>

Halaman Web HTML

Membuat situs web merupakan salah satu pekerjaan yang cukup populer dan menjanjikan saat ini. Syarat minimal untuk membuat situs web adalah kita memahami dasar-dasar HTML. Selanjutnya, HTML bisa dipadukan dengan script PHP, JavaScript, dan script lainnya yang mendukung pemrograman sisi server.

Berikut disajikan beberapa link contoh membuat halaman web dengan HTML:

- [Break](#)
 - [Horizontal Line](#)
 - [Paragraf](#)
-

Selanjutnya, kita dapat membuat posisi teks/ paragraf sesuai dengan layout yang kita inginkan. Bisa berada di posisi tengah seperti paragraf ini.

Posisi Tengah

Selanjutnya, kita dapat membuat posisi teks/ paragraf sesuai dengan layout yang kita inginkan. Bisa berada di posisi tengah seperti paragraf ini.

Posisi Kiri

Selanjutnya, kita dapat membuat posisi teks/ paragraf sesuai dengan layout yang kita inginkan. Bisa berada di posisi sebelah kanan seperti paragraf ini.

Posisi Kanan

Selanjutnya, kita dapat membuat posisi teks/ paragraf sesuai dengan layout yang kita inginkan. Bisa berada di posisi sebelah kanan seperti paragraf ini.

Posisi Teks Miring

Selanjutnya, kita dapat membuat posisi teks/ paragraf sesuai dengan layout yang kita inginkan. Bisa berada di posisi sebelah kanan seperti paragraf ini.

Posisi Teks Bergaris Bawah

[Link Kembali Top](#)

Gambar 2.7: Tampilan Halaman Web sederhana

TUGAS PRAKTIKUM 2

Buatlah contoh halaman web menggunakan HTML dengan tag-tag dasar yang telah disampaikan. Minimal 12 (dua belas) macam tag yang ada pada HTML.

MODUL III

HTML LANJUT (LINK, FRAME, DAN TABEL)

TUJUAN

- a. Mahasiswa mampu memahami dan bisa membuat link dalam HTML
- b. Mahasiswa mampu memahami dan bisa membuat frame dalam HTML
- c. Mahasiswa mampu memahami dan bisa membuat table dalam HTML

KEBUTUHAN

- a. Unit Komputer (Memiliki Sistem Operasi Windows, Linux, dll)
- b. Text editor (Notepad, Sublime Text, dll)
- c. Web server (dalam modul ini digunakan Apache dari modul XAMPP)
- d. Web browser (Mozilla, Chrome, dll)

RINGKASAN

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. Perhatikan bahwa tag untuk link memang hanya 1, tetapi memiliki beberapa atribut tambahan agar link tersebut memberikan fungsi yang berbeda. Untuk keperluan modifikasi tampilan agar link tidak hanya terlihat sebagai link standar, nantinya akan disampaikan dalam materi khusus CSS. Tampilan link juga dapat diubah sedikit lewat modifikasi tag `<body>` dalam suatu file HTML.

Start Tag	Kegunaan
<code><a></code>	Mendefinisikan sebuah anchor

Atribut Target	Kegunaan
target="_blank"	Memuat dokumen baru ke dalam window baru yang kosong
target="_self"	Memuat dokumen baru ke dalam window yang sama dengan anchor (default)
target="_parent"	Memuat dokumen baru ke dalam parent frame (bila menggunakan frame)
target="_top"	Memuat dokumen baru ke dalam keseluruhan window browser (cara yang umum digunakan untuk keluar dari frame)

Sebuah halaman web yang ditampilkan dalam suatu window browser dapat dipecah dalam beberapa tampilan yang berbeda. Masing-masing bagian tampilan dapat berisi sebuah tampilan dari file HTML, sehingga dalam satu window browser dapat dibuka beberapa file HTML sekaligus. Untuk dapat menampilkan beberapa file HTML dalam satu window browser, dibutuhkan tag untuk membuat window menjadi beberapa bingkai (frame). Di bawah ini adalah tag untuk membentuk frame pada window browser.

Start Tag	Kegunaan
<frameset>	Mendefinisikan sebuah himpunan frame
<frame>	Mendefinisikan sebuah sub window (sebuah frame)
<noframes>	Mendefinisikan sebuah bagian noframe untuk browser yang tidak dapat menangani frame
<iframe>	Mendefinisikan sebuah inline sub window (frame)

Dalam suatu halaman web, pemformatan saja tidaklah cukup untuk membentuk tampilan web yang rapi dan enak dipandang. Supaya tampilannya lebih terstruktur, Anda dapat menggunakan tabel. Perhatikan tag-tag di bawah ini.

Start Tag	Kegunaan
<table>	Mendefinisikan sebuah tabel
<th>	Mendefinisikan sebuah header tabel

<tr>	Mendefinisikan suatu barisan dalam tabel
<td>	Mendefinisikan suatu sel dalam table
<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

LATIHAN

Silakan mencoba kode-kode contoh pada modul ini, kemudian silakan mencoba variasi penggunaannya. Gunakan alamat tujuan yang berbeda dari contoh yang ada. Bila tidak terdapat koneksi ke Internet, gunakan alamat lokal saja terlebih dahulu. Alamat lokal juga mempercepat akses Anda, sehingga hasilnya lebih cepat tampil di layar monitor. Bila Anda ingin menggunakan halaman tujuan secara lokal, sebaiknya Anda membuat file HTML tambahan dengan nama file yang disesuaikan agar dapat dipanggil. Isi file HTML tambahan terserah Anda, yang penting bisa digunakan untuk menunjukkan bahwa tag Anda bekerja seperti seharusnya.

Untuk mencoba kode HTML tentang frame, silakan menyiapkan file-file yang diperlukan sesuai dengan nama-nama file yang telah disebutkan dalam contoh kode dalam modul ini. Simpan di tempat yang sesuai dengan penunjukan frame Anda, disarankan disimpan dalam folder yang sama tempat Anda menyimpan file definisi frame. Sebagai awalan, sebaiknya Anda menggunakan browser IE, sebelum mencobakannya ke Navigator atau Opera (bila ada).

Latihan menggunakan tabel mengharuskan Anda mencoba bentuk-bentuk yang ada dalam contoh, kemudian silakan mencoba-coba sendiri variasi contoh dengan menambahkan atau mengurangi sel dan isinya. Variasikan juga parameternya, agar Anda tahu setiap bagian penentu tampilan tabel.

1. Link biasa, link gambar, link ke window baru, dan link internal

```
<!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/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />
<title>Berita Bola</title>
<style type="text/css">
<!--
.style1 {color: #FF0000}
-->
</style>
</head>

<body bgcolor="#CCCCCC">
<h1 align="justify" class="style1">Fulham Redam
Hamburg Tanpa Gol</h1>
<p align="justify">Hamburg gagal memetik kemenangan
di semi-final <em>leg</em> pertama Liga Europa. Tampil
di hadapan pendukungnya sendiri di HSH Nordbank Arena,
Jumat (23/4) dinihari, Hamburg harus puas bermain
imbang tanpa gol melawan Fulham. <br />
<br />
Tim tuan rumah sebenarnya berusaha tampil agresif sejak
awal pertandingan. Di dua menit awal, penyelesaian
yang buruk dari Ruud van Nistelrooy membuat gawang
Fulham yang dikawal Mark Schwarzer masih bisa selamat.
<br />
<br />
Beberapa menit berselang, giliran gelandang Piort
Trochowski yang juga gagal memaksimalkan kesempatan.
Bola hasil tendangan pemain tim nasional Jerman ini
masih terlalu melebar dari gawang Fulham. <br />
<br />
Sementara <em>The Cottagers</em> mendapatkan
kesempatan pertamanya untuk mencetak gol setelah waktu
berjalan hampir 15 menit. Ketika itu umpan panjang
Bobby Zamora juga gagal diselesaikan dengan baik oleh
Zoltan Gea. <br />
<br />
Lalu di lima menit jelang turun minum gelandang Fulham
Joanthan Pitroipa membuka harapan publik tuan rumah.
Pergerakannya yang cepat ke lini pertahanan Fulham itu
kemudian diselesaikan dengan sebuah tendangan. Tapi
```

ancaman tersebut masih bisa diantisipasi oleh Schwarzer dengan baik.

Di babak kedua, Hamburg kembali lagi berusaha untuk tampil lebih dominan dalam menyerang. Kerjasama Paolo Guerrero dan Nistelrooy membuka kesempatan pertama bagi Fulham. Tapi ruang gerak yang terbatas membuat tendangan Guerrero masih melambung di atas sasaran.

Sebaliknya Fulham juga tidak tinggal diam untuk mencuri kesempatan. Salah satu peluang yang tercipta di babak ini melalui pergerakan Gera yang bergerak dari sayap kiri.

Gelandang asal Hungaria ini kemudian melepaskan umpannya kepada Clint Dempsey. Tapi bek Hamburg Joris Mathijsen masih bisa mengantisipasi ancaman tersebut dengan melakukan *sliding* yang bersih kepada Dempsey.

Sementara kesempatan paling besar di babak ini terjadi di menit ke-70. Nistelrooy memberikan umpan ke arah Pitroipa. Gelandang asal Burkina Faso ini kemudian menyambutnya dengan tandukan kepala. Sayang, bola yang mengarah ke pojok atas gawang Fulham masih di tepis oleh Schwarzer.

Dengan hasil imbang ini membuat langkah Fulham untuk menembus ke partai puncak Liga Europa kian terbuka ketika akan bermain di hadapan publiknya sendiri pekan depan.</p>

<p align="justify">Lihat Foto Gambar

</p>

</body>

</html>

2. Link ke Gambar

```
<!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/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />
<title>Untitled Document</title>
</head>
```

```
<body bgcolor="#CCCCCC">
<p>Ini Berita Fotonya :</p>
<p></p>
<p>Sumber : www.goal.com</p>
<p><a href="link.html">Back to homepage</a> </p>
</body>
</html>
```

3. Membuat tabel

```
<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>
```

4. Merger Cell

```
<html>
<head>
<title>Using Table</title>
</head>

<body>
<table bgcolor="CCCCFF" width="62%" border="1"
cellpadding="0">
<tr>
<td colspan="3" align="center">Quarter 1</td>
<td colspan="3" align="center">Quarter 2</td>
</tr>
<tr align="center">
<td>Jan</td>
<td>Feb</td>
<td>Mar</td>
<td>Apr</td>
<td>May</td>
<td>Jun</td>
</tr>
<tr>
<td>100</td>
<td>5000</td>
<td>200</td>
<td>1500</td>
<td>2500</td>
<td>1750</td>
</tr>
<tr>
<td>290</td>
```

```
<td>5050</td>
<td>2300</td>
<td>100</td>
<td>270</td>
<td>300</td>
</tr>
</table>
</body>
</html>
```

5. RowSpan

```
<html>
<head>
<title>Rowspan</title>

</head>

<body>
<table bgcolor="lavender" width="75%" border="1"
cellpadding="0">
<tr>
<td></td>
<td></td>
<td>South</td>
<td>North</td>
</tr>
<tr>
<td rowspan="3">Quarter 1</td>
<td>Jan</td>
<td>1000</td>
<td>12000</td>
</tr>
<tr>
<td>Feb</td>
<td>12500</td>
<td>1345</td>
</tr>
<tr>
<td>Mar</td>
<td>78090</td>
<td>71080</td>
</tr>
</table>
</body>
</html>
```

MODUL IV

FORM DAN INPUT

TUJUAN

- a. Mahasiswa mampu memahami dan bisa cara kerja form menggunakan HTML
- b. Mahasiswa mampu memahami cara kerja input output dari form HTML

KEBUTUHAN

- a. Unit Komputer (Memiliki Sistem Operasi Windows, Linux, dll)
- b. Text editor (Notepad, Sublime Text, dll)
- c. Web server (dalam modul ini digunakan Apache dari modul XAMPP)
- d. Web browser (Mozilla, Chrome, dll)

RINGKASAN

Bila Anda suatu ketika ingin memasukkan informasi ke suatu situs web, maka anda memerlukan fasilitas khusus untuk keperluan tersebut. Dalam HTML Anda dapat membuat tampilan halaman untuk menerima masukan dari pengunjung situs, kemudian menyimpannya di situs Anda atau langsung dimunculkan kembali sebagai suatu halaman. Disinilah perlunya menggunakan form. Perhatikan tag di bawah ini.

Start Tag	Kegunaan
<form>	Mendefinisikan sebuah form untuk input dari pengunjung
<input>	Mendefinisikan sebuah field input
<textarea>	Mendefinisikan sebuah text-area (sebuah kontrol masukan teks jenis multi-line)
<label>	Mendefinisikan sebuah label ke kontrol

<fieldset>	Mendefinisikan sebuah fieldset
<legend>	Mendefinisikan suatu caption untuk sebuah fieldset
<select>	Mendefinisikan sebuah list yang dapat dipilih (drop-down box)
<optgroup>	Mendefinisikan sebuah kelompok pilihan
<option>	Mendefinisikan sebuah pilihan dalam drop-down box
<button>	Mendefinisikan sebuah tombol

Anda tentu saja tidak ingin membuat halaman web yang hanya berisi teks saja. Agar halaman web Anda lebih menarik, gunakan unsur grafis berupa gambar. Anda dapat memanipulasi gambar dalam suatu halaman web sehingga menghasilkan berbagai efek. Tag di bawah ini digunakan untuk keperluan tersebut.

Start Tag	Kegunaan
	Mendefinisikan sebuah gambar
<map>	Mendefinisikan sebuah image map (peta gambar)
<area>	Mendefinisikan sebuah area di dalam suatu image map

LATIHAN

Disini Anda akan berlatih membuat form dengan segala obyeknya. Tetapi hanya sekedar form untuk mengumpulkan data secara interaktif, tanpa memproses data tersebut. Di bagian ini belum akan dibicarakan tentang pemrosesan data, oleh karena itu silakan mencoba-coba setiap obyek dalam form saja, dengan berbagai cara penempatannya yang memungkinkan.

Untuk dapat melihat gambar dalam halaman dokumen Anda, silakan mempersiapkan terlebih dahulu gambar-gambar yang akan digunakan. Untuk gambar bergerak, gunakan saja format animasi GIF. Untuk gambar diam, gunakan format sesuai yang diperlihatkan dalam kode sumber contoh.

