

Modul Praktikum Pemrograman Web



web application



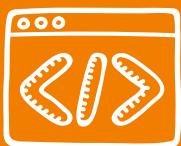
web service



wireless application protocol



web design



oleh: **Winarno, S.Si., M.Eng**

**Program Studi Informatika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Sebelas Maret**

Bab 1

Persiapan

Sebelum kita mempelajari HTML dan CSS, ada beberapa persiapan yang perlu dilakukan. Seperti penginstallan aplikasi-aplikasi yang dibutuhkan dan aplikasi-aplikasi pendukung dalam mempelajari HTML dan CSS.

1.1. Web Browser

Web browser adalah perangkat utama yang akan kita gunakan untuk menampilkan halaman web yang pada dasarnya terbuat dari HTML dan CSS. Saya yakin di setiap komputer telah terinstall Web Browser bawaan seperti Internet Explorer (Windows), Safari (Mac) dan Firefox (Linux Ubuntu).

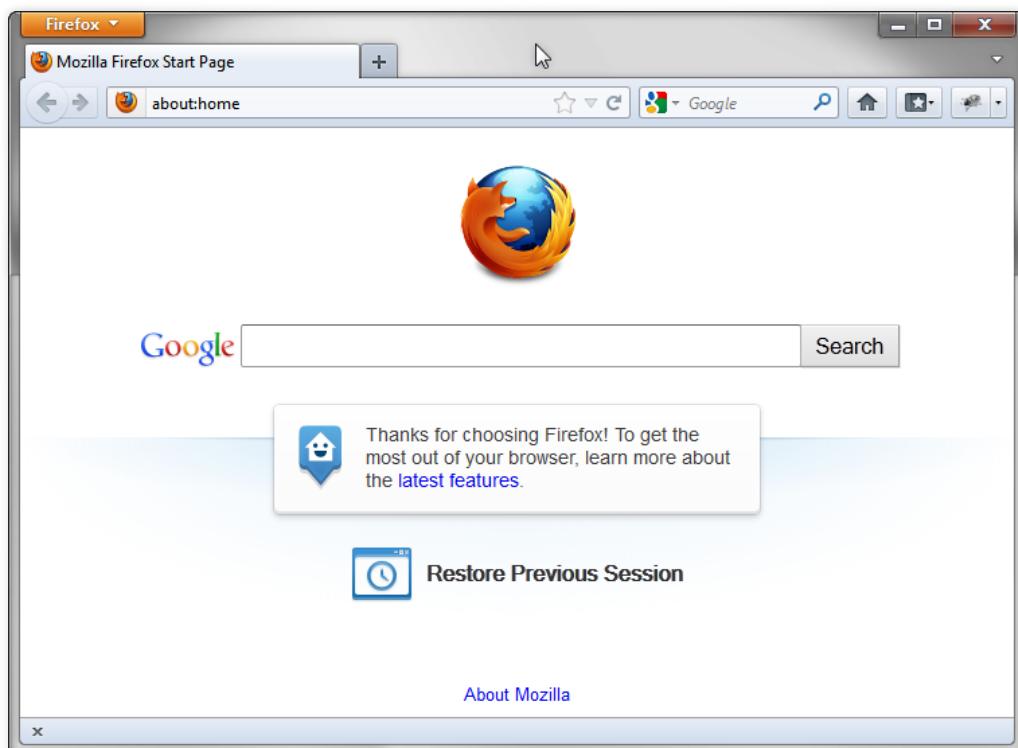
Setiap browser memiliki perbedaan dalam hal menampilkan halaman web dan fitur-fitur yang didukung dalam HTML dan CSS. Boleh jadi halaman web yang anda buat ditampilkan benar pada salah satu browser namun acak-acakan pada browser lainnya. Untuk itu perlu beberapa browser yang terinstall dalam komputer anda untuk menguji penampilan website yang dibuat.

Isu-isu kompatibilitas ini akan anda temui ketika mempelajari CSS nantinya yang berhubungan dalam penampilan halaman web.

Berikut ini adalah beberapa browser yang dapat anda download dari Internet serta review singkat.

Firefox

Web browser ini bisa dikatakan web browser yang paling disenangi oleh para developer web karena kekayaannya dalam hal Add-on¹. Anda dapat mendownloadnya di <http://firefox.com>



Gambar 1 Mozilla Firefox 9

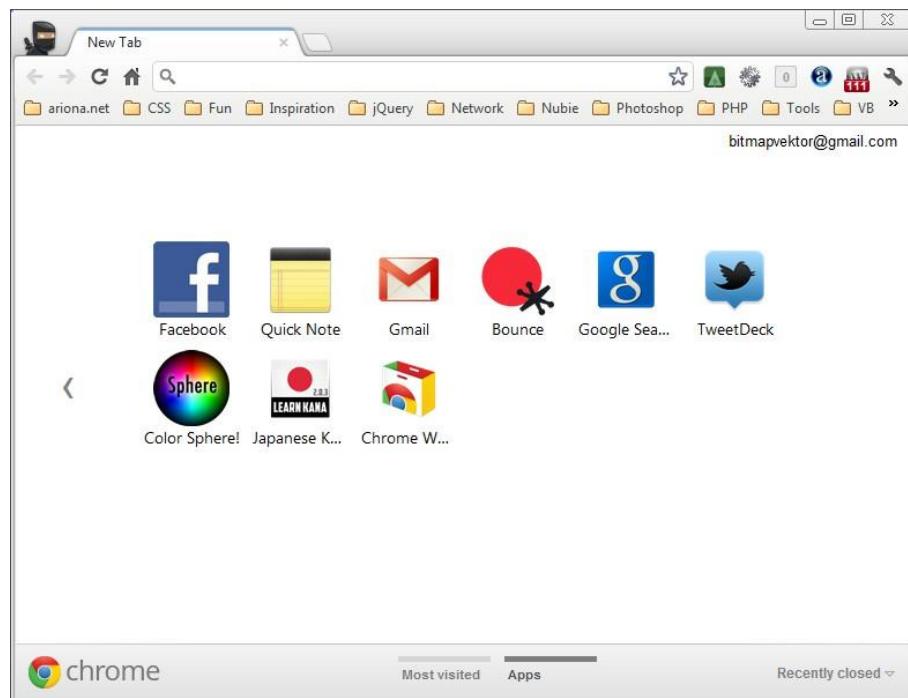
Google Chrome

Google Chrome adalah web browser besutan Google yang memiliki beberapa keunggulan. Chrome bisa dikatakan browser yang sangat ringan, cepat² dan kaya akan aplikasi/Add-on. Selain itu Chrome menggunakan engine³ Webkit, engine browser yang rata-rata mendukung fitur-fitur terbaru dari teknologi HTML dan CSS. Anda dapat mendownloadnya di <http://google.com/chrome>

¹ Add-on/Plugin adalah Aplikasi tambahan yang dapat dipasang sehingga memperkaya fitur software yang bersangkutan

² Kecepatan loading web tergantung dari kecepatan internet yang anda gunakan

³ Engine/Mesin adalah kode utama dalam aplikasi untuk menampilkan halaman web contohnya Mozilla untuk firefox, Webkit untuk Google Chrome dan Safari.