1. Field teks dan password

```
<html>
<body>
<form>
Nama depan:
<input type="text" name="namadepan">
<br>
Nama belakang:
<input type="text" name="namabelakang">
<br>
Password:
<input type="password" name="password">
<input name="btnKirim" type="submit" id="btnKirim" value="Kirim">
<input name="btnCancel" type="reset" id="btnCancel" value="Cancel">
</form>
<p>
Perhatikan bahwa ketika Anda mengetikkan karakter-karakter dalam suatu field password, browser akan menampilkan asterisk atau bullet saja, bukannya karakter yang diketikkan.
</p>
</body>
</html>
```

2. Checkbox dan tombol radio

```
<html>
<body>
<form>
Saya memiliki sebuah sepeda:
<input type="checkbox" name="Sepeda">
<br>
Saya memiliki sebuah mobil:
<input type="checkbox" name="Mobil" value="ON">
<br>
Jenis kelamin Anda?
<br>
Pria:
<input type="radio" name="Sex" value="pria">
<br>
Wanita:
<input type="radio" name="Sex" value="wanita" checked>
```

```
</form>
<p>
Ketika seorang pengguna memberikan klik mouse pada
sebuah tombol radio, tombol tersebut terlihat diberi
tanda cek, dan semua tombol lain yang berada dalam
nama yang sama akan kehilangan tanda cek-nya
</p>
</body>
</html>
```

3. Kotak drop down dengan nilai default

```
<html>
<body>
<form>
<select name="mobil">
<option value="volvo">Volvo
<option value="peugeot">Peugeot
<option value="fiat" selected>Fiat
<option value="jaguar">Jaguar
</select>
</form>
</body>
</html>
```

4. Tombol dan fieldset

```
<html>
<body>
<form>
<p>Nama depan: &nbsp;&nbsp;&nbsp;
<input type="text" name="namadepan">
<br>
Nama belakang:
<input type="text" name="namabelakang">
<br>
Password:
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<input type="password" name="password">
</p>
<p>Jenis Kelamin :
<label>
<input name="radiobutton" type="radio"
value="radiobutton">
Laki-Laki</label>
<label>
```

```
<input name="radiobutton" type="radio"
value="radiobutton">
    Perempuan</label>
</p>
<p>Hobi :
    <p><label>
        <input name="checkbox" type="checkbox"
value="checkbox">
            Sepak Bola</label><p>
            <label>
                <input name="checkbox" type="checkbox"
value="checkbox">
                    Basket</label>
            <p>
                <label>
                    <input name="checkbox" type="checkbox"
value="checkbox">
                        Berenang</label>

<p>
    <input name="btnKirim" type="submit" id="btnKirim"
value="Kirim">
    <input name="btnCancel"
type="reset" id="btnCancel" value="Cancel">
</p>
</form>
</body>
</html>
```

MODUL V

PENGANTAR PHP (PHP HYPETEXT PREPROCESSOR)

TUJUAN

- a. Mahasiswa mengetahui dasar-dasar PHP
- b. Mahasiswa mengetahui struktur penulisan dokumen PHP
- c. Mahasiswa mampu memadukan penulisan PHP dengan HTML

KEBUTUHAN

- a. Unit Komputer (Memiliki Sistem Operasi Windows, Linux, dll)
- b. Text editor (Notepad, Sublime Text, dll)
- c. Web server (dalam modul ini digunakan Apache dari modul XAMPP)
- d. Web browser (Mozilla, Chrome, dll)

PHP (*Hypertext Preprocessor*), merupakan bahasa pemrograman web bersifat *server-side*, artinya bahasa berbentuk *script* yang disimpan dan dijalankan di komputer *server* (WebServer) sedang hasilnya yang dikirimkan ke komputer *client* (WebBrowser) dalam bentuk *script* HTML (Hypertext Mark up Language).

Karakteristik *script* PHP dapat diuraikan sebagai berikut :

1. file PHP disimpan dengan extensi filenya yaitu : *.php3, *.php4, *.php
2. *Script* PHP biasanya diawali dengan tag ‘<?’ atau ‘<?php’ dan ditutup dengan tag ‘?>’
3. File PHP dapat menginduk atau disisipkan pada bahasa script lainnya atau dapat berdiri sendiri. Contoh skrip PHP yang disisipkan pada HTML :

```
<html>
  <head>
    <title>Contoh</title>
  </head>
  <body>
    <?php
```

```
        echo "Hai, Aku adalah skrip PHP!";
    ?>
    </body>
</html>

Sedang bentuk skrip PHP yang berdiri sendiri :

<?php
    echo "Hai, Aku adalah skrip PHP!";
?>
```

NB : **echo()**, berfungsi untuk menampilkan output.

Apa yang dapat dilakukan oleh PHP?

Pada level dasar, PHP dapat melakukan semua apa yang dapat dilakukan oleh pemrograman berbasis CGI lainnya, dan juga kekuatan utamanya adalah dalam pembuatan aplikasi web database. Hampir sebagian besar produk software DBMS dapat didukung oleh PHP baik yang berjalan pada sistem operasi Windows, Linux maupun sistem operasi lainnya. Sebagian DBMS yang dapat didukung oleh PHP, di antaranya adalah sebagai berikut :

Adabas	D Ingres	Oracle (OCI7 and OCI8)
DBase	InterBase	Ovrimos
Empress	FrontBase	PostgreSQL
FilePro (read-only)	mSQL	Solid
Hyperwave	Direct MS-SQL	Sybase
IBM DB2	MySQL	Velocis
Informix	ODBC	Unix dbm

Pada dasar-dasar PHP ini akan dijabarkan tentang penulisan komentar, tipe data, konstanta, variabel dan operator.

Komentar

Komentar adalah bagian dari program yang berfungsi sebagai penjelas atau pemberi keterangan dalam program. Komentar ini tidak akan dieksekusi/dikerjakan oleh interpreter.

Untuk mendefinisikan komentar dipergunakan simbol-simbol karakter berikut :

1. Dengan symbol **dobel-slash(//)**, biasanya untuk komentar satu baris

Contoh :

```
<?php
    // nama program : komentar1.php
    // dibuat tanggal : 3 Juni 2004
    echo "Contoh Komentar dengan '//' ";
?>
```

2. Diawali dengan symbol **slash-asterik (*)** dan ditutup dengan **asterik-slash(*)**, biasanya digunakan untuk memberikan komentar lebih dari satu baris.

Contoh :

```
<?php
/* nama program : komentar2.php dibuat tanggal
 : 3 Juni 2004
 */
echo "Contoh Komentar dengan '/*' dan '*/'
?>
```

baris komentar tidak ditampilkan di halaman webbrowser karena komentar akan diabaikan oleh interpreter.

Tipe Data

Tipe data dasar PHP terdiri dari

- **integer**, termasuk jenis data bilangan bulat
- **double**, termasuk jenis data bilangan pecahan/desimal
- **string**, termasuk jenis data teks/untiaian karakter

Contoh :

```
<?php
$a=10; //variable $a memiliki tipe data integer
echo $a;
$b=22.33; //variable $b memiliki tipe data double
echo $b;
```

```
$c="Skrip PHP"; //variable $c memiliki tipe data  
string  
echo $c;  
?>
```

Variabel

Variabel adalah suatu pengenal dalam program yang berfungsi untuk menyimpan nilai secara sementara dan dapat diubah-ubah nilai. Untuk mendefinisikan variable, diawali dengan simbol karakter **dollar** ('\$') dan diikuti oleh **nama pengenal**.

```
$NamaPengenal = nilai;
```

Adapun aturan dalam menyusun pengenal :

1. tersusun dari karakter huruf, angka dan underscore(_)
2. tidak boleh mengandung spasi
3. karakter pertama nama pengenal harus dari karakter huruf atau underscore.
4. huruf kecil dan besar dibedakan

Dalam PHP, tidak diperlukan pendeklarasian variabel dengan tipe datanya seperti bahasa pemrograman pascal. Setiap variable yang terbentuk dalam program dianggap bertipe variant, dengan kata lain dapat menampung tipe data dengan jenis apapun.

Contoh :

```
<?php  
$info=10; //variable $info menampung bilangan bulat  
echo $info;  
$info=22.33; //variable $info menampung bilangan pecahan  
echo $info;  
$info="Skrip PHP"; //variable $info menampung data teks/string  
echo $info;  
?>
```

Konstanta

Konstanta adalah suatu tetapan nilai dalam program. Konstanta tidak dapat dirubah nilai sewaktu program dijalankan, kalau hal itu dilakukan akan menyebabkan error.

Untuk mendefinisikan konstanta digunakan :

```
define (NamaPengenal, nilai_konstanta);
```

Contoh :

```
<?php
    // konstanta Judul="Hitung Luas Lingkaran"
    define ("Judul", "Hitung Luas Lingkaran");

    // konstanta PHI=3.14
    define ("PHI", 3.14);

    echo Judul;
    $r=10;
    echo "<BR>Jari-jari : $r<BR>\n";

    $luas=PHI * $r * $r;
    echo "Luas Lingkaran = $luas";
?>
```

Operator

Operator adalah suatu symbol yang berfungsi untuk menyusun sebuah ekspresi maupun operasi. Sedangkan yang dioperasikan operator disebut dengan operand. Adapun macam-macam operator yaitu :

1. Operator Aritmetika

Merupakan symbol-simbol operator untuk melakukan operasi matematis.

Operator	Fungsi	Prioritas
+	Penjumlahan	Ketiga
-	Pengurangan	Ketiga
*	Perkalian	Kedua
/	Pembagian	Kedua
%	Sisa Pembagian	Kedua
++	Penaikan	Pertama
--	Penurunan	Pertama

Contoh :

```

<?php
    $bill1 = 200;
    $bil2 = 33;

    $hasil = $bill1 + $bil2;
    echo "$bill1 + $bil2 = $hasil<BR>\n";

    $hasil = $bill1 - $bil2;
    echo "$bill1 - $bil2 = $hasil<BR>\n";

    $hasil = $bill1 * $bil2;
    echo "$bill1 * $bil2 = $hasil<BR>\n";

    $hasil = $bill1 / $bil2;
    echo "$bill1 / $bil2 = $hasil<BR>\n";

    $hasil = $bill1 % $bil2;
    echo "$bill1 % $bil2 = $hasil<BR>\n";

    $hasil = $bill1++;
    echo "$bill1++ = $hasil<BR>\n";

    $hasil = $bil2--;
    echo "$bil2-- = $hasil<BR>\n";
?>

```

2. Operator Pembandingan

Merupakan simbol-simbol operator untuk melakukan pembandingan antara dua buah operand. Hasil pembandingan bernilai **satu(1)** jika benar dan bernilai **nol(0)** jika salah.

Operator	Fungsi
<	Lebih kecil
>	Lebih besar
<=	Lebih kecil atau sama dengan
>=	Lebih besar atau sama dengan
==	Sama dengan
!=	Tidak sama dengan

Contoh :

```
<?php
    $bil1 = 100;
    $bil2 = 20;
    $teks1 = "PHP";
    $teks2 = "php";
    printf("%d == %d adalah %d<BR>\n", $bil1, $bil2, $bil1
    == $bil2);
    printf("%d != %d adalah %d<BR>\n", $bil1, $bil2, $bil1
    != $bil2);
    printf("%d >= %d adalah %d<BR>\n", $bil1, $bil2, $bil1
    >= $bil2);
    printf("%s == %s adalah %d<BR>\n", $teks1, $teks2,
    $teks1 == $teks2);
    printf("%s != %s adalah %d<BR>\n", $teks1, $teks2,
    $teks1 != $teks2);
?>
```

3. Operator Logika

Merupakan symbol-simbol operator untuk menyusun kalimat ekspresi/ungkapan logika. Hasil operasi ini akan didapatkan nilai **satu**(1) jika bernilai benar atau **nol**(0) jika bernilai salah.

Operator	Fungsi
AND atau &&	Operasi logika and
OR atau 	Operasi logika or
XOR	Operasi logika eksklusif or
!	Ingkaran/negasi

Untuk lebih jelasnya mengenai penggunaan operator-operator di atas, perhatikan table kebenaran sebagai berikut :

\$p	\$q	\$p and \$q	\$p or \$q	\$p xor \$q	! (\$p and \$q)
1	1	1	1	0	0

1	0	0	1	1	1
0	1	0	1	1	1
0	0	0	0	0	1

Contoh :

```
<?php
    $bil1 = 100;
    $bil2 = 20;
    $teks1 = "PHP";
    $teks2 = "php";
    $hasil = ($bil1 <> $bil2) or ($teks1 == $teks2);
    printf("(%d <> %d) or (%s == %s) adalah %d<BR>\n",
    $bil1, $bil2, $teks1, $teks2, $hasil);
    $hasil = ! ($teks1 == $teks2);
    printf("! (%s == %s) adalah %d<BR>\n", $teks1,
    $teks2, $hasil);
?>
```

4. Operator String

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

Contoh :

```
<?php
    $teks1 = "Aku Sedang Belajar";
    $teks2 = "Pemrograman WEB";
    $teks3 = "PHP 4";
    $hasil = $teks1 . $teks2 . $teks3;
    printf("hasil : %s<BR>\n", $hasil);
    $hasil = $teks1 . " " . $teks2 . " " . $teks3;
    printf("hasil : %s<BR>\n", $hasil);
?>
```

Praktik :

1. Ketikkan contoh-contoh program di atas dengan **notepad.exe**, dan jalankan.
2. Amati dan analisa hasil tampilan(*output*) setiap program di atas.
3. Berikan kesimpulan mengenai program-program di atas.

Tugas :

Buatlah program untuk menghitung sisa pembagian antara **angka1** dengan **angka2**.

MODUL VI

PERNYATAAN SELEKSI DAN PERULANGAN

TUJUAN

- a. Mahasiswa memahami proses seleksi dalam PHP
- b. Mahasiswa memahami proses perulangan dalam PHP
- c. Mahasiswa mampu membuat program seleksi dan perulangan menggunakan PHP

KEBUTUHAN

- a. Unit Komputer (Memiliki Sistem Operasi Windows, Linux, dll)
- b. Text editor (Notepad, Sublime Text, dll)
- c. Web server (dalam modul ini digunakan Apache dari modul XAMPP)
- d. Web browser (Mozilla, Chrome, dll)

Apa yang dimaksud pernyataan seleksi?

Sebagian besar bahasa pemrograman mengandung pernyataan seleksi. Pada dasarnya pernyataan seleksi adalah suatu mekanisme suatu pernyataan akan dikerjakan atau tidak, hal tergantung pada kondisi yang dirumuskan. Dalam PHP atau bahasa pemrograman lainnya pernyataan seleksi diterapkan dengan menggunakan statemen **if**.

Bentuk umum pernyataan **if**, adalah sbb:

```
If (kondisi) {  
    statement;  
}
```

Prinsip Kerja : *Statement* di atas akan dikerjakan apabila kondisi bernilai **TRUE** (1) sedangkan jika kondisi bernilai **FALSE** (0), maka *statement* di atas tidak akan dikerjakan.

Contoh : Mengabsolutkan nilai bulat

```
<?php
    $a = -10;
    if ($a < 0) {
        $a = $a * -1;
    }
    echo $a;
?>
```

else

sebenarnya pernyataan else merupakan bagian dari pernyataan if, yang mana pernyataan else ini digunakan untuk memberikan alternatif perintah apabila kondisi bernilai **FALSE** (0). Bentuk umum pernyataan :

```
If (kondisi) {
    Statement1;
} else {
    Statement2;
}
```

Contoh :

```
<?php
$nilai = 40;
if ($nilai >= 60) {
    echo "Nilai Anda : $nilai. Selamat, Anda Telah Lulus!!";
} else {
    echo "Nilai Anda : $nilai. Sorry, Anda Tidak Lulus!!";
?>
```

Contoh di atas akan menguji kondisi $\$nilai \geq 60$, ternyata kondisi bernilai salah (**FALSE**) sehingga yang dikerjakan adalah statemen dibawah else, yaitu akan ditampilkan hasil output program “Nilai Anda : 40. Sorry, Anda Tidak Lulus!!”.

elseif

jika pernyataan else memberikan alternatif pilihan kedua pada pernyataan di atas, maka pernyataan elseif adalah untuk merumuskan banyak alternatif pilihan (lebih dari dua pilihan). Adapun bentuk umumnya :

```
If  (kondisi1) {
    statement_1;
} elseif (kondisi2) {
    statement_2;
} .....
.....
} else {
    statement_n;
}
```

Contoh :

```
<?php
$nilai = 50;
if (($nilai >= 0) && ($nilai <50)) {
    $indek = "E";
} elseif (($nilai >= 50) && ($nilai <60)) {
    $indek = "D";
} elseif (($nilai >= 60) && ($nilai <75)) {
    $indek = "C";
} elseif (($nilai >= 75) && ($nilai <85)) {
    $indek = "B";
} elseif (($nilai >= 85) && ($nilai <100)) {
    $indek = "A";
} else {
    $indek = "Nilai diluar jangkuan";
}
echo "Nilai Anda : $nilai, dikonversi menjadi '$indek'
?>
```

switch

Serupa dengan pernyataan if – elseif – else, pernyataan switch juga memberikan banyak alternatif pilihan pernyataan. Adapun bentuk umum switch :

```
switch (nilai_ekspresi) {
    case nilai_1 : statement_1; break;
    case nilai_2 : statement_2; break;
    default :
```

```
    statement_n;  
}
```

contoh :

```
<?  
$angka = 6;  
switch ($angka) {  
case 0: $terbilang = "NOL"; break;  
case 1: $terbilang = "SATU"; break;  
case 2: $terbilang = "DUA"; break;  
case 3: $terbilang = "TIGA"; break;  
case 4: $terbilang = "EMPAT"; break;  
case 5: $terbilang = "LIMA"; break;  
case 6: $terbilang = "ENAM"; break;  
case 7: $terbilang = "TUJUH"; break;  
case 8: $terbilang = "DELAPAN"; break;  
case 9: $terbilang = "SEMBILAN"; break;  
default: $terbilang = "Nilai diluar jangkuan!!";  
}  
printf("Bentuk terbilang dari angka '%d' adalah '%s' ",  
$angka, $terbilang);  
?>
```

Praktik :

1. Ketikkan setiap contoh-contoh program di atas dengan **notepad.exe** dan kemudian jalankan!
2. Amati dan analisa hasil tampilan atau output program!
3. Modifikasilah dengan memberikan input nilai yang berbeda, kemudian amati dan analisa hasil tampilan(output) program!
4. Apa yang anda dapat simpulkan terhadap praktikum kali ini?

PERNYATAAN PERULANGAN

Pernyataan perulangan digunakan memproses/mengeksekusi pernyataan atau statemen lebih dari satu kali. Macam-macam pernyataan perulangan di PHP meliputi :

while

Pernyataan while akan melakukan memproses secara berulang terhadap pernyataan berdasarkan atas kondisi.

Bentuk Umum while

```
while (kondisi) {  
    Statemen;  
}
```

Prinsip Kerja : pertama kali pernyataan while akan menguji kondisi yang dirumuskan, jika kondisi bernilai TRUE (1) maka statemen di bawahnya akan dikerjakan sekali lagi. Namun jika kondisi bernilai FALSE (0) maka perulangan akan dihentikan. Setiap kali statemen di bawahnya selesai dikerjakan kondisi akan selalu diuji.

Contoh :

```
<?php  
    $I = 1;  
    while ($I <= 7) {  
        echo "<FONT      SIZE=$I>Perulangan      ke-  
        $I</FONT><BR>\n";  
        $I++;  
    }  
?>
```

Hal yang perlu diperhatikan dalam pernyataan while yaitu

- inisialisasi nilai konter
 - pada program di atas ditunjukkan pada pernyataan \$I = 1;
- penetapan kondisi
 - pada program di atas ditunjukkan pada pernyataan \$I <= 7; yang memiliki maksud, program akan mengulang selama nilai \$I kurang atau sama dengan 7
- Operasi penaikan konter

Hal ini ditunjukkan pada pernyataan `$I++`, maksudnya adalah untuk merubah nilai `$I` setiap kali looping terjadi, sehingga suatu saat perulangan akan dihentikan.

do - while

Seperti halnya pernyataan while, pernyataan do – while memiliki prinsip kerja yang sama dengan pernyataan while, hanya saja pernyataan do – while akan menguji kondisi pada setiap akhir perulangan. Adapun bentuk umum pernyataan do – while:

```
do {  
    Statement;  
} while (kondisi);
```

Prinsip Kerja : Mula-mula statemen akan dikerjakan pertama kali tanpa perlu persyaratan kondisi, selesai penggeraan statemen akan diuji kondisi. Bila kondisi bernilai TRUE (1) maka program akan mengeksekusi statemen sekali lagi. Dan jika kondisi bernilai FALSE maka perulangan akan berakhir.

Contoh :

```
<?php  
    $I = 1;  
    do{  
        echo "<FONT SIZE=$I>Perulangan ke-  
        $I</FONT><BR>\n";  
        $I++;  
    } while ($I <= 7);  
?>
```

for

Pernyataan for biasanya digunakan apabila jumlah perulangannya telah pasti.

Bentuk Umum pernyataan for adalah sebagai berikut :

```
for (ekspresi1; ekspresi2; ekspresi3) {  
    Statement;
```

}

Keterangan :

- ekspresi1 → inisialisasi/penentuan nilai awal konter
- ekspresi2 → perumusan kondisi
- ekspresi3 → operasi penaikan/penurunan konter

Contoh

```
<?php
    for ($I = 1; $I <= 7; $I++) {
        echo "<FONT      SIZE=$I>Perulangan      ke-
$I</FONT><BR>\n";
    }
?>
```

Praktik :

1. Ketikkan setiap program contoh di atas dengan **notepad.exe** dan kemudian jalankan!
2. Amati dan analisa hasil setiap contoh program di atas!
3. Berdasarkan logika Anda, apa perbedaan antara pernyataan **while** dengan **do -while** jika kondisi perulangan diubah menjadi **\$I <= 0**.
4. Kita ketahui bahwa setiap program di atas menampilkan output secara urut dari kecil ke besar (Ascending), Apa yang perlu dirubah pada setiap contoh program di atas apabila output harus menampilkan dari besar ke kecil (Descending).
5. Apa kesimpulan anda praktik kali ini.

Tugas :

1. Buatlah program untuk menampilkan deret bilangan kelipatan tiga (3) dari range 0 sampai 100.
2. Buatlah program untuk menampilkan deret bilangan yang habis dibagi dengan bilangan 6 dan 9 dari range deret 0 sampai 100.

MODUL VII

BASIS DATA MYSQL

TUJUAN

- a. Mahasiswa mengerti apa itu basis data
- b. Mahasiswa memahami struktur data
- c. Mahasiswa mampu menghubungkan web server dengan basis data server
- d. Mahasiswa mampu membuat tabel-tabel dalam basis data dan mengurnya (Manajemen Basis Data)
- e. Mahasiswa memahami mampu membuat query create, read, update, dan delete (CRUD) dalam basis data dengan PHP

KEBUTUHAN

- a. Unit Komputer (Memiliki Sistem Operasi Windows, Linux, dll)
- b. Command Prompt (CMD, Terminal, dll)
- c. Basis data server (dalam modul ini digunakan MySQL dari modul XAMPP)
- d. Web browser (Mozilla, Chrome, dll)

Sekilas Tentang MySQL

MySQL merupakan produk software DBMS yang cukup populer dipasaran, karena DBMS ini didistribusikan secara freeware atau gratis dan juga support untuk berbagai platform bahkan sekarang juga telah tersedia paketan untuk platform sistem operasi Windows. Selain gratis DBMS MySQL ini juga memiliki kecepatan akses yang lebih baik selain produk-produk DBMS sekelas sehingga MySQL sangat cocok untuk diterapkan pada aplikasi-aplikasi pemrograman berbasis internet.

Berikut ini akan diuraikan beberapa perintah-perintah untuk mengakses DBMS MySQL yang dapat diketikkan pada aplikasi MySQL Client.

1. Perintah membuat database

Bentuk umumnya:

```
CREATE DATABASE namadatabase
```

contoh :

```
CREATE DATABASE cobadb;
```

apabila perintah di atas dijalankan DBMS Server MySQL akan menciptakan sebuah database dengan nama 'cobadb'. Tentu saja perintah diketikkan setelah berhasil login ke MySQL. Adapun perintah untuk login ke database seperti berikut ini:

```
prompt>mysql -u username -p password
```

contoh :

```
C:\>mysql -u root -p rahasia
```

sehingga apabila berhasil login akan muncul tampilan berikut ini.

```
mysql>_
```

setelah muncul tampilan di atas mysql telah siap menerima perintah-perintah query untuk akses data seperti membuat database di atas.

2. Perintah mengaktifkan database.

Bentuk umumnya:

```
USE namadatabase
```

contoh:

```
USE cobadb
```

3. Perintah membuat tabel di database.

bentuk umum:

```
CREATE TABLE namabel
(
namafeld1 type [NOT NULL] [AUTO_INCREMENT],
namafeld2 type [NOT NULL] [AUTO_INCREMENT],
.....
[PRIMARY KEY(namafeld) ]
) ;
```

keterangan:

namafeld1, namafeld2, adalah nama field anggota tabel.

NOT NULL, adalah berfungsi untuk membuat status field sebagai field mandatory field, sehingga apabila saat terjadi penambah record tabel field ini tidak boleh dikosongkan.

contoh:

```
CREATE TABLE tamu
(
idtamu int NOT NULL AUTO_INCREMENT,
nmtamu VARCHAR(35) NOT NULL,
email VARCHAR(35) NOT NULL,
PRIMARY KEY(idtamu)
) ;
```

apabila perintah di atas dijalankan maka haruslah databasenya diaktifkan dulu dengan perintah

```
USE cobadb;
```

tabel di atas memiliki tiga macam field dimana field idtamu berfungsi sebagai field kunci primer.

4. Perintah untuk menghapus struktur tabel

Bentuk umumnya:

```
DROP TABLE namabel;
```

contoh:

```
DROP TABLE tamu;
```

efek perintah jika berhasil dijalankan maka tabel dengan nama 'tamu' akan dihapus secara permanen berikut dengan data yang ada dalam tabel juga akan terhapus.

5. Mengubah struktur tabel

bentuk umum:

```
ALTER TABLE namatabel  
ADD | DROP | MODIFY field
```

contoh, menambah field sex (jenis kelamin) pada tabel tamu.

```
ALTER TABLE tamu  
ADD sex enum('L', 'P');
```

sedangkan contoh menghapus field 'sex' pada tabel tamu adalah sebagai berikut:

```
ALTER TABLE tamu  
DROP sex;
```

perintah-perintah di atas tergolong perintah DDL(Data Definition Language), yang hanya dijalankan pada saat database akan dikonstruksikan. Sedangkan perintah-perintah untuk memanipulasi data dalam database (DML singkatan dari Data Manipulation Language) akan diuraikan berikut ini:

6. Perintah input data dalam tabel

Bentuk umumnya adalah seperti berikut ini:

```
INSERT INTO namatabel [(field1,field2, ...)]  
VALUES (datafield1, datafield2, ....);
```

contoh:

```
INSERT INTO tamu (nmtamu, email)
VALUES ('Budiman Raharjo', 'budiman@telkom.net');
```

7. Perintah untuk mengupdate data tabel.

Bentuk Umumnya :

```
UPDATE namatabel
SET namafield1=databaru1, namafield2=databaru2, ...
WHERE kondisi
```

keterangan:

namatabel, adalah nama tabel yang akan dilakukan pengupdatean datanya.

namafield1,namafield2 adalah field-field yang akan disunting datanya dengan databarunya

databaru, adalah diisi dengan data baru yang akan mereplace data lama.

kondisi, adalah merupakan penentuan kriteria record-record yang akan diupdate.

contoh:

```
UPDATE tamu
SET nmtamu='Budiman Sukarno'
WHERE nmtamu='Budiman Raharjo'
```

8. Menghapus record tabel

Bentuk umumnya:

```
DELETE FROM namatabel
WHERE kondisi
```

keterangan:

namatabel, adalah nama tabel yang akan dilakukan penghapusan record datanya.

kondisi, adalah merupakan penentuan kriteria record-record yang akan dihapus.

contoh:

```
DELETE FROM tamu
WHERE idtamu=3
```

9. Perintah untuk menampilkan record-record data tabel

Bentuk umumnya:

```
SELECT * | field1, field2, ...
FROM nama_table1 [ , nama_table2, ...]
[ WHERE kondisi ]
[ GROUP BY ekspresi ]
[ HAVING kondisi_pencarian ]
[ ORDER BY ekspresi_pengurutan [ ASC | DESC ] ]
```

Keterangan :

SELECT *

berfungsi untuk menampilkan data seluruh field dari table yang disebutkan setelah klausua FROM

SELECT field1, field2, ...

berfungsi untuk menampilkan data-data dari field-field yang dituliskan secara eksplisit dari table yang disebutkan setelah klausua SELECT

FROM nama_table1, nama_table2

berfungsi untuk menentukan tabel yang field-field akan ditampilkan data recordnya.

WHERE kondisi

berfungsi untuk menentukan criteria record-record yang akan dihapus.

GROUP BY ekspresi

berisi daftar record-record untuk dijadikan acuan kriteria terhadap record-record data yang dikelompokkan.

HAVING kondisi_pencarian

adalah rumusan kondisi/kriteria record yang digroup yang akan ditampilkan

ORDER BY ekspresi_pengurutan

berisi daftar field-field yang akan dijadikan acuan pengurutan.

ASC

adalah model pengurutan record-record data secara menaik.

DESC

adalah model pengurutan record-record data secara menurun.

Contoh :

```
SELECT * FROM tamu;
```

Perintah di atas jika dijalankan akan menampilkan seluruh record dari seluruh field dalam tabel tamu.

```
SELECT nmtamu, email FROM tamu;
```

perintah di atas jika dijalankan akan menampilkan seluruh data record-record dari field nmtamu dan email.

```
SELECT * FROM tamu  
WHERE nmtamu LIKE 'Budi%';
```

perintah di atas menampilkan data-data record yang memiliki nama dengan awalan 'Budi'

```
SELECT * FROM tamu  
WHERE idtamu=2;
```

perintah di atas akan menampilkan data record tabel yang memiliki idtamu bernilai 2.

contoh lain: untuk menampilkan data dari kombinasi field-field beberapa tabel yang saling berelasi.

tabel kategori

kdkategori *
nmkategori

tabel barang

kdbarang *
nmbarang
jmlbarang
hrgbarang
kdkategori **

berdasarkan struktur tabel-tabel di atas akan dibuat laporan dengan susunan field yang dibutuhkan adalah sebagai berikut:

kdbarang	nmbarang	jmlbarang	hrgbarang	nmkategori
...

sehingga bentuk script SQL yang dapat disusun adalah sebagai berikut:

```
SELECT barang.kdbarang, barang.nmbarang, barang.jmlbarang,  
barang.hrgbarang, kategori.nmkategori  
FROM barang, kategori  
WHERE barang.kdkategori = kategori.kdkategori;
```

Koneksi Data PHP dengan MySQL

Berikut ini akan diuraikan pemakaian fungsi-fungsi untuk mengkoneksikan data ke DBMS MySQL dengan PHP.