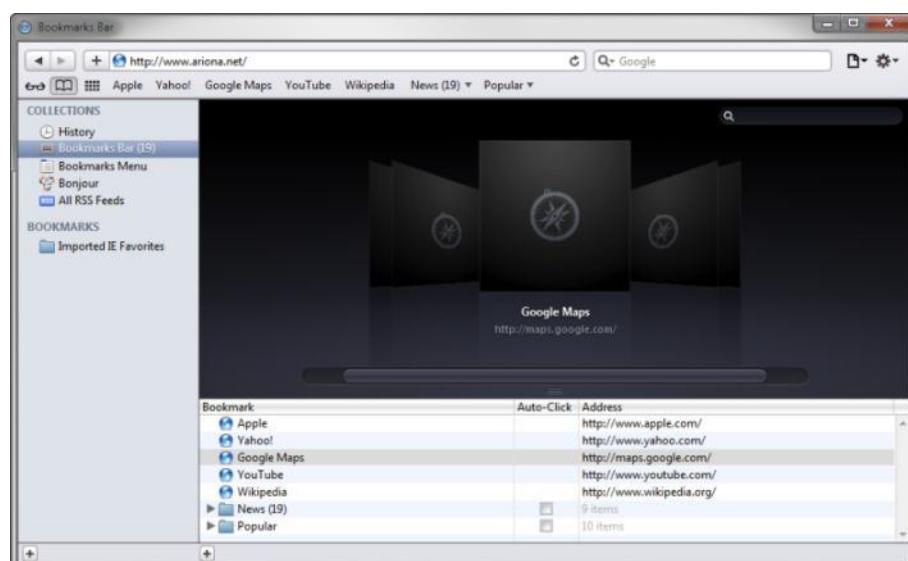


Gambar 2 Google Chrome 17

Safari

Safari adalah web browser dari Apple. Sama halnya dengan Chrome, safari menggunakan engine webkit sehingga mendukung fitur-fitur terbaru HTML dan CSS.

Browser ini tersedia untuk Mac dan Windows saja. <http://www.apple.com/safari>

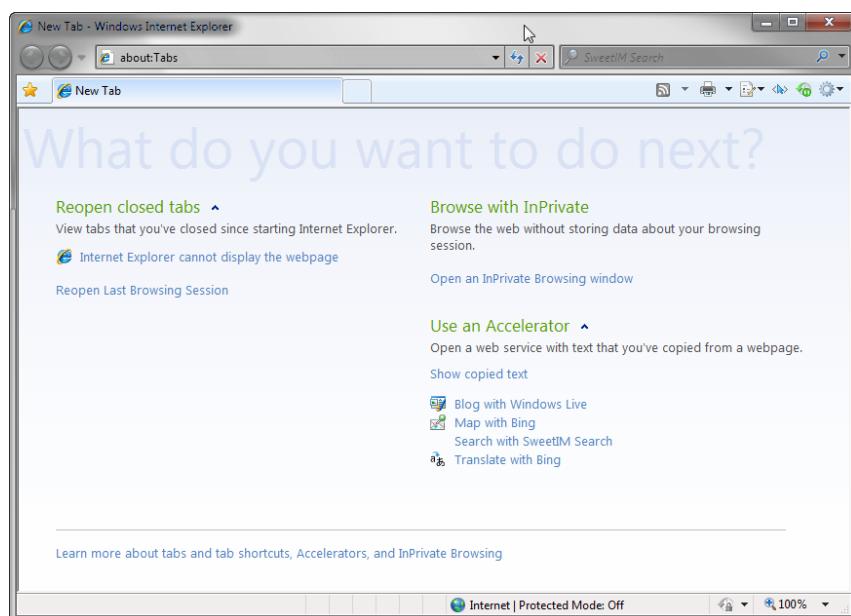


Gambar 3 Safari 5

Internet Explorer

Browser yang pasti sudah terinstall pada komputer dengan sistem operasi windows. Jika anda masih menggunakan Internet Explorer versi 6, 7 dan 8, saya anjurkan untuk mendownload versi terbaru yakni Internet Explorer 9/10 (untuk Windows Vista/7).

Penggunaan Internet Explorer dalam pengujian halaman web sangat tidak dianjurkan (terutama versi 7 ke bawah) karena dukungannya yang kurang serta proses penampilan halaman web yang terkadang menjadi berantakan. Namun karena pengguna Internet explorer di dunia masih mendominasi, mau tidak mau kita harus menggunakan Internet Explorer.



Gambar 4 Internet Explorer 8

1.2. Text Editor

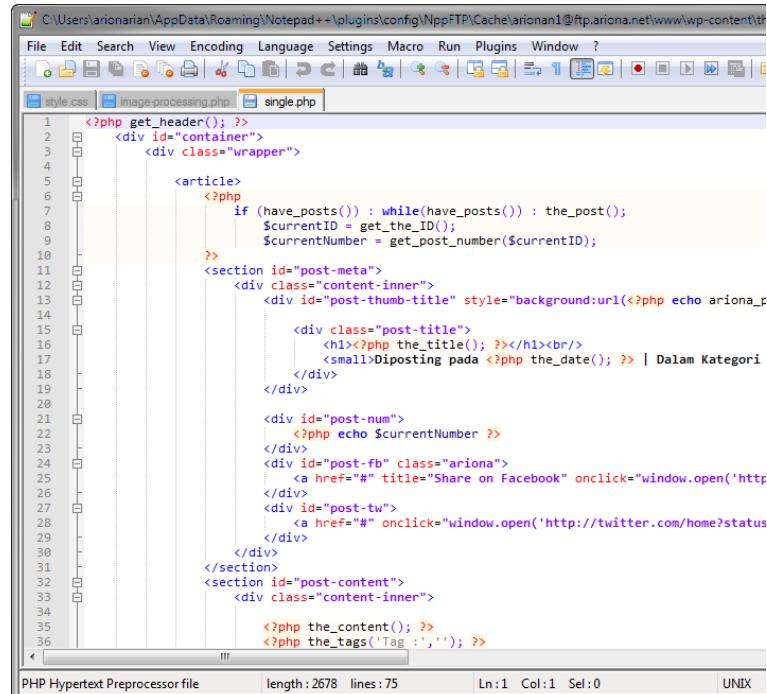
Untuk membuat halaman Web anda memerlukan sebuah text editor. Jika anda mempunyai Adobe Dreamweaver, saya sarankan untuk tidak menggunakannya, Mengingat anda masih dalam tahap belajar.

Text editor yang akan kita gunakan adalah text editor ringan namun penuh dengan fitur yang sangat membantu dalam proses *Coding(Penulisan Kode)* HTML dan CSS.

Notepad++

Tersedia untuk Windows, Notepad++ memiliki beragam fitur yang sangat mendukung para programmer. Kemampuannya untuk memperkaya diri dengan pluginpun menjadi kelebihan lainnya dengan text editor sejenisnya.

Anda dapat mendownload Notepad++ di <http://notepad-plus-plus.org>



The screenshot shows the Notepad++ interface with a PHP file named 'single.php' open. The code is a template for displaying a blog post. It includes sections for the header, container, article, post-meta, and post-content. The code uses PHP short tags (e.g., <?php) and HTML/CSS/JavaScript. The Notepad++ status bar at the bottom indicates it's a PHP Hypertext Preprocessor file, 2678 length, 75 lines, Ln:1 Col:1 Sel:0, and UNIX.