1. fungsi koneksi ke instance server MySQL

```
integer mysql_connect (nama_server, nama_user, password);
```

keterangan:

nama_server, adalah nama komputer server yang memuat DBMS MySQL yang dituju.

nama_user, adalah nama user DBMS MySQL yang memiliki otoritas untuk login ke database.

password, adalah password user login ke DBMS MySQL.

contoh:

```
<?
$Id=mysql_connect("localhost", "root", "");
if (! $Id)
    echo "Gagal koneksi dengan Server";
else
    echo "Sukses koneksi dengan Server";
?>
```

program di atas apabila dijalankan akan menampilkan output

Sukses koneksi dengan Server

jika berhasil terkoneksi ke dbms server mysql, namun jika gagal terkoneksi ke database akan ditampilkan output:

Gagal koneksi dengan Server

2. Memilih Database dalam MySQL

```
integer mysql_select_db(string nama_database [, integer id_koneksi]);
```

keterangan:

nama_database, adalah diisi dengan nama database yang akan dituju dalam DBMS MySQL.

id_koneksi, diisi dengan pengenal koneksi yang telah aktif.

contoh:

```
<?php  
$id=mysql_connect("localhost", "root", "rahasia");  
$db=mysql_select_db("test", $id);  
if (! $db)  
    echo "gagal membuka database test!";  
else  
    echo "sukses membuka database test!";  
?>
```

3. fungsi untuk merequest query ke DBMS MySQL

Bentuk umumnya:

```
result mysql_query( string perintah_sql [, integer idkoneksi]);
```

keterangan:

perintah_sql, berisi perintah-perintah untuk mengakses data dalam database dalam sintax SQL

idkoneksi, diisi dengan identitas koneksi yang aktif.

4. fungsi mysql_fetch_row()

bentuk umum:

```
array mysql_fetch_row(result r)
```

keterangan:

r, adalah variabel yang menampung hasil eksekusi query pada fungsi mysql_query().

contoh:

```
<?php
$id=mysql_connect("localhost", "root", "rahasia");
mysql_select_db("test", $id);
$sql="SELECT nimhs, nmmhs, prodi FROM msmhs ";
$hasil=mysql_query($sql, $id);
while ($row=mysql_fetch_row($hasil)){
echo "$row[0], $row[1], $row[2]<br>\n";
}
?>
```

5. Menampilkan Jumlah Record Pada Recordset

Bentuk Umum:

```
integer mysql_num_rows(result r)
```

keterangan:

r, adalah variabel yang menampung hasil eksekusi query pada fungsi mysql_query().

fungsi di atas akan menghasilkan sejumlah record dari recordset.

6. Fungsi *Mysql_Affected_Rows()*

Berfungsi untuk mengetahui jumlah record yang termanipulasi, biasanya fungsi ini digunakan untuk mengidentifikasi keberhasilan perintah query update & delete.

Adapun bentuk umumnya adalah:

```
integer mysql_affected_rows(result r)
```

contoh:

```
<?php
$id=mysql_connect("localhost", "root", "rahasia");
mysql_select_db("test", $id);
```

```
$sql="DELETE FROM msmhs WHERE kdpst='TI' ";
mysql_query($sql, $id);
printf("Telah terhapus %d record", mysql_affected_rows());
?>
```

Praktik:

Diketahui sebuah tabel tamu dalam Database MySQL adalah sebagai berikut :

Nama Field	Tipe Data	Panjang	Keterangan
idtamu	int		identitas tamu, sebagai field kunci primer dan AUTO_INCREMENT
nmtamu	varchar	35	untuk mengisikan data nama tamu
email	varchar	40	untuk mengisikan data alamat email tamu

Buatlah form entry data tamu dengan menggunakan pemrograman PHP!

MODUL VIII

JAVASCRIPT: DASAR, VARIABEL, & FUNGSI

TUJUAN

- a. Mahasiswa memahami dasar-dasar Java Script
- b. Mahasiswa memahami variable dan fungsi dalam Java Script
- c. Mahasiswa mampu membuat dan memadukan document Java Script

KEBUTUHAN

- a. Unit Komputer (Memiliki Sistem Operasi Windows, Linux, dll)
- b. Text editor (Sublime, Notepad, dll)
- c. Web server (dalam modul ini digunakan Apache dari modul XAMPP)
- d. Web browser (Mozilla, Chrome, dll)

RINGKASAN

Setelah Anda berlatih dan menguasai segi tampilan dari web menggunakan HTML, PHP, dan MySQL, sudah saatnya Anda berlatih untuk mengenal pemrograman web yang sebenarnya, yaitu pemrograman skrip. Mulai dari tahap ini Anda akan belajar JavaScript sebagai skrip dasar yang dikenal semua browser (skrip pada sisi client). Anda tidak akan langsung melompat belajar skrip dari sisi server, karena skrip dari sisi server akan selalu memaksa Anda untuk menggunakan satu produk tertentu (ASP, CF, PHP, dan lain-lain), dan biasanya Anda akan menjadi fanatik pada satu produk saja tanpa mengetahui kemampuan produk lain. Anda juga tidak akan belajar bahasa pemrograman Java secara khusus, karena Java fokusnya adalah portabilitas, bukan pada web saja. Disamping JavaScript, bilamana waktu memungkinkan Anda nantinya juga akan berlatih menggunakan VBScript (salah satu anggota keluarga Visual Basic yang mencakup VB, VBScript, VBA/macro, ASP, VB.net, dan ASP.net).

Operator Aritmetika

Operator	Description	Example	Result
+	Addition	x=2 x+2	4
-	Subtraction	x=2 5-x	3
*	Multiplication	x=4 x*5	20
/	Division	15/5 5/2	3 2.5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

Operator Penunjukan

Operator	Example	Is The Same As
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
=	x=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

Operator Perbandingan

Operator	Description	Example
==	is equal to	5==8 returns false
!=	is not equal	5!=8 returns true
>	is greater than	5>8 returns false
<	is less than	5<8 returns true
>=	is greater than or equal to	5>=8 returns false

<code><=</code>	is less than or equal to	<code>5<=8</code> returns true
--------------------	--------------------------	-----------------------------------

Operator Logika

Operator	Description	Example
<code>&&</code>	and	<code>x=6</code> <code>y=3</code> <code>(x < 10 && y > 1)</code> returns true
<code> </code>	or	<code>x=6</code> <code>y=3</code> <code>(x==5 y==5)</code> returns false
<code>!</code>	not	<code>x=6</code> <code>y=3</code> <code>x != y</code> returns true

LATIHAN

Sederhana saja, agar Anda mau membiasakan diri dengan JavaScript, maka Anda harus banyak berlatih mulai dari yang paling dasar. Buatlah semua contoh di bawah ini, jalankan untuk melihat hasilnya, dan lakukan modifikasi untuk lebih mengenal dan mengetahui kegunaan skrip yang baru Anda buat.

1. Menuliskan teks

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

2. Memformat teks dengan tag HTML

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

3. JavaScript yang diletakkan pada bagian HEAD

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

4. JavaScript yang diletakkan pada bagian BODY

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

5. JavaScript eksternal

```
<html>
  <head>
  </head>
  <body>
    <script src="xxx.js">
    </script>
    <p>
      The actual script is in an external script file called
      "xxx.js".
    </p>
</html>
```

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

6. Deklarasi variabel, memberi nilai, dan menampilkannya

```
<html>
  <body>
    <script type="text/javascript">
      var name = "Hege"
      document.write(name)
      document.write("<h1>" + name + "</h1>")
    </script>
    <p>This example declares a variable, assigns a value to it, and then displays the variable.</p>
    <p>Then the variable is displayed one more time, only this time as a heading.</p>
  </body>
</html>
```

7. Fungsi

```
<html>
  <head>
    <script type="text/javascript">
      function myfunction()
      {
        alert("HELLO")
      }
    </script>
  </head>
  <body>
    <form>
      <input type="button"
        onclick="myfunction()"
        value="Call function">
    </form>
    <p>By pressing the button, a function will be called. The function will alert a message.</p>
  </body>
</html>
```

8. Fungsi dengan argumen

```
<html>
  <head>
    <script type="text/javascript">
      function myfunction(txt)
      {
        alert(txt)
      }
    </script>
  </head>
```

```
<body>
<form>
<input type="button"
onclick="myfunction('Hello')"
value="Call function">
</form>
<p>By pressing the button, a function with an argument
will be called. The function will alert
this argument.</p>
</body>
</html>
```

9. Fungsi dengan argumen (lagi)

```
<html>
<head>
<script type="text/javascript">
function myfunction(txt)
{
alert(txt)
}
</script>
</head>
<body>
<form>
<input type="button"
onclick="myfunction('Good Morning!')"
value="In the Morning">
<input type="button"
onclick="myfunction('Good Evening!')"
value="In the Evening">
</form>
<p>
When you click on one of the buttons, a function will
be called. The function will alert
the argument that is passed to it.
</p>
</body>
</html>
```

10. Fungsi yang mengembalikan suatu nilai

```
<html>
<head>
<script type="text/javascript">
function myFunction()
{
return ("Hello, have a nice day!")
}
</script>
</head>
<body>
<script type="text/javascript">
```

```
document.write(myFunction())
</script>
<p>The script in the body section calls a function.</p>
<p>The function returns a text.</p>
</body>
</html>
```

11. Fungsi dengan argumen yang mengembalikan suatu nilai

```
<html>
  <head>
    <script type="text/javascript">
      function total(numberA, numberB)
      {
        return numberA + numberB
      }
    </script>
  </head>
  <body>
    <script type="text/javascript">
      document.write(total(2,3))
    </script>
    <p>The script in the body section calls a function with
    two arguments, 2 and 3.</p>
    <p>The function returns the sum of these two
    arguments.</p>
  </body>
</html>
```

MODUL IX

JAVASCRIPT: OBYEK STRING, DATE, & ARRAY

TUJUAN

- a. Mahasiswa memahami penggunaan Obyek, String, date, dan Array Java Script
- b. Mahasiswa mampu membuat dan memadukan document Java Script dengan HTML dan PHP

KEBUTUHAN

- a. Unit Komputer (Memiliki Sistem Operasi Windows, Linux, dll)
- b. Text editor (Sublime, Notepad, dll)
- c. Web server (dalam modul ini digunakan Apache dari modul XAMPP)
- d. Web browser (Mozilla, Chrome, dll)

RINGKASAN

Setelah Anda terbiasa dengan dasar-dasar JavaScript beserta penggunaan variabel, termasuk mulai mengenal pemrograman skrip fungsi, maka selanjutnya Anda dapat mulai berlatih menggunakan berbagai obyek yang tersedia dalam JavaScript. Seperti telah disampaikan dalam perkuliahan, JavaScript menyediakan banyak obyek *built-in* yang dapat langsung diaplikasikan oleh pengguna. Dengan beragam modifikasi atribut serta nilai yang dapat diberikan pada suatu obyek, maka Anda akan mendapatkan beragam hasil pula.

Properti obyek string

Properties	Explanation
Length	Returns the number of characters in a string

Metode obyek string

Methods	Explanation
anchor()	Returns a string as an anchor
big()	Returns a string in big text
blink()	Returns a string blinking
bold()	Returns a string in bold
charAt()	Returns the character at a specified position
charCodeAt()	Returns the Unicode of the character at a specified position
concat()	Returns two concatenated strings
fixed()	Returns a string as teletype
fontcolor()	Returns a string in a specified color
fontsize()	Returns a string in a specified size
fromCharCode()	Returns the character value of a Unicode
indexOf()	Returns the position of the first occurrence of a specified string inside another string. Returns -1 if it never occurs
italics()	Returns a string in italic
lastIndexOf()	Returns the position of the first occurrence of a specified string inside another string. Returns -1 if it never occurs. Note: This method starts from the right and moves left!
link()	Returns a string as a hyperlink
match()	Similar to indexOf and lastIndexOf, but this method returns the specified string, or "null", instead of a numeric value
replace()	Replaces some specified characters with some new specified characters
search()	Returns an integer if the string contains some specified characters, if not it returns -1
slice()	Returns a string containing a specified character index
small()	Returns a string as small text
split()	Replaces a string with a comma
strike()	Returns a string strikethrough
sub()	Returns a string as subscript
substr()	Returns the specified characters. 14,7 returns 7 characters, from the 14th character (starts at 0)
substring()	Returns the specified characters. 7,14 returns all characters from the 7th up to but not including the 14th (starts at 0)
sup()	Returns a string as superscript
toLowerCase()	Converts a string to lower case

toUpperCase()	Converts a string to upper case
---------------	---------------------------------

Metode obyek date

Methods	Explanation
Date()	Returns a Date object
getDate()	Returns the date of a Date object (from 1-31)
getDay()	Returns the day of a Date object (from 0-6. 0=Sunday, 1=Monday, etc.)
getMonth()	Returns the month of a Date object (from 0-11. 0=January, 1=February, etc.)
getFullYear()	Returns the year of a Date object (four digits)
getYear()	Returns the year of a Date object (from 0-99). Use getFullYear instead !!
getHours()	Returns the hour of a Date object (from 0-23)
getMinutes()	Returns the minute of a Date object (from 0-59)
getSeconds()	Returns the second of a Date object (from 0-59)
getMilliseconds()	Returns the millisecond of a Date object (from 0-999)
getTime()	Returns the number of milliseconds since midnight 1/1-1970
getTimezoneOffset()	Returns the time difference between the user's computer and GMT
getUTCDate()	Returns the date of a Date object in universal (UTC) time
getUTCDay()	Returns the day of a Date object in universal time
getUTCMonth()	Returns the month of a Date object in universal time
getUTCFullYear()	Returns the four-digit year of a Date object in universal time
getUTCHours()	Returns the hour of a Date object in universal time
getUTCMinutes()	Returns the minutes of a Date object in universal time
getUTCSeconds()	Returns the seconds of a Date object in universal time

<code>getUTCMilliseconds()</code>	Returns the milliseconds of a Date object in universal time
<code>parse()</code>	Returns a string date value that holds the number of milliseconds since January 01 1970 00:00:00
<code> setDate()</code>	Sets the date of the month in the Date object (from 1-31)
<code>setFullYear()</code>	Sets the year in the Date object (four digits)
<code>setHours()</code>	Sets the hour in the Date object (from 0-23)
<code>setMilliseconds()</code>	Sets the millisecond in the Date object (from 0-999)
<code>setMinutes()</code>	Set the minute in the Date object (from 0-59)
<code>setMonth()</code>	Sets the month in the Date object (from 0-11. 0=January, 1=February)
<code>setSeconds()</code>	Sets the second in the Date object (from 0-59)
<code> setTime()</code>	Sets the milliseconds after 1/1-1970
<code>setYear()</code>	Sets the year in the Date object (00-99)
<code>setUTCDate()</code>	Sets the date in the Date object, in universal time (from 1-31)
<code>setUTCDay()</code>	Sets the day in the Date object, in universal time (from 0-6. Sunday=0, Monday=1, etc.)
<code>setUTCMonth()</code>	Sets the month in the Date object, in universal time (from 0-11. 0=January, 1=February)
<code>setUTCFullYear()</code>	Sets the year in the Date object, in universal time (four digits)
<code>setUTCHour()</code>	Sets the hour in the Date object, in universal time (from 0-23)
<code>setUTCMinutes()</code>	Sets the minutes in the Date object, in universal time (from 0-59)
<code>setUTCSeconds()</code>	Sets the seconds in the Date object, in universal time (from 0-59)

<code>setUTCMilliseconds()</code>	Sets the milliseconds in the Date object, in universal time (from 0-999)
<code>toGMTString()</code>	Converts the Date object to a string, set to GMT time zone
<code>toLocaleString()</code>	Converts the Date object to a string, set to the current time zone
<code>toString()</code>	Converts the Date object to a string

Metode obyek array

Methods	Explanation
<code>length</code>	Returns the number of elements in an array. This property is assigned a value when an array is created
<code>concat()</code>	Returns an array concatenated of two arrays
<code>join()</code>	Returns a string of all the elements of an array concatenated together
<code>reverse()</code>	Returns the array reversed
<code>slice()</code>	Returns a specified part of the array
<code>sort()</code>	Returns a sorted array

LATIHAN

Seperti dalam pertemuan sebelumnya, gunakan contoh-contoh di bawah ini agar Anda dapat lebih mengenal akrab obyek-obyek dalam JavaScript dan cara memanfaatkannya. Perhatikan bahwa Anda benar-benar harus menelaah setiap baris skrip agar Anda tahu proses kerja setiap skrip yang Anda buat.