Gambar 5 Notepad++

Sublime Text

Text editor yang terbilang masih baru yang sangat mudah digunakan, penampilan simple namun enak dipandang. Sublime text adalah aplikasi berbayar tapi anda dapat mendownload versi demo-nya (meskipun versi demo tapi tidak ada batasan dalam penggunaannya).

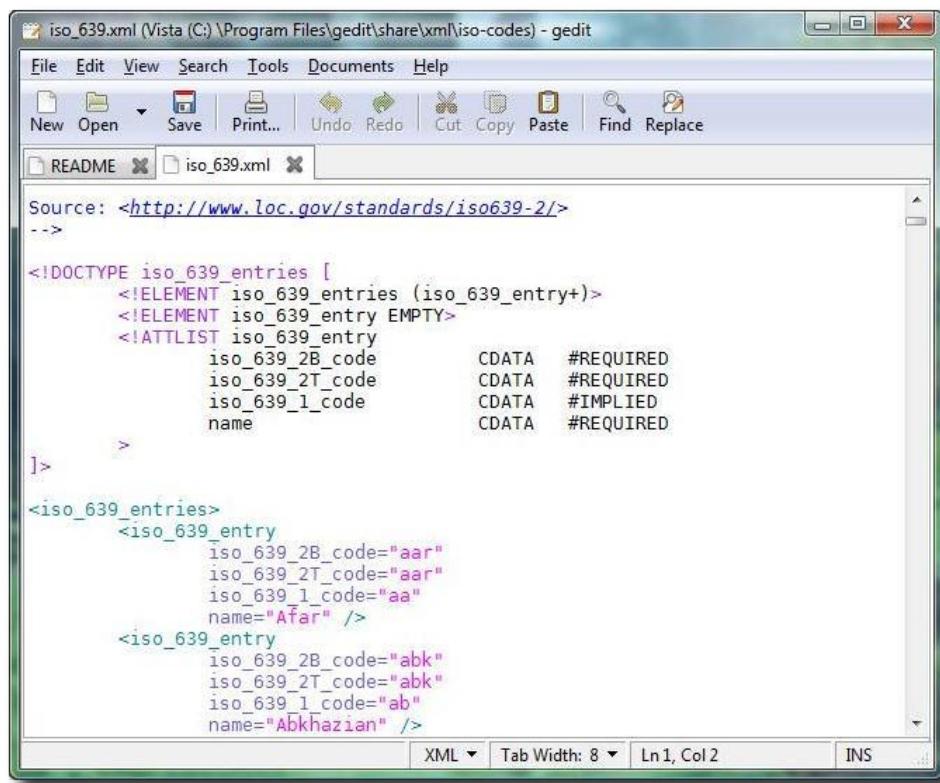
Sublime text tersedia untuk sistem Operasi Windows, Linux dan Mac, download di alamat berikut <http://www.sublimetext.com/2>

Gambar 6 Sublime Text2

Gedit

Anda pengguna Linux? Khususnya Linux dengan desktop Gnome sudah memiliki text editor bawaan, GEdit. GEdit bisa digunakan untuk menuliskan berbagai macam bahasa pemrograman.

Anda pengguna Windows-pun dapat mendownload GEdit versi windows di situs resminya <http://projects.gnome.org>.



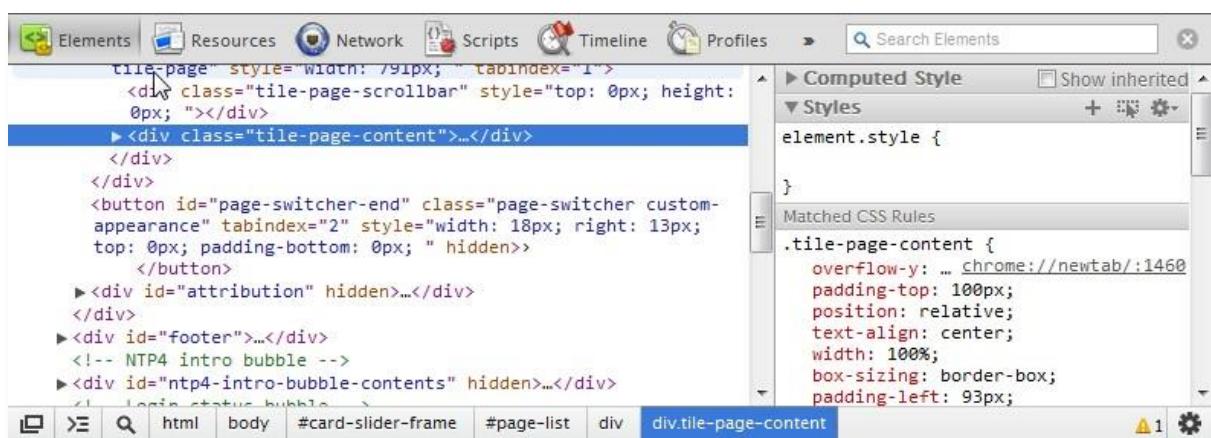
Gambar 7 GEdit untuk Windows

1.3. Add-On

Firebug

Firebug akan kita gunakan untuk memeriksa kode HTML dari setiap halaman web, melihat CSS yang digunakan dan untuk menguji script javascript. Anda dapat menginstallnya pada browser Firefox. Untuk menginstallnya cukup masuk ke halaman Add-on dan cari add-on firebug.

Jika anda menggunakan Google Chrome atau Safari, tidak usah menginstall firebug karena kedua browser dengan engine webkit ini sudah memiliki tool serupa yang dinamai dengan Developer Tool.



Gambar 9 Developer Tools pada Google Chrome

Web Developer Tools

Add-on ini digunakan untuk menguji halaman web dan melakukan beberapa perubahan/perbaikan sementara terhadap halaman web. Anda dapat menginstallnya dari add-on firefox atau dari Chrome Web Store.

Gambar 10 Web Developer Tools memiliki sejumlah tools yang sangat membantu



Itulah aplikasi-aplikasi yang diperlukan untuk mempelajari HTML dan CSS, tapi yang terpenting adalah kehadiran **Web Browser** dan **Text editor**, sisanya adalah *optional* yang keberadannya hanya mendukung aktifitas kita dalam pembangunan halaman web. Seperti Firebug dan Developer tools memang penting namun anda tidak harus menginstall keduanya. Keberadaan kedua tools tersebut sangat membantu untuk menguji dan memperbaiki kesalahan yang ditimbulkan baik kesalahan dalam pengetikan kode atau lainnya.