1. Menghitung karakter suatu string

```
<html>
  <body>
    <script type="text/javascript">
      var str="W3CSchools is great!"
      document.write("</p>" + str + "</p>")
      document.write(str.length)
    </script>
  </body>
</html>
```

2. Menguji apakah string berisi karakter tertentu, dengan indexOf()

```
<html>
  <body>
    <script type="text/javascript">
      var str="W3CSchools is great!"
      var pos=str.indexOf("School")
      if (pos>=0)
      {
        document.write("School found at position: ")
        document.write(pos + "<br />")
      }
      else
      {
        document.write("School not found!")
      }
    </script>
    <p>This example tests if a string contains a specified word. If the word is found it returns the position of the first character of the word in the original string. Note: The first position in the string is 0!</p>
  </body>
</html>
```

3. Menguji apakah string berisikan karakter tertentu, dengan match()

```
<html>
  <body>
    <script type="text/javascript">
      var str = "W3CSchools is great!"
      document.write(str.match("great"))
    </script>
    <p>This example tests if a string contains a specified word. If the word is found it returns the word.</p>
  </body>
</html>
```

4. Memunculkan bagian string tertentu

```
<html>
  <body>
    <script type="text/javascript">
      var str="W3CSchools is great!"
      document.write(str.substr(2,6))
      document.write("<br /><br />")
      document.write(str.substring(2,6))
    </script>
    <p>
      The substr() method returns a specified part of a string. If you specify (2,6) the returned string will be from the second character (start at 0) and 6 long.
    </p>
</html>
```

```
</p>
<p>
The substring() method also returns a specified part of
a string. If you specify (2,6) it returns all characters
from the second character (start at 0) and up to, but not
including, the sixth character.
</p>
</body>
</html>
```

5. Mengubah menjadi huruf besar atau kecil

```
<html>
<body>
<script type="text/javascript">
var str=("Hello JavaScripters!")
document.write(str.toLowerCase())
document.write("<br>")
document.write(str.toUpperCase())
</script>
</body>
</html>
```

6. Membuat array berisi nama-nama

```
<html>
<body>
<script type="text/javascript">
var famname = new Array(6)
famname[0] = "Jan Egil"
famname[1] = "Tove"
famname[2] = "Hege"
famname[3] = "Stale"
famname[4] = "Kai Jim"
famname[5] = "Borge"
for (i=0; i<6; i++)
{
document.write(famname[i] + "<br>")
}
</script>
</body>
</html>
```

7. Menghitung elemen yang berada dalam array

```
<html>
<body>
<script type="text/javascript">
var famname = new Array("Jan Egil", "Tove", "Hege", "Stale", "Kai
Jim", "Borge")
for (i=0; i<famname.length; i++)
{
document.write(famname[i] + "<br>")
}
```

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

8. Tanggal hari ini

```
<html>
  <body>
    <script type="text/javascript">
      var d = new Date()
      document.write(d.getDate())
      document.write(".")
      document.write(d.getMonth() + 1)
      document.write(".")
      document.write(d.getFullYear())
    </script>
  </body>
</html>
```

9. Jam saat ini

```
<html>
  <body>
    <script type="text/javascript">
      var d = new Date()
      document.write(d.getHours())
      document.write(".")
      document.write(d.getMinutes())
      document.write(".")
      document.write(d.getSeconds())
    </script>
  </body>
</html>
```

10. Mengeset tanggal

```
<html>
  <body>
    <script type="text/javascript">
      var d = new Date()
      d.setFullYear("1990")
      document.write(d)
    </script>
  </body>
</html>
```

11. Melihat waktu UTC

```
<html>
  <body>
    <script type="text/javascript">
      var d = new Date()
      document.write(d.getUTCHours())
```

```
document.write(".")
document.write(d.getUTCMinutes())
document.write(".")
document.write(d.getUTCSeconds())
</script>
</body>
</html>
```

12. Memunculkan nama-nama hari

```
<html>
  <body>
    <script type="text/javascript">
      var d=new Date()
      var weekday=new
        Array("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
        ", "Friday", "Saturday")
      document.write("Today is " + weekday[d.getDay()])
    </script>
  </body>
</html>
```

13. Memunculkan tanggal lengkap

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

14. Membuat jam

```
<html>
  <head>
    <script type="text/javascript">
      var timer = null
      function stop()
      {
        clearTimeout(timer)
      }
```

```
function start()
{
var time = new Date()
var hours = time.getHours()
var minutes = time.getMinutes()
minutes=((minutes < 10) ? "0" : "") + minutes
var seconds = time.getSeconds()
seconds=((seconds < 10) ? "0" : "") + seconds
var clock = hours + ":" + minutes + ":" + seconds
document.forms[0].display.value = clock
timer = setTimeout("start()",1000)
}
</script>
</head>
<body onload="start()" onunload="stop()">
<form>
<input type="text" name="display" size="20">
</form>
</body>
</html>
```

MODUL X

JAVASCRIPT: GAMBAR, FRAME, FORM, DAN CLIENT

TUJUAN

- a. Mahasiswa memahami cara penggunaan Gambar, Frame, Form, Dan Client pada Java Script
- b. Mahasiswa mampu membuat Gambar, Frame, Form, Dan Client pada document Java Script dengan HTML dan PHP

KEBUTUHAN

- a. Unit Komputer (Memiliki Sistem Operasi Windows, Linux, dll)
- b. Text editor (Sublime, Notepad, dll)
- c. Web server (dalam modul ini digunakan Apache dari modul XAMPP)
- d. Web browser (Mozilla, Chrome, dll)

RINGKASAN

Bagian berikutnya dari latihan menggunakan JavaScript adalah penguasaan skrip untuk menangani obyek gambar, frame, form, dan pendektsian elemen web pada client yaitu browser yang digunakan. Pada dasarnya, HTML memang telah memiliki tag untuk penanganan obyek gambar, frame, dan form, tetapi seperti halnya sifat HTML yang statis, maka tag HTML untuk penanganan obyek-obyek tersebut juga memberikan hasil yang statis pula. JavaScript selain dapat digunakan untuk membuatnya lebih dinamis, juga digunakan untuk menjadi *pre-processor* dari masukan yang diberikan pengunjung halaman web ke situs web Anda, sehingga situs web Anda memiliki interaktivitas yang sebenarnya.

LATIHAN

Silakan Anda kerjakan latihan-latihan di bawah ini, dan seperti sebelumnya, perhatikan bagian mana saja yang digunakan agar dapat memberikan hasil yang diinginkan. Untuk beberapa contoh, Anda memerlukan file-file HTML tambahan, silakan Anda buat file HTML tambahan dengan isi yang berbeda-beda agar hasilnya lebih terlihat jelas. Diantara file tambahan yang dibutuhkan terdapat file pemroses masukan dari form. Bahasan file pemroses masukan form belum dijangkau dalam perkuliahan (dibahas khusus dalam pemrograman web tingkat lanjut, skrip pada sisi server), sehingga file HTML untuk pemroses form cukup file *dummy* saja yang berisi pesan teks “Data telah diproses”.

1. **Preload gambar ke memori**

```
<html>
  <head>
    <script type="text/javascript">
      if (document.images)
      {
        a = new Image(160, 120)
        a.src = "gambar.jpg"
      }
    </script>
  </head>
  <body>
    
  </body>
</html>
```

2. Keluar dari frame

```
<html>
  <head>
    <script type="text/javascript">
      function breakout()
      {
        if (window.top != window.self)
        {
          window.top.location="targetpage.htm"
        }
      }
    </script>
  </head>
  <body>
    <form>
      To break out of the frame:
    </form>
  </body>
</html>
```

```
<input type="button" onclick="breakout()" value="Click  
me">  
</form>  
</body>  
</html>
```

3. Melakukan update halaman pada 2 iframe

```
<html>  
  <head>  
    <script type="text/javascript">  
      function twoframes()  
      {  
        document.all("frame1").src="frame_c.htm"  
        document.all("frame2").src="frame_d.htm"  
      }  
    </script>  
  </head>  
  <body>  
    <iframe src="frame_a.htm" name="frame1"></iframe>  
    <iframe src="frame_b.htm" name="frame2"></iframe>  
    <br />  
    <form>  
      <input type="button" onclick="twoframes()" value="Change  
      url of the two iframes">  
    </form>  
  </body>  
</html>
```

4. Validasi alamat e-mail

```
<html>  
  <head>  
    <script type="text/javascript">  
      function validate()  
      {  
        x=document.myForm  
        at=x.myEmail.value.indexOf("@")  
        if (at == -1)  
        {  
          alert("Not a valid e-mail")  
          return false  
        }  
      }  
    </script>  
  </head>  
  <body>  
    <form name="myForm" action="prosesform.htm"  
    onsubmit="return validate()">  
      Enter your E-mail address:  
      <input type="text" name="myEmail">
```

```
<input type="submit" value="Send input">
</form>
<p>This example only tests if the email address contains
an "@" character.</p>
<p>Any "real life" code will have to test for
punctuations, spaces and other things as
well.</p>
</body>
</html>
```

5. Validasi panjang masukan teks

```
<html>
  <head>
    <script type="text/javascript">
      function validate()
      {
        x=document.myForm
        input=x.myInput.value
        if (input.length>5)
        {
          alert("Do not insert more than 5 characters")
          return false
        }
        else
        {
          return true
        }
      }
    </script>
  </head>
  <body>
    <form name="myForm" action="prosesform.htm"
      onsubmit="return validate()">
      In this input box you are not allowed to insert more than
      5 characters:
      <input type="text" name="myInput">
      <input type="submit" value="Send input">
    </form>
  </body>
</html>
```

6. Menjadikan teks masukan sebagai obyek aktif

```
<html>
  <head>
    <script type="text/javascript">
      function setfocus()
      {
        document.forms[0].field.select()
        document.forms[0].field.focus()
      }
    </script>
  </head>
  <body>
    <input type="text" name="field">
  </body>
</html>
```

```
</script>
</head>
<body>
<form>
<input type="text" name="field" size="30" value="input
text">
<input type="button" value="Selected"
onclick="setfocus()">
</form>
</body>
</html>
```

7. Menggunakan tombol radio

```
<html>
<head>
<script type="text/javascript">
function check(browser)
{
document.forms[0].answer.value=browser
}
</script>
</head>
<body>
<form>
Which browser is your favorite<br>
<input type="radio" name="browser"
onclick="check(this.value)" value="Explorer">
Microsoft Internet Explorer<br>
<input type="radio" name="browser"
onclick="check(this.value)" value="Netscape">
Netscape Navigator<br>
<input type="text" name="answer">
</form>
</body>
</html>
```

8. Menggunakan kotak cek

```
<html>
<head>
<script type="text/javascript">
function check()
{
coffee=document.forms[0].coffee
answer=document.forms[0].answer
txt=""
for (i = 0; i<coffee.length; ++ i)
{
if (coffee[i].checked)
{
txt=txt + coffee[i].value + " "
}
```

```
        }
        answer.value="You ordered a coffee with " + txt
    }
</script>
</head>
<body>
<form>
How would you like your coffee?<br>
<input type="checkbox" name="coffee" value="cream">With
cream<br>
<input type="checkbox" name="coffee" value="sugar">With
sugar<br>
<input type="text" name="answer" size="30">
<input type="button" name="test" onclick="check()" value="Order">
</form>
</body>
</html>
```

9. Menggunakan kotak drop-down

```
<html>
<head>
<script type="text/javascript">
function put()
{
option=document.forms[0].dropdown.options[document.forms[0].dropdown.selectedIndex].text
txt=option
document.forms[0].favorite.value=txt
}
</script>
</head>
<body>
<form>
<p>
Select your favorite browser:
<select name="dropdown" onchange="put()">
<option>Internet Explorer
<option>Netscape Navigator
</select>
</p>
<p>
Your favorite browser is:
<input type="text" name="favorite" value="Internet
Explorer">
</p>
</form>
</body>
</html>
```

10. Kotak drop-down sebagai menu

```
<html>
  <head>
    <script type="text/javascript">
      function go(form)
      {
        location=form.selectmenu.value
      }
    </script>
  </head>
  <body>
    <form>
      <select name="selectmenu" onchange="go(this.form)">
        <option>--Select page--
        <option value="http://www.telkom.net">TelkomNet
        <option value="http://www.google.com">Google
        <option value="http://www.unpak.ac.id">UNPAK
      </select>
    </form>
  </body>
</html>
```

11. Kotak teks yang otomatis berpindah fokus bila batasan masukan terpenuhi

```
<html>
  <head>
    <script type="text/javascript">
      function toUnicode(elmnt,content)
      {
        if (content.length==elmnt.maxLength)
        {
          next=elmnt.tabIndex
          if (next<document.forms[0].elements.length)
          {
            document.forms[0].elements[next].focus()
          }
        }
      }
    </script>
  </head>
  <body>
    <p>This script automatically sets focus to the next input field when the current input field's maxlength has been reached</p>

    <form name="myForm">
      <input size="3" tabindex="1" name="myInput"
        maxlength="3" onkeyup="toUnicode(this,this.value)">
      <input size="3" tabindex="2" name="mySecond"
        maxlength="3" onkeyup="toUnicode(this,this.value)">
      <input size="3" tabindex="3" name="myThird"
        maxlength="3" onkeyup="toUnicode(this,this.value)">
    </form>
  </body>
</html>
```

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

12. Deteksi browser yang digunakan

```
<html>
  <head>
    <script type="text/javascript">
      document.write("You are browsing this site with: " +
        navigator.appName)
    </script>
  </head>
  <body>
  </body>
</html>
```

13. Deteksi konfigurasi tampilan yang digunakan

```
<html>
  <body>
    <script type="text/javascript">
      document.write("SCREEN RESOLUTION: ")
      document.write(screen.width + "*")
      document.write(screen.height + "<br>")
      document.write("AVAILABLE VIEW AREA: ")
      document.write(window.screen.availWidth + "*")
      document.write(window.screen.availHeight + "<br>")
      document.write("COLOR DEPTH: ")
      document.write(window.screen.colorDepth + "<br>")
    </script>
  </body>
</html>
```

14. Redirect ke situs web berdasarkan browser yang digunakan

```
<html>
  <head>
    <script type="text/javascript">
      function redirectme()
      {
        bname=navigator.appName
        if (bname.indexOf("Netscape") != -1)
        {
          window.location="http://www.netscape.com"
          return
        }
        if (bname.indexOf("Microsoft") != -1)
        {
          window.location="http://www.microsoft.com"
          return
        }
        window.location="http://www.w3.org"
```

```
    }
</script>
</head>
<body>
<form>
<input type="button" onclick="redirectme()" value="Redirect">
</form>
</body>
</html>
```

MODUL XI

BOOTSTRAP TEMPLATE

TUJUAN

- a. Mahasiswa mengetahui template, fungsi, dan manfaat bootstrap dalam pemrograman web
- b. Mahasiswa mampu mengimplementasikan bootstrap menjadi halaman web

KEBUTUHAN

- a. Unit Komputer (Memiliki Sistem Operasi Windows, Linux, dll)
- b. Text editor (Sublime, Notepad, dll)
- c. Bootstrap Template
- d. Web server (dalam modul ini digunakan Apache dari modul XAMPP)
- e. Web browser (Mozilla, Chrome, dll)

Mengenal Layout Responsive

Layout (desain) responsive, menurut Winarno (2015:3)¹² “*bahwa semua display harus menerima konten yang sama, namun dibangun dengan fleksibel, sehingga mengoptimalkan tampilan untuk tiap piranti*”. Salah satu library untuk membuat desain responsif adalah Bootstrap. Didalam bukunya Abdulloh (2015:141)³ dikatakan “*Hal penting yang perlu diperhatikan dalam membuat desain responsive adalah Layout. Layout website harus bisa berubah-ubah sesuai dengan layar device yang digunakan untuk menampilkan website*”.