Bab 2

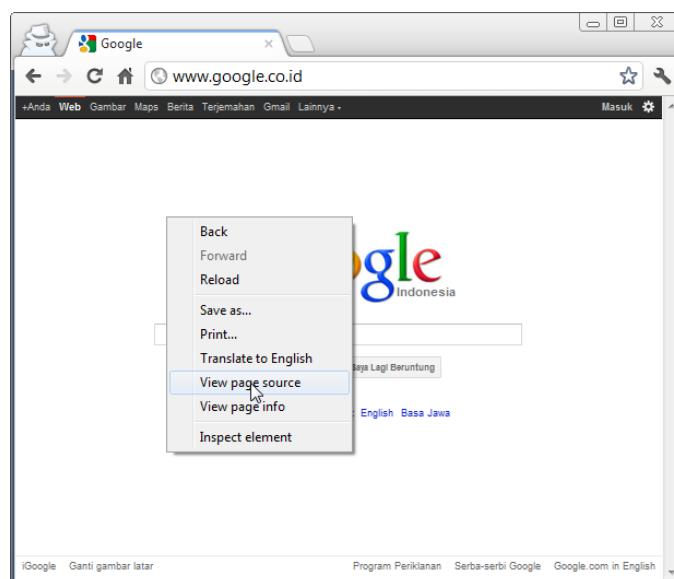
Pengenalan HTML

2.1. Apa itu HTML?

Semua halaman web yang sering anda buka, seperti facebook.com, twitter.com, google.com dan lain sebagainya ditampilkan dengan menggunakan HTML. Jadi bisa dikatakan HTML adalah bahasa dasar untuk menampilkan halaman web pada web browser.

Buka Web browser anda lalu bukalah sebuah situs, <http://www.google.com> misalnya.

1. Lalu klik kanan di sembarang area, dan pilih “**Inspect element**” (menu paling akhir)



Gambar 11 Inspect Element

2. Perhatikan kode hasil inspect element, yang terlihat adalah sebagai berikut :

Gambar 12 Inspect Element akan menampilkan kode HTML dari situs yang bersangkutan



The screenshot shows the Chrome DevTools interface with several tabs at the top: Elements, Resources, Network, Sources, Timeline, Profiles, Audits, and Console. The Elements tab is currently selected. Below the tabs, there is a large code editor window displaying the HTML code of the Google homepage. The code includes the DOCTYPE declaration, the opening tag with itemscope and itemtype attributes, the head section, and the body section which contains the Google logo and navigation links.

```
<!DOCTYPE html>
<html itemscope="itemscope" itemtype="http://schema.org/WebPage">
  <head>...</head>
  <body onload="try{if(!google.j.b){document.f&&document.f.q.focus();document.gbqf&&document.gbqf.q.focus();}}catch(e){}if(document.images)new Image().src='/images/nav_logo114.png'" dir="ltr" alink="#dd4b39" bgcolor="#fff" id="gsr" link="#12c" text="#222" vlink="#61c" style cz-shortcut-listen="true" class>
    <div id="pocs" style="display: none; position: absolute; ">...</div>
    <a href="https://www.google.co.id/>
```

Baris pertama pasti diawali dengan `<!DOCTYPE HTML>`, ini menandakan bahwa dokumen yang sedang anda buka saat ini adalah HTML. Begitu juga dengan baris kedua : `<HTML>`, kode tersebut menandakan bahkan kode-kode yang ditulis di dalamnya adalah kode HTML.

Jika anda ingin tahu kepanjangan HTML, HTML adalah kependekan dari *Hypertext Markup Language*. Artinya adalah bahasa markup (penanda) berbasis text atau bisa juga disebut sebagai *formatting language* (bahasa untuk memformat), Jadi sudah jelas bahwa HTML **bukanlah** bahasa pemrograman, melainkan bahasa markup/formatting.

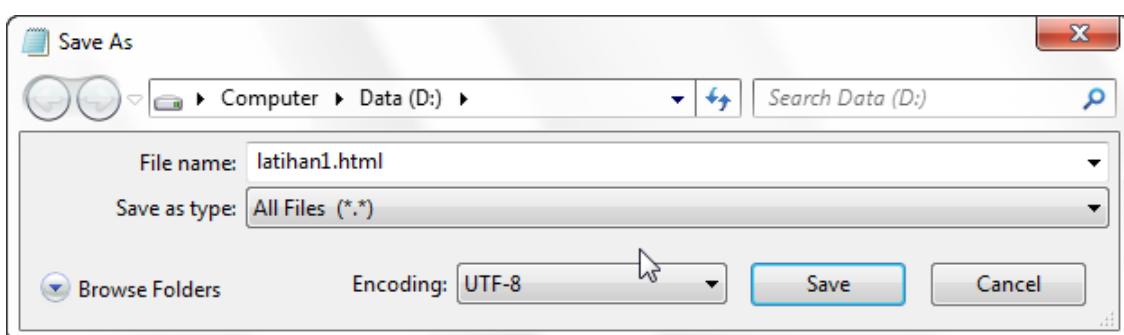
2.2. Membuat Website dalam 1 Menit

Sudah siap membuat website? Kita mulai membuat website hanya dalam 1 menit!

1. Buka Notepad (*All Programs > Accessories > Notepad*)
2. Ketikkan teks berikut :

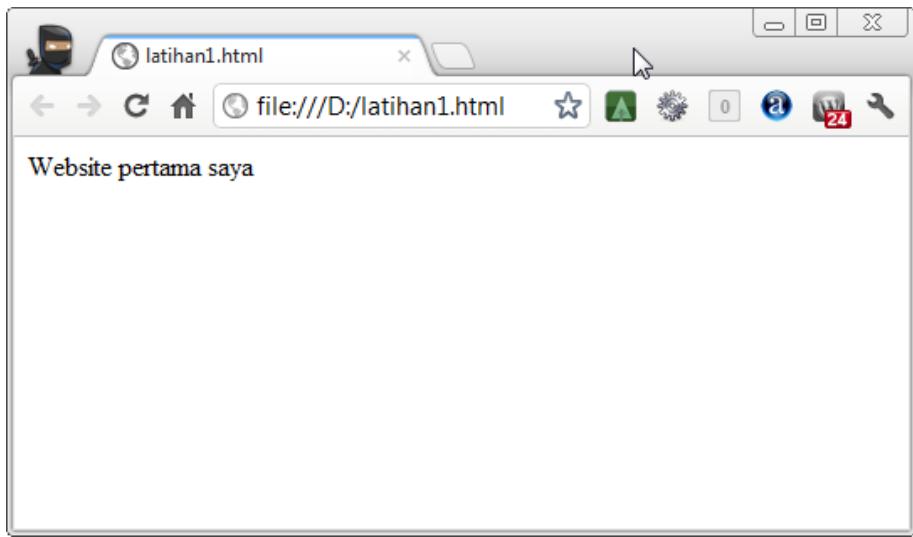
```
Website pertama saya
```

3. Pilih menu File > Save As
4. Beri nama 'latihan1.html' (tanpa tanda kutip)
5. Pada bagian Save as type, pilih *All Files (*.*)* Lalu klik tombol save.



Gambar 13 Save As type

6. Sekarang klik dua kali file HTML yang telah anda buat, (atau Klik dan Drag ke Web Browser anda, disini saya menggunakan Google Chrome)



Gambar 14 Website pertama anda

Demonstrasi di atas hanyalah langkah untuk membuat file HTML, *mudah bukan?*

2.3. Pengenalan Tag HTML

Seperti yang saya sebutkan sebelumnya, bahwa HTML adalah bahasa Markup, artinya bahasa HTML hanya digunakan untuk *me-markup* suatu dokumen. Sebagai penyederhanaan kita analogikan dengan pembuatan dokumen di Microsoft Word atau aplikasi Word Processing lainnya.