¹² Winarno, Edy., dkk. 2015. Desain Web Responsif dengan HTML5 dan CSS3. PT Gramedia . Jakarta..

³ Abdulloh, Rohi. 2015. Web Programming is Easy Panduan Membangun Web Responsive dengan PHP, jQuery, dan Bootstrap. PT Gramedia. Jakarta.

Layout responsive adalah sebuah tampilan website dimana display sebuah perangkat dapat merespon halaman website mengikuti bentuk dari display perangkat pengaksesnya.

Viewport

Tag ini digunakan untuk menyesuaikan tampilan web agar sesuai atau menyesuaikan secara otomatis mengikuti lebar perangkat. Misalkan apabila Anda menggunakan smartphone, ketika perangkat tersebut berada pada posisi horizontal atau vertikal maka lebar halaman web juga akan berubah lebarnya menyesuaikan posisi perangkat tersebut. Selain itu ukuran web juga akan menyesuaikan diri dengan ukuran display (layar) masing-masing perangkat baik itu smartphone, tablet maupun PC.

Code untuk viewport :

Script ini diletakkan diantara tag <head> ... </head>

```
<meta name="viewport" content="width=device-width,  
initial-scale=1">
```

Image Responsive

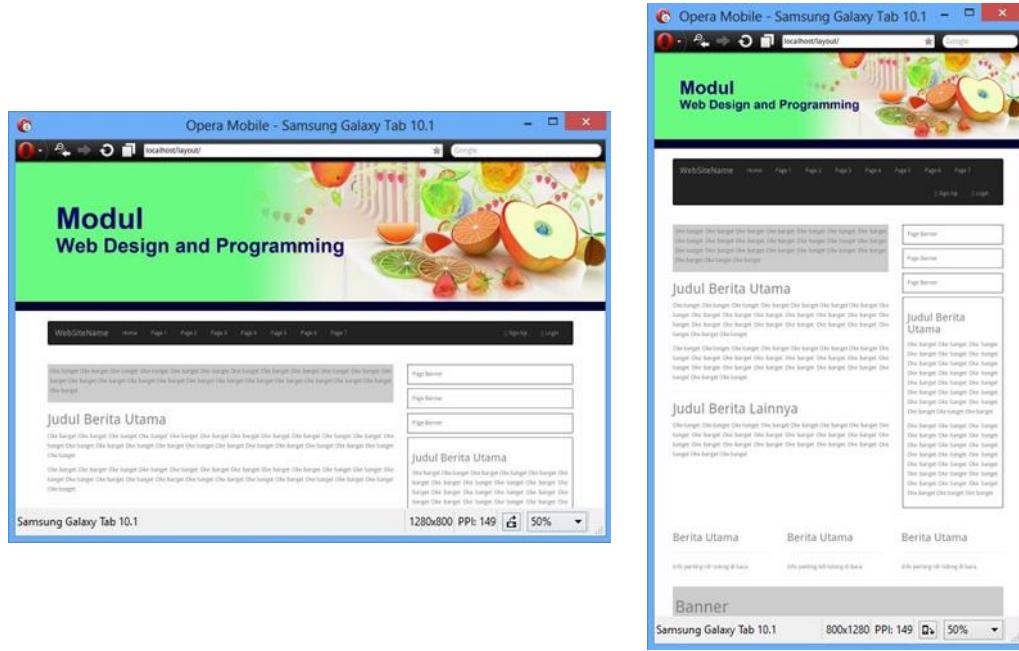
Images responsive maksudnya adalah apabila terdapat sebuah gambar pada bagian isi (content) web, maka gambar tersebut akan ditampilkan sesuai dengan aslinya apabila diakses menggunakan perangkat dekstop seperti PC, tetapi apabila ditampilkan menggunakan perangkat smartphone atau tab maka tampilan gambar tersebut akan mengecil mengikuti posisi dari display tersebut.

Script untuk image responsive :

Script ini diletakkan diantara tag <head> ... </head>

```
<style> img {max-width:100%;} </style>
```

Berikut adalah contoh dari image responsive dan viewport



Posisi horizontal

Posisi vertikal

Sistem Grid

Grid System pada bootstrap merupakan pengaturan ukuran yang di tampilkan pada monitor. Grid system berfungsi untuk membuat pengaturan untuk lebar dari masing-masing komponen web sehingga kita dapat dengan bebas mengatur ke-responsivitas halaman website yang kita buat dengan bootstrap.

Menurut Winarno (2015:149)³“Sistem grid adalah struktur di mana Anda bisa membangun layout dari website. Ini terdiri dari baris horizontal dan kolom vertikal”.

Pada dasarnya sistem grid pada bootstrap merupakan pengaturan ukuran yang di tampilkan pada display monitor, baik itu untuk display monitor, tablet atau smartphone. Sistem grid bootstrap memiliki kegunaan untuk membuat pengaturan untuk lebar (width) layout dari masing-masing komponen web.

³ Winarno, Edy., Ali Zaki, SmitDev Community, 2015. Desain Web Responsif dengan HTML5 dan CSS3. Jakarta. PT. Elex Media Computindo.

Bootstrap memiliki 12 grid dan metode untuk memanggilnya menggunakan *class*. Bootstrap memiliki beberapa class grid dan setiap *class* grid memiliki fungsinya masing-masing. Jenis grid pada bootstrap sebagai berikut :

1. **col-lg-***

digunakan untuk mengatur grid pada monitor komputer yang berukuran besar (≥ 1200 px).

2. **col-md-***

digunakan untuk mengatur grid pada layar monitor komputer yang berukuran sedang (≥ 992 px).

3. **col-sm-***

digunakan untuk mengatur grid pada monitor yang berukuran tablet (≥ 768 px).

4. **col-xs-***

digunakan untuk mengatur grid untuk ukuran handphone (> 768 px).

Tanda (*) merupakan nilai (*value*) berupa angka antara 1 – 12. Jika nilai yang diberikan kecil maka banyaknya grid semakin banyak dan berlaku sebaliknya semakin besar nilai angka yang diberikan semakin sedikit pula jumlah gridnya.

Metode pemanggilan class grid.

```
<div class='col-sm-12'> .. </div>
```

class='col-sm-12' dapat diartikan bahwa grid yang dibuat berbentuk colom (col) dengan ukuran sedang (sm) dan jumlah grid nilainya adalah (12). Berikut contoh sistem grid menggunakan bootstrap untuk layar ukuran layar ukuran sedang (md) :

Ketik script di bawah ini kemudian simpan dengan nama **grid.php**

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<script type="text/javascript"
src="bootstrap/js/jquery.js"></script>
<script type="text/javascript"
src="bootstrap/js/bootstrap.js"></script>
<link rel="stylesheet" type="text/css"
href="bootstrap/css/bootstrap.min.css" />
<title>Blog Kang Arya</title>
<style>img {max-width:100%;} </style>
<style type="text/css"> div{
background: #2ea3f2; text-align: center; border: 1px solid
#ffff; padding: 10px;
color: #ffff;
}
</style>
</head>
<body>
<center><h1>Mengenal Grid System Bootstrap</h1></center>
<br/>
<div class="col-md-12">.col-md-12</div>
<div class="col-md-6">.col-md-6</div>
<div class="col-md-6">.col-md-6</div>
<div class="col-md-4">.col-md-4</div>
<div class="col-md-4">.col-md-4</div>
<div class="col-md-4">.col-md-4</div>
<div class="col-md-3">.col-md-3</div>
<div class="col-md-3">.col-md-3</div>
<div class="col-md-3">.col-md-3</div>
<div class="col-md-2">.col-md-2</div>
<div class="col-md-2">.col-md-2</div>
<div class="col-md-2">.col-md-2</div>
<div class="col-md-2">.col-md-2</div>
<div class="col-md-2">.col-md-2</div>
<div class="col-md-1">.col-md-1</div>
</body>
</html>
```

Berikut adalah hasilnya dapat dilihat di bawah ini :

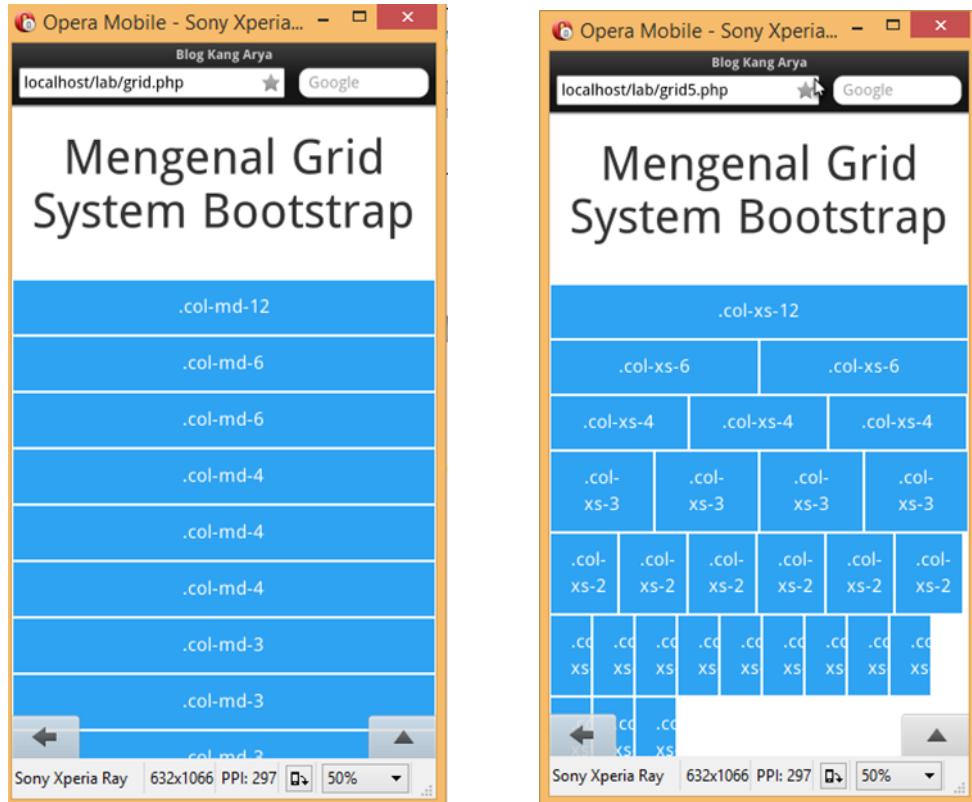
Mengenal Grid System Bootstrap											
col-md-12						col-md-6					
.col-md-6				.col-md-4				.col-md-4			
.col-md-4			.col-md-3			.col-md-3			.col-md-4		
.col-md-3		.col-md-3		.col-md-2		.col-md-2		.col-md-2		.col-md-2	
.col-md-2		.col-md-2		.col-md-2		.col-md-2		.col-md-2		.col-md-2	
col-md-1	col-md-1	col-md-1	col-md-1	col-md-1	col-md-1	col-md-1	col-md-1	col-md-1	col-md-1	col-md-1	col-md-1

Sistem grid pada bootstrap memungkinkan membuat kolom sampai dengan 12 kolom dalam satu halaman. Dari 12 kolom tersebut dapat digunakan sesuai dengan kebutuhan yang pada intinya total kolom harus berjumlah 12 dalam satu halaman.

Jadi dapat di simpulkan bahwa bootstrap membagi halaman website menjadi 12 grid. Seperti sudah dijelaskan di awal bahwa jenis grid yang disediakan oleh bootstrap terdiri dari empat ukuran, yaitu untuk layar monitor menggunakan col-lg-*, untuk layar monitor ukuran sedang menggunakan col-sm-*, untuk layar monitor tablet menggunakan col-md-*, dan selanjutnya untuk layar smartphone menggunakan col-xs-*.

Perbedaannya akan sangat tampak jika menggunakan col-md-* dan col-xs-* apabila diakses menggunakan opera emulator dengan type Sony Experia Ray (632x906 pixel). Berikut hasilnya :

Perbandingan hasil antara menggunakan col-md-* dan col-xs-*:



Menggunakan col-md-*

Menggunakan col-xs-*

Struktur Dasar Grid :

```
<div class="container">
  <div class="row">
    <div class="col-*-*"> ... </div>
    <div class="col-*-*"> ... </div>
    <div class="col-*-*"> ... </div>
  </div>
</div>
```

Keterangan :

- class="container" : digunakan untuk membuat layout layar menjorok kedalam class="container-fluid" : digunakan untuk membuat layar full
- class="row" : digunakan untuk membuat baris
- col : perintah untuk membuat kolom

Table Grid System

	Extra small devices Phones (<768px)	Small devices Tablets (>=768px)	Medium devices Desktops (>=992px)	Large devices Desktops (>=1200px)
Grid behaviour	Horizontal at all times	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints
Container width	None (auto)	750px	970px	1170px
Class prefix	.col-xs-	.col-sm-	.col-md-	.col-lg-
Number of columns	12	12	12	12
Column width	Auto	~62px	~81px	~97px
Gutter width	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)
Nestable	Yes	Yes	Yes	Yes
Offsets	Yes	Yes	Yes	Yes
Column ordering	Yes	Yes	Yes	Yes

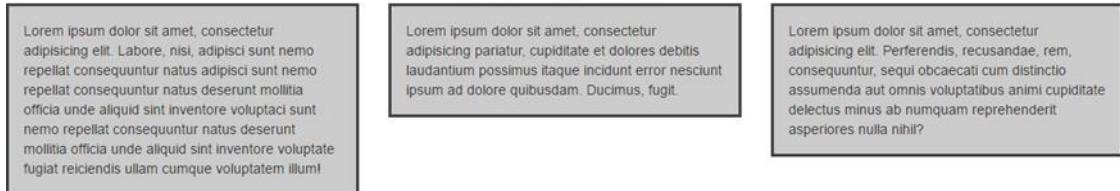
Membuat Grid Sama Tinggi

Untuk membuat grid sama tinggi maka langkah yang efektif adalah dengan menambahkan script CSS. Jika grid yang dibuat tidak sama tinggi atau menyesuaikan dengan perangkat maka pengaturan tinggi didalam style CSS cukup ditambahkan script seperti berikut :

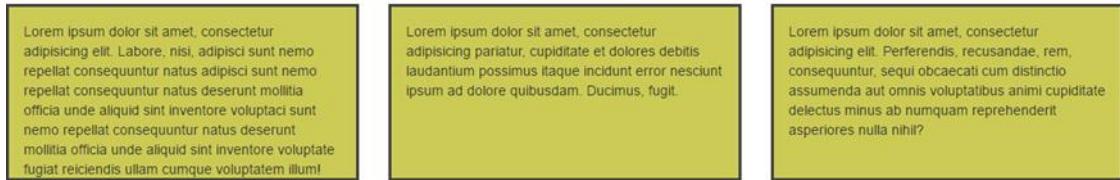
height : 0 auto; → digunakan untuk kolom pada grid menjadi responsive (dinamis/berubah) height : 120px; → digunakan untuk kolom pada grid menjadi fixed (tetap)

Berikut contoh perbedaan antara kolom dinamis dan tetap.

Grid Responsive



Grid Fixed



Membuat Form dengan Bootstrap

Bootstrap menyediakan class khusus lagi untuk mendesign *form* dan pastinya terlihat sangat modern dan bersih, berikut model desain *form* yang disediakan oleh bootstrap:

1) *Form-Vertical*

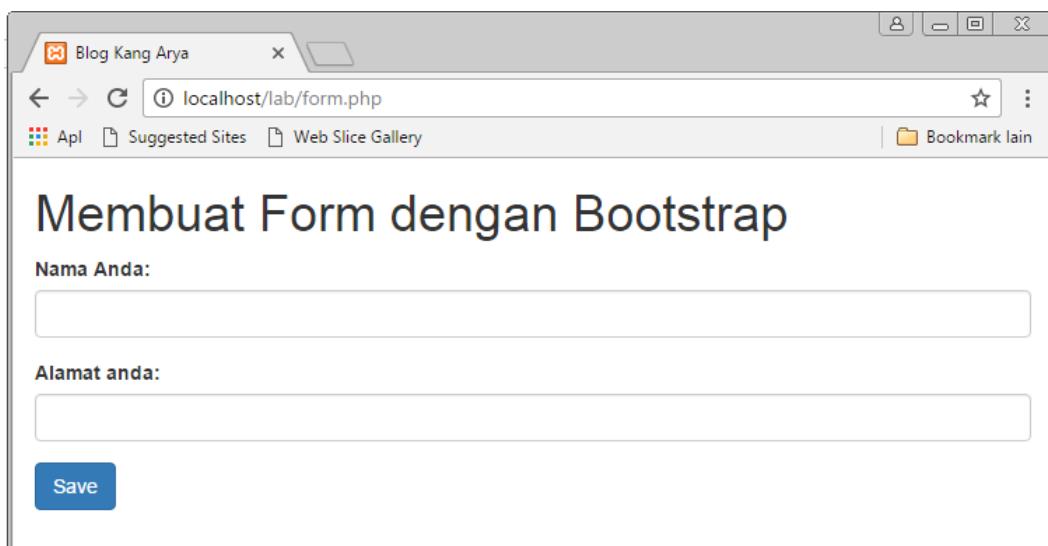
Ketik code di bawah ini kemudian simpan dengan nama *form.php*

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<script type="text/javascript"
src="bootstrap/js/jquery.js"></script>
<script type="text/javascript"
src="bootstrap/js/bootstrap.js"></script>
<link rel="stylesheet" type="text/css"
href="bootstrap/css/bootstrap.min.css" />
<title>Blog Kang Arya</title>
</head>

<body>
<div class="container">
<h1>Membuat Form dengan Bootstrap</h1>
<form>
```

```
<div class="form-group">
<label for="nama">Nama Anda:</label>
<input type="text" class="form-control">
</div>
<div class="form-group">
<label for="alamat">Alamat anda:</label>
<input type="text" class="form-control" id="alamat">
</div>
<div class="form-group">
<label for="save"> </label>
<input type="submit" value="Save" class="btn btn-primary
btn-md">
</form>
</div>
</body>
</html>
```

Hasilnya dapat dilihat di bawah ini:



2) **Form-Inline**

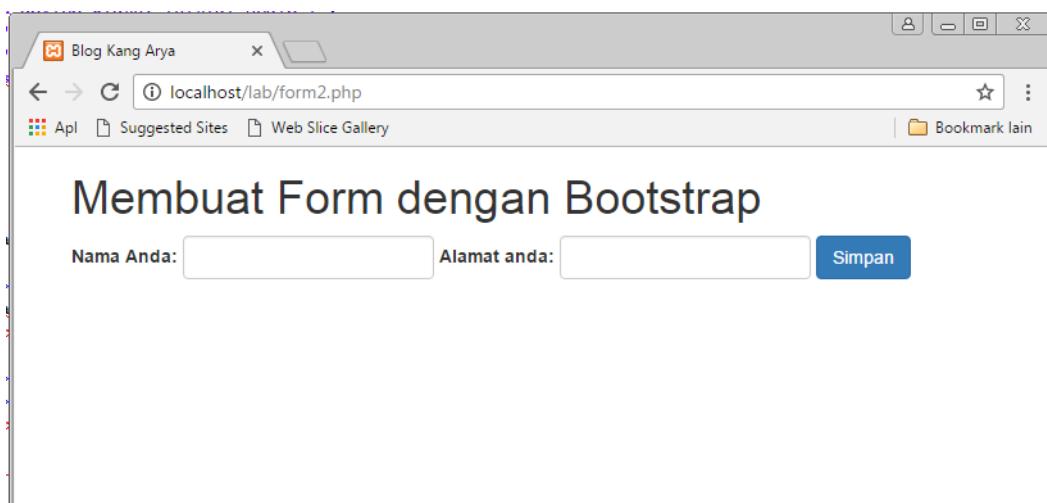
Ketik code di bawah ini kemudian simpan dengan nama *form2.php*

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width,
initial-scale=1">
```

```
<script type="text/javascript">
src="bootstrap/js/jquery.js"></script>
<script type="text/javascript">
src="bootstrap/js/bootstrap.js"></script>
<link rel="stylesheet" type="text/css"
href="bootstrap/css/bootstrap.min.css" />
<title>Blog Kang Arya</title>
</head>

<body>
<div class="container">
<h1>Membuat Form dengan Bootstrap </h1>
<form class="form-inline">
    <div class="form-group">
        <label for="nama">Nama Anda:</label>
        <input type="text" class="form-control">
    </div>
    <div class="form-group">
        <label for="alamat">Alamat anda:</label>
        <input type="text" class="form-control"
id="alamat">
    </div>
    <button type="submit" class="btn btn-primary">Simpan</button>
</form>
</div>
</body>
</html>
```

Berikut hasilnya dapat dilihat di bawah ini :



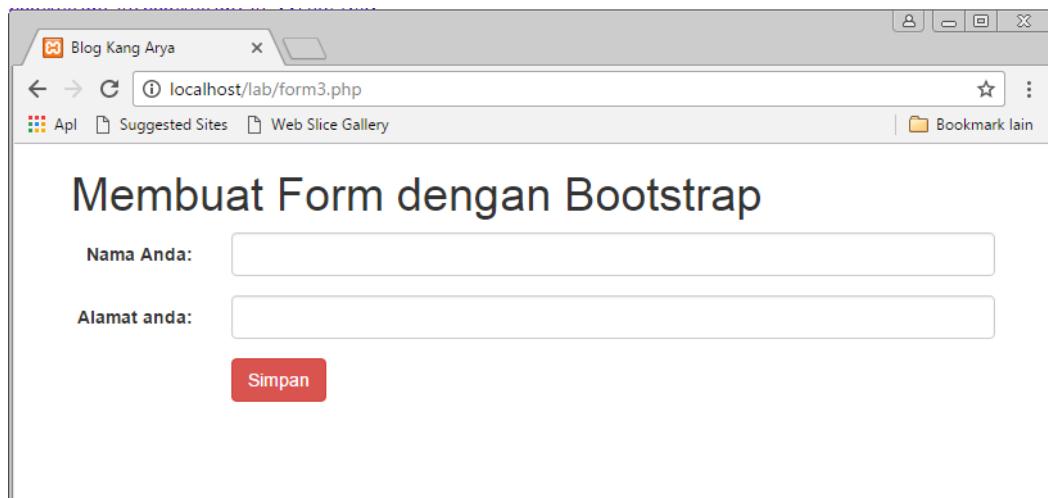
3) **Form-Horizontal**

Ketik code di bawah ini kemudian simpan dengan nama *form3.php*

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<script type="text/javascript"
src="bootstrap/js/jquery.js"></script>
<script type="text/javascript"
src="bootstrap/js/bootstrap.js"></script>
<link rel="stylesheet" type="text/css"
href="bootstrap/css/bootstrap.min.css" />
<title>Blog Kang Arya</title>
</head>

<body>
<div class="container">
<h1>Membuat Form dengan Bootstrap</h1>
<form class="form-horizontal">
<div class="form-group">
<label class="control-label col-sm-2"
for="nama">Nama Anda:</label>
<div class="col-sm-10">
<input type="text" class="form-control">
</div>
</div>
<div class="form-group">
<label class="control-label col-sm-2"
for="alamat">Alamat anda:</label>
<div class="col-sm-10">
<input type="text" class="form-control"
id="alamat">
</div>
</div>
<div class="form-group">
<label class="control-label col-sm-2"
for="tombol"></label>
<div class="col-sm-10">
<input type="Submit" value="Simpan"
class="btn btn-danger">
</div>
</div>
</form>
</div>
</body>
</html>
```

Berikut hasilnya dapat dilihat di bawah ini :



Membuat Table dengan Bootstrap

Membuat table dengan bootstrap ini akan di bahas tentang cara membuat atau mendesign table dengan menggunakan 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.

Tabel standar

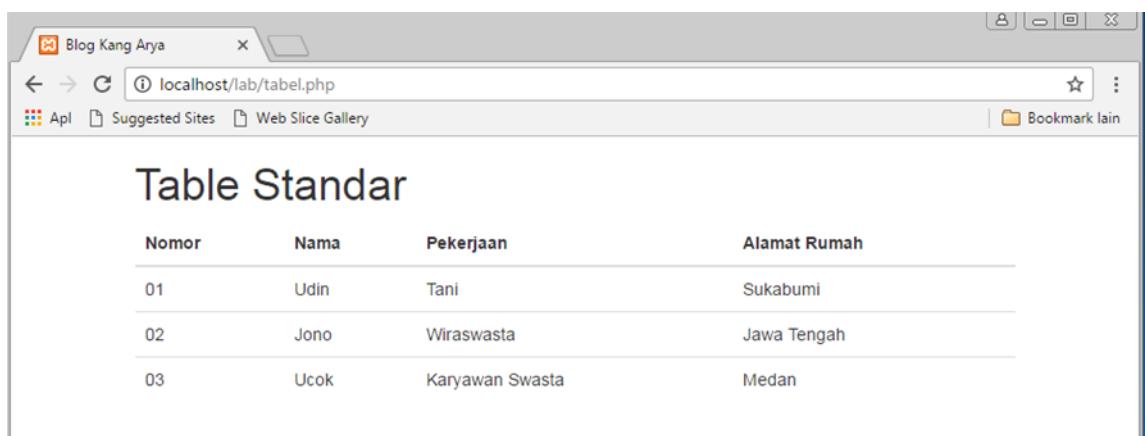
Untuk membuat tabel standar cukup dengan menambahkan class .table pada tag <table>
Ketik script di bawah ini kemudian simpan dengan nama **table.php**

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<link rel="stylesheet" type="text/css"
href="bootstrap/css/bootstrap.min.css" />
<style>img {max-width:100%;}</style>
<title>Blog Kang Arya</title>
</head>

<body>
<div class="container">
<div class="row">
    <div class="col-sm-12">
        <h1>Table Standar</h1>
```

```
<table  
class="table">  
    <thead>  
        <tr>  
            <th>Nomor</th>  
            <th>Nama</th>  
            <th>Pekerjaan</th>  
            <th>Alamat Rumah</th>  
        </tr>  
    </thead>  
    <tbody>  
        <tr>  
            <td>01</td>  
            <td>Udin</td>  
            <td>Tani</td>  
            <td>Sukabumi</td>  
        </tr>  
        <tr>  
            <td>02</td>  
            <td>Jono</td>  
            <td>Wiraswasta</td>  
            <td>Jawa Tengah</td>  
        <tr>  
            <td>03</td>  
            <td>Ucok</td>  
            <td>Karyawan Swasta</td>  
            <td>Medan</td>  
        </tr>  
    </tbody>  
</table>  
</div>  
</div>  
</div>  
</body>  
</html>
```

Berikut hasilnya dapat dilihat di bawah ini :



The screenshot shows a web browser window titled "Blog Kang Arya". The address bar displays "localhost/lab/tabel.php". The main content area has a heading "Table Standar". Below the heading is a table with four columns: "Nomor", "Nama", "Pekerjaan", and "Alamat Rumah". The table contains three rows of data:

Nomor	Nama	Pekerjaan	Alamat Rumah
01	Udin	Tani	Sukabumi
02	Jono	Wiraswasta	Jawa Tengah
03	Ucok	Karyawan Swasta	Medan

Tabel Bordered

Caranya adalah dengan menambahkan class . table-bordered di dalam tag <table>
Ketik script di bawah ini kemudian simpan dengan nama table_boot2.php

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<script type="text/javascript"
src="bootstrap/js/jquery.js"></script>
<script type="text/javascript"
src="bootstrap/js/bootstrap.js"></script>
<link rel="stylesheet" type="text/css"
href="bootstrap/css/bootstrap.min.css" />
<title>Blog Kang Arya</title>
</head>

<body>
<div class="container">
<h1>Cara Membuat table dengan Bootstrap</h1>
    <table class="table table-bordered">
        <thead>
            <tr>
                <th>No</th>
                <th>Makanan</th>
                <th>Harga</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>1</td>
                <td>Bakso</td>
                <td>12.000</td>
            </tr>
            <tr>
                <td>2</td>
                <td>Mie Goreng</td>
                <td>7.000</td>
            </tr>
            <tr>
                <td>3</td>
                <td>Nasi Goreng</td>
                <td>15.000</td>
            </tr>
        </tbody>
    </table>
</div>
```

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

Berikut hasilnya dapat dilihat di bawah ini :

No	Makanan	Harga
1	Bakso	12.000
2	Mie Goreng	7.000
3	Nasi Goreng	15.000

Tabel Bordered Stripped

Caranya adalah dengan menambahkan class .table-bordered table-striped di dalam tag <table> Ketik script di bawah ini kemudian simpan dengan nama **tabel_boot3.php**