Ketika membuat sebuah dokumen word, biasanya kita melakukan *formatting* pada teks. seperti menebalkan teks, memiringkan, memberi garis bawah, Membuat Penjedulan (Heading), membuat *list* (Daftar) dan lain sebagainya. Pada HTML, *formatting* ini bisa kita sebut dengan Proses *Markup* dan akan dilakukan dengan menggunakan tag HTML.

Misalnya, untuk mem-format suatu teks menjadimiring kita tambahkan tag ``atau`<i>`. Edit file sebelumnya, menjadi seperti berikut :

```
Website <em>pertama</em> saya
```

Dan ketika dibuka kembali pada browser, anda dapat melihat kata pertama menjadi miring.

` .. ` disebut dengan Tag HTML, yang digunakan untuk *me-markup*(memformat) dokumen HTML. Perhatikan bahwa setiap tag memiliki pasangan yakni tag penutupnya.

Artinya hanya teks yang berada di dalam tag tersebut yang akan diberi format miring. Tag penutup ditandai dengan penambahan tanda slash (/) sebelum nama tagnya. Berikut ini adalah anatomi dari tag HTML :

```
<namatag atribut=,nilai Atribut>
    Isi atau Konten
</namatag>
```

Setiap tag HTML bisa menjadi “isi” dari tag HTML lainnya, untuk membuat suatu hierarki dari dokumen HTML.

Dalam setiap tag HTML bisa disisipkan berbagai atribut yang berfungsi untuk menampung informasi-informasi tertentu, misalnya atribut id dan class yang berfungsi untuk memberi nama suatu tag sebagai referensi CSS atau javascript nantinya.

Sekarang bagaimana jika teks yang dimiringkan tersebut ingin kita tebalkan? Untuk menebalan teks anda dapat menggunakan tag **** atau ****, menjadi seperti berikut:

```
Website <strong><em>pertama</em></strong> saya
```

2.4. Struktur file HTML

Suatu dokumen HTML juga mempunyai aturan dalam penulisannya, ada beberapa tag yang harus anda tuliskan dan sudah menjadi ketentuan. Jika kita perbaiki latihan kita sebelumnya menjadi dokumen HTML yang benar maka kodennya menjadi seperti berikut:

```
1.  <!DOCTYPE HTML>
2.  <HTML>
3.      <head>
4.          <title>Judul File HTML</title>
5.      </head>
6.
7.      <body>
8.          Website <strong><em>pertama</em></strong> saya
9.      </body>
10. </HTML>
```

Bingung? Tenang, saya jelaskan tiap baris kode tersebut.

```
<!DOCTYPE HTML>
```

Adalah Tag awal dari setiap dokumen HTML, tag ini berfungsi untuk menginformasikan pada browser bahwa dokumen yang sedang dibuka adalah dokumen HTML. Tag ini perlu dicantumkan disetiap dokumen HTML yang akan anda buat.

```
<HTML> ... </HTML>
```

Awal dari isi dokumen HTML dimulai dari sini, semua kode HTML yang akan anda buat akan ditulis di dalam tag ini, perhatikan juga bahwa setiap tag harus memiliki tag penutup.

```
<head> ... </head>
```

Tag Head akan digunakan untuk menyimpan berbagai informasi tentang dokumen HTML, lebih detailnya akan kita temui pada bab selanjutnya. Apa yang terdapat di dalam tag ini, tidak akan ditampilkan secara langsung pada web browser.

```
<title> ... </title>
```

Salah satu contoh informasi yang terdapat di dalam tag Head adalah title. Title akan menampilkan teks pada judul browser atau pada tab browser.

```
<body> ... </body>
```

Nah, apa yang ingin anda tampilkan pada browser akan ditulis di dalam tag ini, tag body merupakan tag pembuka dari badan dokumen HTML.

Mudah dimengerti bukan? Lalu apakah penulisan tag-tag HTML harus seperti contoh di atas? Seperti penambahan tab/indentasi pada setiap tag?. Sebenarnya Anda bisa saja menuliskan kode HTML seperti ini :

```
<!DOCTYPE      HTML><HTML><head><title>Judul      File      HTML</title></head><body>Website<i>pertama</i> saya</body></HTML>
```

Dan yang anda dapatkan dalam browserpun sama, namun anda akan **kesulitan** membaca dokumen HTML tersebut. Kalau saja dokumen HTML yang anda buat hanya

10 baris mungkin hal ini tidak akan menjadi masalah, namun bagaimana jika 1000 baris?

Nah loh?!

Untuk itu kita menambahkan indentasi, disetiap isi (konten)dari suatu tag. Misalnya:

```
<head>
    <title>Judul File HTML</title>
</head>
```

Ini menandakan bahwa tag title berada di dalam tag head. Selain itu kita juga bisa mengontrol, apakah suatu tag sudah ditutup atau belum.

Bab 3

Memuat Gambar

Setelah anda mengetahui struktur dari sebuah file HTML dan apa saja yang harus ditulis pertama kali. Selanjutnya saya akan mengajak anda untuk mempelajari cara memuat/menampilkan sebuah gambar dan memahami apa itu atribut HTML.

Untuk memuat sebuah gambar pada dokumen HTML, kita akan menggunakan tag ``. Lalu dimana kita meletakkan gambar tersebut? Di dalam tag `` kah?

3.1. Mengenal Atribut HTML

Setiap tag HTML dapat memiliki satu atau lebih atribut, atribut ini berfungsi untuk menyimpan informasi yang berkaitan dengan tag tersebut.

Pada tag `` kita akan menggunakan atribut `src` untuk menyimpan lokasi gambar :

```
<img src>
```

Selanjutnya untuk memasukkan nilai pada atribut, kita gunakan operator sama dengan (=) diikuti dengan lokasi gambar yang diapit oleh tanda kutip.

```
<img src=,gambar.jpg>
```

Tag img termasuk ke dalam tag spesial dan tidak memiliki tag penutup (Disebut juga sebagai *Self Closed Tag*), seperti tag-tag lainnya. Ini dikarenakan tag ini tidak memiliki konten seperti :

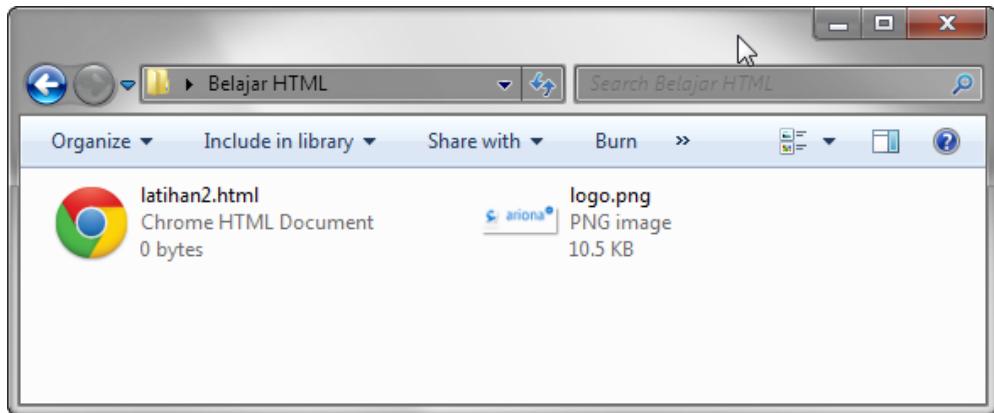
```
<title>Ini adalah Konten tag title</title>
```

Sebagai pengganti, kita tambahkan tanda slash (/) sebelum kurung tutup :

```
<img src=,gambar.jpg />
```

Sekarang saatnya anda mencoba memuat gambar pada file HTML.