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<script type="text/javascript" src="bootstrap/js/jquery.js"></script>
<script type="text/javascript" src="bootstrap/js/bootstrap.js"></script>
<link rel="stylesheet" type="text/css" href="bootstrap/css/bootstrap.min.css" />
<title>Blog Kang Arya</title>
</head>

<body>
<div class="container">
<h1>Cara Membuat table dengan Bootstrap</h1>
<table class="table table-bordered table-striped">
<thead>
<tr>
```

```
<th>No</th>
<th>Makanan</th>
<th>Harga</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bakso</td>
<td>12.000</td>
</tr>
<tr>
<td>2</td>
<td>Mie Goreng</td>
<td>7.000</td>
</tr>
<tr>
<td>3</td>
<td>Nasi Goreng</td>
<td>15.000</td>
</tr>
</tbody>
</table>
</div>
</body>
</html>
```

Berikut hasilnya dapat dilihat di bawah ini :

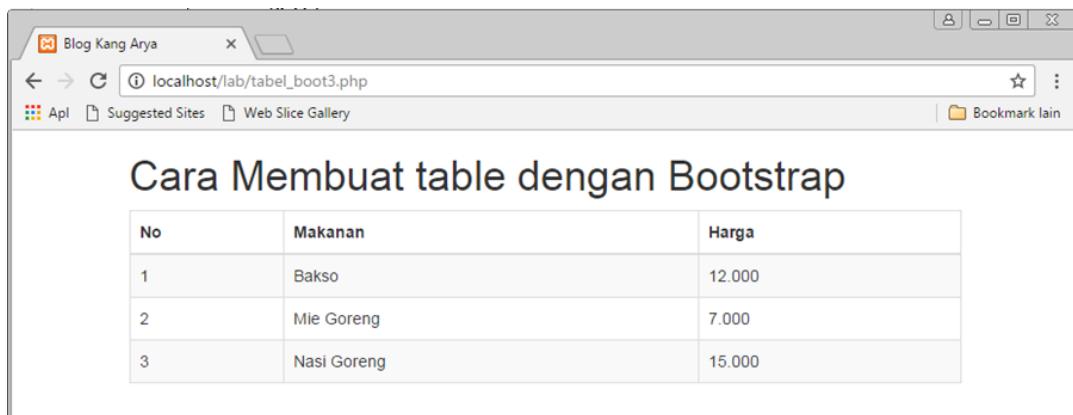


Table Hover

Caranya adalah dengan menambahkan class .table-bordered table-striped table-hover di dalam tag <table>. Ketik script di bawah ini kemudian simpan dengan nama tabel_boot4.php

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<script type="text/javascript"
src="bootstrap/js/jquery.js"></script>
<script type="text/javascript"
src="bootstrap/js/bootstrap.js"></script>
<link rel="stylesheet" type="text/css"
href="bootstrap/css/bootstrap.min.css" />
<title>Blog Kang Arya</title>
</head>

<body>
<div class="container">
<h1>Cara Membuat table dengan Bootstrap</h1>
<div class="table-responsive">
<table class="table table-bordered table-striped table-
hover">
<thead>
<tr>
<th>No</th>
<th>Makanan</th>
<th>Harga</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bakso</td>
<td>12.000</td>
</tr>
```

```
<tr>
<td>2</td>
<td>Mie Goreng</td>
<td>7.000</td>
</tr>
<tr>
<td>3</td>
<td>Nasi Goreng</td>
<td>15.000</td>
</tr>
</tbody>
</table>
</div>
</div>
</body>
</html>
```

Berikut hasilnya dapat dilihat di bawah ini :



The screenshot shows a web browser window titled "Blog Kang Arya". The address bar displays "localhost/lab/tabel_boot3.php". The main content area has a heading "Cara Membuat table dengan Bootstrap". Below the heading is a table with the following data:

No	Makanan	Harga
1	Bakso	12.000
2	Mie Goreng	7.000
3	Nasi Goreng	15.000

MODUL XI

FRAMEWORK CODEIGNITER (RAMOS SOMYA)

TUJUAN

- a. Mahasiswa mengetahui prinsip, cara kerja, serta fungsi Framework Codeigniter
- b. Mahasiswa memahami model, view, dan controller dalam Framework Codeigniter

KEBUTUHAN

- a. Unit Komputer (Memiliki Sistem Operasi Windows, Linux, dll)
- b. Text editor (Sublime, Notepad, dll)
- c. Framework Codeigniter
- d. Web server (dalam modul ini digunakan Apache dari modul XAMPP)
- e. Web browser (Mozilla, Chrome, dll)

Prinsip Dasar Framework

Framework secara umum adalah sebuah susunan atau rangkaian kerja yang tetap dan dibuat sedemikian rupa yang kemudian dapat digunakan kembali dalam sebuah aktifitas kerja yang lain tapi tetap dalam satu area kerja dengan rangkaian kerja sebelumnya. Sedangkan framework web yang akan dipelajari pada modul ini mempunyai definisi sebagai kumpulan modul dalam bentuk class library yang dapat digunakan untuk membentuk aplikasi web yang besar dengan memanfaatkan modul-modul class library tersebut.

Sebagai contoh, jika dalam sebuah framework terdapat class “Shopping Cart”, di mana class ini berfungsi untuk membuat keranjang belanja pada aplikasi toko online, maka setiap aplikasi yang dirasa membutuhkan fungsi ini, programmer

hanya perlu “menempelkan” class ini pada aplikasi. Class ini dapat digunakan berulang kali dalam aplikasi web yang berbeda-beda.

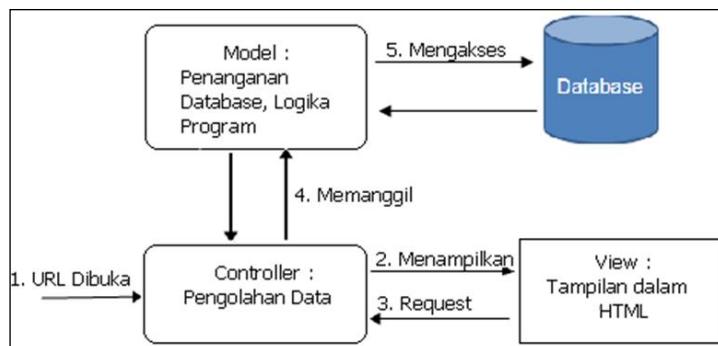
Untuk menggunakan library-library yang disediakan oleh suatu framework, developer harus menyesuaikan diri dengan aturan-aturan pada framework tersebut, karena setiap framework mempunyai aturan yang berbeda. Saat ini terdapat banyak framework PHP yang dapat dipakai dengan gratis, yaitu: CodeIgniter, CakePHP, Yii, Laravel, Prado, Zend dan beberapa lainnya.

Framework CodeIgniter

Framework CodeIgniter (CI) merupakan salah satu framework PHP yang populer di kalangan web developer. Pada tahun 2015, framework ini menduduki peringkat 1 di Indonesia sebagai framework PHP yang paling banyak dipakai. Keunggulan CI antara lain menganut prinsip Model View Controller (MVC), open source (gratis), sangat ringan di semua platform, library lengkap, dokumentasi lengkap dan berbasis pada PHP5.

Prinsip Model View Controller (MVC)

MVC memungkinkan programmer untuk memisah program ke dalam 3 folder sesuai dengan fungsinya, yaitu bagian Model, View dan Controller. Cara ini dapat mempermudah pembedaan antara tampilan dan program. Secara khusus sangat baik untuk project di mana designer bekerja dengan file template yang akan memperkecil banyaknya kode di setiap template karena sudah dipisah dengan baik. Arsitektur MVC digambarkan pada Gambar 1.



Gambar 1 Arsitektur MVC pada CI

Saat pertama kali halaman web diakses, sebenarnya yang dipanggil adalah bagian controller, di mana controller akan memanggil halaman view (tampilan dalam HTML) dan menampilkannya pada browser. Jika membutuhkan data dari basis data untuk diakses / ditampilkan, maka bagian controller akan memanggil bagian model terlebih dahulu untuk mengambil data dari basis data tersebut, kemudian memanggil bagian view dan menampilkannya pada browser.

Memulai Framework CodeIgniter

Untuk menggunakan framework CI dalam membuat web, programmer bisa mengunduh framework ini melalui website CI (www.codeigniter.com). Saat modul ini ditulis, versi CI yang terbaru adalah versi 3.1.5. Sedangkan untuk mempelajari CI, pada modul ini akan menggunakan CI versi 3.0.3. Letakkan framework CI yang sudah diekstrak ke dalam directory WWW (Wamp Server) atau htdocs (XAMPP). Ubah nama folder CI tersebut dengan nama cobaci. Susunan folder dalam CI dapat dilihat pada Gambar 2.



Gambar 2 Susunan Folder pada CI

Secara default, framework CI terdiri dari 3 folder utama, yaitu application, system dan user_guide. Folder utama adalah folder application di mana pada folder ini akan tersusun dalam beberapa subfolder. Subfolder yang paling penting adalah config, model, view dan controller. Programmer juga dapat menambahkan library atau file pendukung lainnya yang tidak disediakan oleh CI, misalnya Bootstrap. Untuk library tambahan dapat disimpan pada directory project supaya dapat diakses oleh aplikasi. Selanjutnya, jika diakses melalui browser, maka akan terlihat seperti Gambar 3.



Gambar 2 Tampilan Default CI pada Browser

Penjelasan:

1. Buka file routes.php yang terletak di folder application/config.
2. Pada baris ke 52 terdapat perintah \$route['default_controller'] = 'welcome';
3. Hal ini berfungsi untuk menentukan controller mana yang akan dieksekusi pertama kali saat website dijalankan pada browser.
4. Buka file Welcome.php yang terletak di folder application/controllers (Gambar 3).

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Welcome extends CI_Controller {
5
6     public function index()
7     {
8         $this->load->view('welcome_message');
9     }
10 }

```

Gambar 3 File Welcome.php

Pada baris ke 6 terdapat fungsi index(). Fungsi ini akan dijalankan ketika controller Welcome dipanggil.

5. Pada baris ke 8 terdapat perintah untuk menampilkan halaman view dengan nama welcome_message.
6. Bukanlah file welcome_message.php yang ada di folder application/views.
7. Inilah halaman web yang ditampilkan di browser.

Latihan:

Ketikkan kode program berikut pada fungsi index pada controller Welcome:

Kode Program 1 Modifikasi Controller Welcome

```

$pesan["pesan1"] = "Ini adalah pesan pertama";
$pesan["pesan2"] = "Ini adalah pesan kedua";
$this->load->view('myview', $pesan);

```

Buat sebuah view baru dengan nama myview.php dan simpan pada folder application/views dan ketikkan Kode Program 2.

Kode Program 2 File myview.php

```

<html>
<head>
<title>File View Pertama</title>
</head>

<body>
File view berhasil ditampilkan
<p>Nilai array pesan1 adalah: <?php echo $pesan1;
?></p>

```

```
<p>Nilai array pesan1 adalah: <?php echo $pesan2;  
?></p>  
</body>  
  
</html>
```

Jalankan pada browser, lihat output-nya dan pelajari proses yang terjadi tersebut dengan prinsip MVC pada CI. Jika dapat dipahami, maka dapat disimpulkan bahwa Anda sudah paham dengan cara kerja MVC pada CI dan dapat beralih ke bagian selanjutnya.

Memproses Inputan Form dengan CI

Untuk mempelajari pemrosesan inputan dari form dengan CI, tambahkan sebuah file dengan nama myform.php di bagian view dan ketikkan Kode Program 3.

Kode Program 3 View myform

```
<html>  
<head>  
    <title>Form CI</title>  
</head>  
<body>  
  
<form action="<?php echo  
site_url('ProsesController/proses') ?>" method="post">  
    <tr>  
        <td>Masukkan Nama Anda:</td>  
        <td><input type="text" name="nama" size="20"/>  
            <input type="submit" value="Kirim" />  
        </td>  
    </tr>  
  
    </form>  
</table>  
  
</body>  
</html>
```

Tampilan myform akan ditampilkan pertama kali ketika project cobaci diakses dari browser. Oleh karena itu perlu dilakukan modifikasi pada bagian controller. Tambahan sebuah file controller baru pada folder controller dengan nama ProsesController.php dan tambahkan Kode Program 4.

Kode Program 4 Controller ProsesController

```
<?php

class ProsesController extends CI_Controller {

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

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

Supaya pada saat project diakses oleh browser dan menampilkan halaman myform, pada bagian file route harus diubah. Buka file route yang ada di folder config dan pada baris ke 52 ubah menjadi: **\$route['default_controller'] = 'ProsesController';**

Selanjutnya buka pada browser dan akan muncul sebuah form dengan sebuah inputan berupa textfield dan sebuah tombol untuk mengirimkan inputan tersebut seperti Gambar 4.

Gambar 4 Tampilan pada Browser

Untuk dapat memproses inputan pada form, perlu dibuat sebuah fungsi pada ProsesController seperti yang sudah ditetapkan pada bagian form action di myform. Oleh sebab itu tambahkan Kode Program 5.

Kode Program 5 Fungsi proses pada ProsesController

```
public function proses()
{
    $nama = $this->input->post("nama");
    $data['welcome'] = "Selamat Datang, ".$nama;
    $this->load->view('greeting', $data);
}
```

Selanjutnya tambahkan sebuah file baru pada folder view dan beri nama greeting.php. Ketikkan Kode Program 6.

Kode Program 6 File greeting pada Folder View

```
<html>
<head>
<title>Form CI</title>
</head>

<body>
<p><h2><?php echo $welcome; ?></h2></p>
</body>

</html>
```

Jalankan project pada browser, masukkan inputan pada textfield dan tekan tombol kirim. Inputan tersebut akan dikirimkan ke fungsi proses dan akan ditampilkan hasilnya pada halaman greeting.

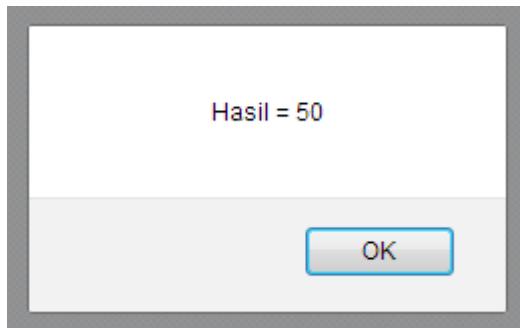
Latihan:

Buatlah program kalkulator sederhana menggunakan framework CodeIgniter (Buat Controller dan file View baru pada project sebelumnya). Tampilan program terlihat pada Gambar 5.

Form Kalkulator	
Angka 1	10
Angka 2	5
Operator	*
<input type="button" value="hitung"/> <input type="button" value="reset"/>	

Gambar 5 Tampilan Form Kalkulator

Jika diproses, maka hasil operasi akan ditampilkan pada MessageBox seperti pada Gambar 6.



Gambar 6 Tampilan Hasil Operasi Kalkulator

Referensi:

Hakim, L. 2010. *9 Langkah Menjadi Master Framework Codeigniter*. Yogyakarta: Penerbit Lokomedia.