1. Buatlah file HTML baru dengan nama latihan2.html
2. Carilah sebuah gambar dan simpan satu folder dengan file HTML anda.



Gambar 15 Penempatan gambar sesuai dengan lokasi file HTML

3. Ketikkanlah kode HTML berikut :

```
1. <!DOCTYPE HTML>
2. <HTML>
3.   <head>
4.     <title>Judul File HTML</title>
5.   </head>
6.
7.   <body>
8.     <p>ini adalah contoh pemuatan gambar pada file
    HTML</p>
9.     <img src=,logo.png` />
10.  </body>
11. </HTML>
```

4. Buka pada browser, anda akan melihat gambar telah termuat pada file HTML anda.



Gambar 16 Gambar ditampilkan pada browser

Setiap tag img, selain memiliki atribut src untuk menyimpan lokasi gambar, anda juga harus menyertakan atribut alt yang akan digunakan sebagai teks alternatif ketika gambar tidak berhasil dimuat atau gambar hilang.

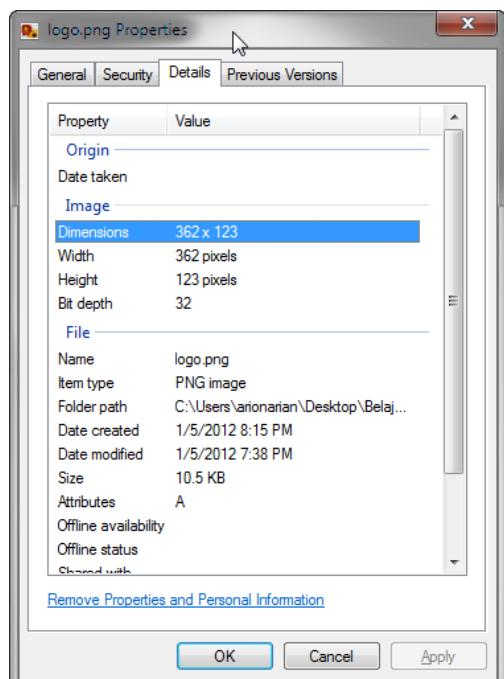
```

```

Dan atribut lainnya adalah atribut width dan height, yang berfungsi untuk menentukan lebar dan tinggi dari gambar tersebut. Memang hal ini tidak perlu dilakukan karena gambar secara otomatis akan terload dengan ukuran sebenarnya, namun hal ini sangat dianjurkan untuk mempercepat proses pemenuhan gambar.

```

```



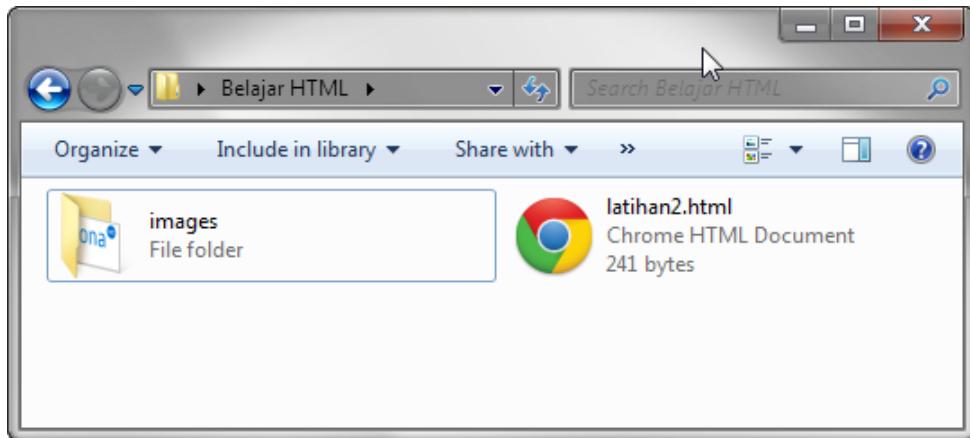
Gambar 17 File Properties gambar

Anda tidak perlu memberi tanda kutip pada nilai lebar dan panjang. Nilai ini bisa anda dapatkan dengan mengklik kanan gambar tersebut dan pilih menu properties. Pada tab details akan anda temui ukuran gambar tersebut.

Dengan penambahan atribut alt, ketika gambar yang dimaksud hilang/tidak/gagal termuat maka akan muncul sebuah icon broken file, dan isi dari atribut alt akan ditampilkan disana.

3.2. Cara penulisan lokasi file

Jika anda menyimpan gambar tersebut pada sebuah folder seperti berikut.



Gambar 18 Menyimpan gambar pada folder images

Maka yang perlu anda lakukan adalah menambahkan nama folder tersebut diikuti dengan tanda *slash* (/) dan nama file gambar yang akan dimuat:

```
<img src=,images/logo.png` alt=,Ini adalah teks alternatif` />
```

Jika dalam folder tersebut terdapat folder lagi, dan gambar yang ingin anda muat ada di dalamnya maka penulisannya menjadi seperti berikut :

```
<img src=,images/folder1/folder2/logo.png` alt=,Ini adalah teks alternatif` />
```

Selain penggunaan lokasi gambar seperti di atas, anda juga bisa menampilkan gambar yang sudah terdapat di internet, penulisan seperti ini disebut dengan *hotlinking* misalnya:

```
<img src=,http://arionna.net/images/gambar.jpg` alt=,logo` />
```

Penggunaan hotlinking sangat tidak dianjurkan karena gambar yang dimuat bukanlah milik kita dan tentunya juga akan merugikan si pemilik gambar.

Bab 4

Membuat Link

Yang sering anda temui dalam sebuah website adalah link. Link ini akan membuat konten atau *elemen HTML*⁴ dapat di klik dan akan mengarahkan/membawa anda ke halaman web lainnya. Biasanya suatu link ditampilkan berwarna biru dan bergaris bawah (selama belum diberi style).

4.1. Link Standar

Link atau biasa disebut dengan anchor (pengait) dapat dibuat dengan menambahkan tag `<a>` pada teks yang ingin kita buat menjadi link.

Klik `<a>disini` untuk mendownload

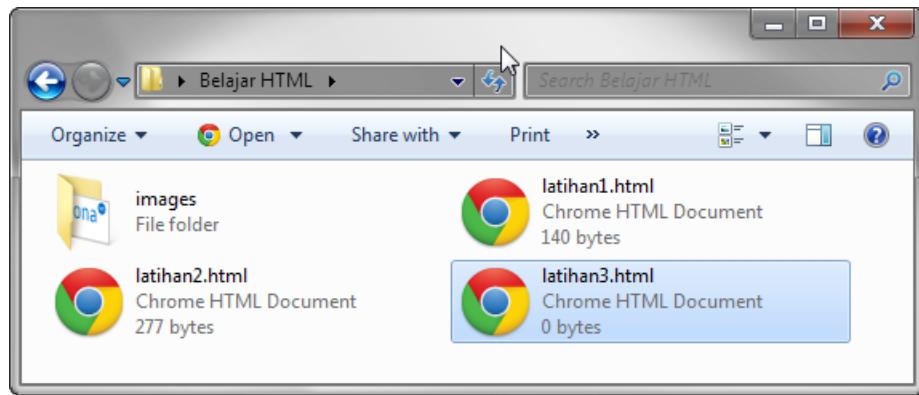
Namun anda tidak akan melihat perubahan pada teks “disini” karena kita belum “mengaitkannya” ke halaman web lain. Untuk itu kita akan menggunakan attribut `href` untuk menyimpan alamat web yang akan dituju ketika link di klik (penulisan lokasi sama halnya dengan attribut `src` pada tag `img`).

1. Klik `disini` untuk mendownload

Sebagai latihan kita akan membuat file HTML baru dan membuat link untuk mengaitkannya ke latihan-latihan sebelumnya.

1. Buatlah file baru dengan nama `latihan3.html`, simpanlah di folder yang sama dengan latihan-latihan sebelumnya.

⁴ Tag-tag HTML juga biasa sebut dengan elemen.



Gambar 19 Penempatan file pada folder yang sama

2. Ketikkan kode berikut pada file latihan3.html

```
1. <!DOCTYPE HTML>
2. <HTML>
3.   <head>
4.     <title>Judul File HTML</title>
5.   </head>
6.   <body>
7.     <h1>Selamat datang di website saya</h1>
8.     <a href=,latihan1.html>buka Latihan 1</a><br>
9.     <a href=,latihan2.html>buka Latihan 2</a>
10.    </body>
11. </HTML>
```

3. Buka kembali file **latihan1.html** dan **latihan2.html**, dan tambahkan sebuah link yang akan mengait ke file latihan3.html tepat sebelum penutup tag body (`</body>`).

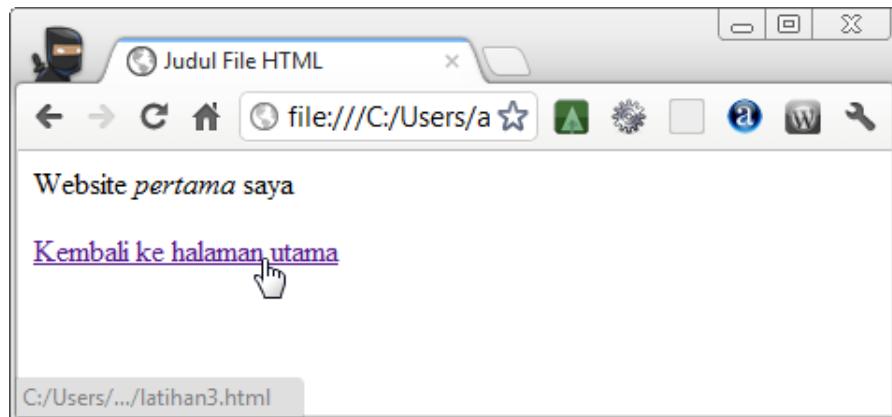
```
...
<a href=,latihan3.html>Kembali ke halaman utama <a>
</body>
</HTML>
```

4. Save pekerjaan anda, dan bukalah file latihan3.html pada browser. Anda dapat mengklik linknya satu persatu.



Gambar 20 tampilan file Latihan3.html

Ketika link “buka latihan 1” di klik, browser akan menampilkan file latihan1.html

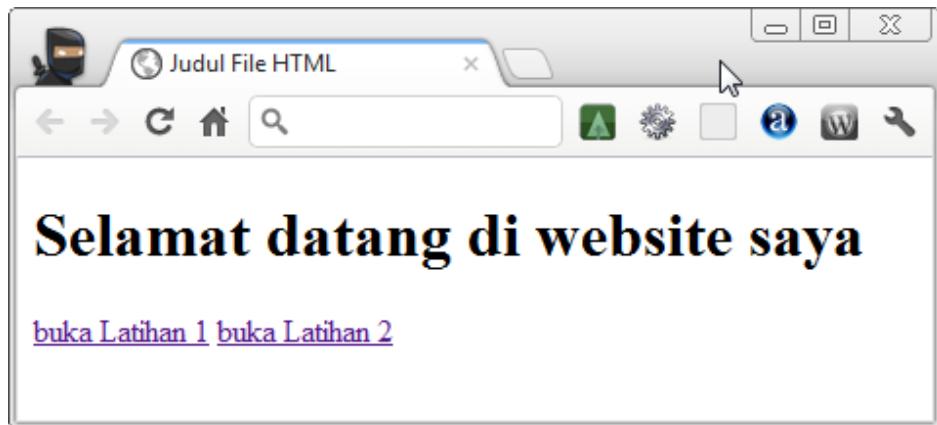


Gambar 21 Link untuk kembali ke halaman utama

Dan Klik link “Kembali ke halaman utama” untuk menampilkan latihan3.html kembali.

Jika anda perhatikan pada kode yang terdapat dalam latihan3.html, ada beberapa tag yang belum saya jelaskan, yaitu tag <h1> dan
.

Tag
 digunakan untuk memindahkan teks kebaris baru, sehingga teks yang dipisahkan dengan tag ini akan ditampilkan di baris yang berbeda. Jika anda menghapus tag
 ini, maka teks akan ditampilkan satu baris.

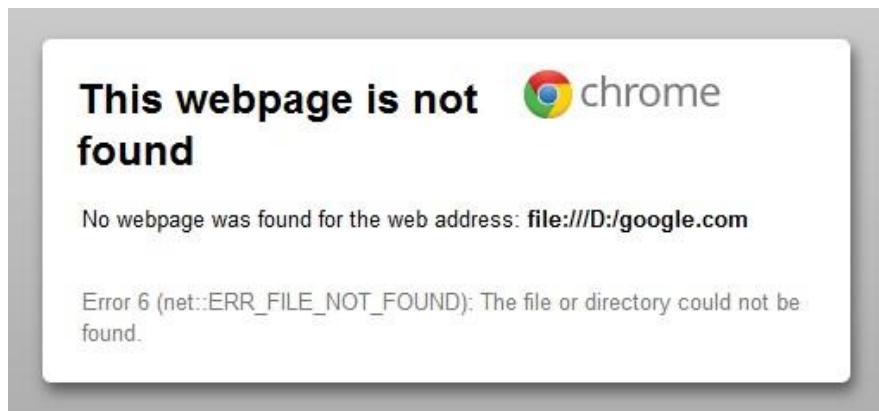


Gambar 22 link ditampilkan satu baris tanpa tag

Untuk membuat link yang mengacu ke halaman web lain, pastikan anda menyertakan http:// pada atribut href link tersebut.

```
<a href="http://www.google.com"> Buka Google </a>
```

Jika anda tidak menyertakan http:// maka link tersebut tidak akan bekerja sebagaimana mestinya.



Gambar 23 Error ketika http:// tidak ditambahkan pada alamat web lain

Setiap link yang diklik akan ditampilkan di window/tab yang sama, bagaimana jika anda ingin membuka link tersebut di tab/window baru? Tambahkan atribut target="_blank".

```
<a href="http://arionna.net" target="_blank">arionna.net</a>
```

4.2. Link Email

Anda juga dapat membuat link email, link ini berisi alamat email yang ketika diklik, aplikasi untuk mengirim email akan otomatis terbuka dan tujuan email secara otomatis terisi dengan alamat yang telah ditentukan.

Untuk membuat link email, anda tinggal menambahkan `mailto:alamat@email` di dalam atribut href.

```
<a href="mailto:mrxyz@yahoo.com">Kirim Email</a>
```

Selain tipe link-link di atas, masih banyak lagi tipe link lainnya, seperti link No. Telp, Aplikasi dan lainnya.

Bab 5

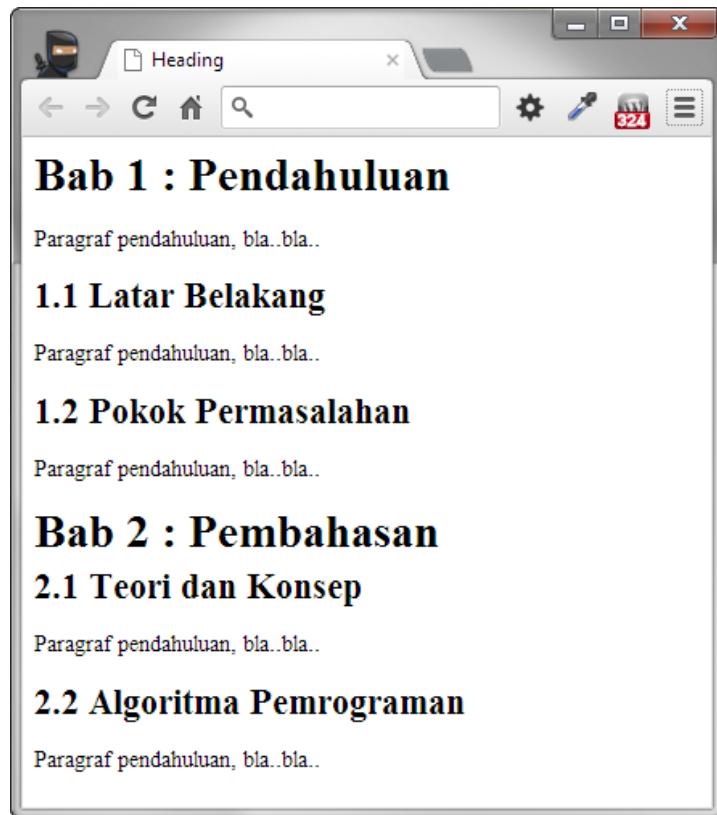
Heading/Penjudulan

Heading, digunakan untuk memberikan penjaduluan pada suatu dokumen HTML. Bayangkanlah sebuah skripsi atau buku yang memiliki bab dan subbab-subbab di dalamnya. Untuk memformat penjaduluan dalam HTML, kita gunakan tag `<h1>` untuk judul utama dan untuk judul subbabnya anda dapat menggunakan tag `<h2>` sampai dengan`<h6>`.

Setiap level judul memiliki ukuran huruf yang berbeda-beda (namun anda masih bisa merubah ukurannya melalui CSS).

Sebagai latihan, buatlah file HTML baru dengan nama `latihan4.html` lalu ketikkan kode HTML berikut:

```
1. <!DOCTYPE HTML>
2. <HTML lang=,en-US>
3.   <head>
4.     <title>Heading</title>
5.   </head>
6.   <body>
7.     <h1>Bab 1 : Pendahuluan</h1>
8.     <h2>1.1 Latar Belakang</h2>
9.     <p>Paragraf pendahuluan, bla..bla.. </p>
10.    <h2>1.2 Pokok Permasalahan</h2>
11.    <p>Paragraf pendahuluan, bla..bla.. </p>
12.    <h1>Bab 2 : Pembahasan</h1>
13.    <h2>2.1 Teori dan Konsep</h2>
14.    <p>Paragraf pendahuluan, bla..bla.. </p>
15.    <h2>2.2 Algoritma Pemrograman</h2>
16.    <p>Paragraf pendahuluan, bla..bla.. </p>
17.  </body>
18. </HTML>
```



Gambar 24 Penjudulan pada dokumen HTML

Adanya penjudulan dimaksudkan agar suatu dokumen HTML lebih terstruktur layaknya sebuah dokumen resmi seperti skripsi /paper yang mengharuskan adanya perbedaan antara Bab utama dan sub-babnya.

Perhatikan contoh penggunaan Heading dalam suatu website di halaman selanjutnya.

Dalam website tersebut, Logo/Judul dari website diberi Heading Level 1, untuk judul artikel diberi heading level 2 dan judul dari bagian-bagian di dalamnya diberi heading level 3 s.d 4

Bab 6

Membuat daftar/list

Sebuah daftar dapat memiliki penomoran atau hanya sebuah simbol berupa lingkaran atau bentuk lainnya. Dalam HTML, daftar yang menggunakan penomoran disebut dengan *ordered list* dan yang menggunakan simbol disebut dengan *unordered list*.

6.1. Ordered List

Ordered list atau Daftar berurutan dapat dibuat dengan menggunakan tag `` (singkatan dari ordered list) dan untuk setiap listnya kita gunakan tag `` (singkatan dari list item/item daftar). Sebagai contoh, perhatikan kode HTML berikut :

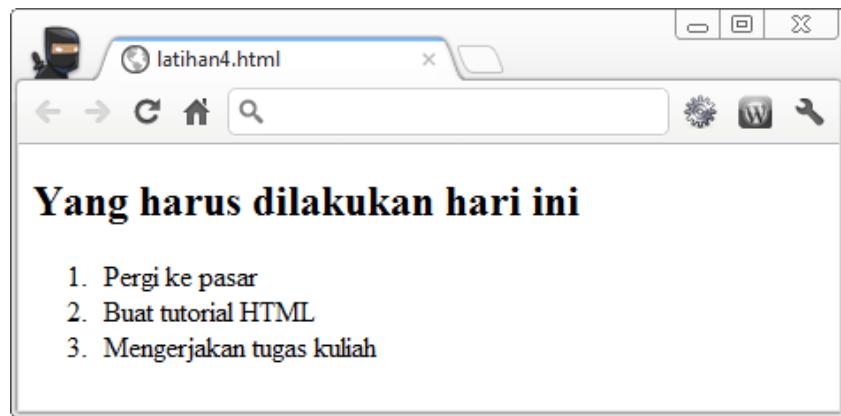
```
1. <h2>Yang harus dilakukan hari ini</h2>
2. <ol>
3.     <li>Pergi ke pasar</li>
4.     <li>Buat tutorial HTML</li>
5.     <li>Mengerjakan tugas kuliah</li>
6. </ol>
```

Penomoran daftar akan dilakukan secara otomatis ketika anda menambahkan daftar item.



Gunakan Ordered list, ketika sebuah list/daftar memerlukan urutan yang benar, seperti cara menyalaikan motor dan lain sebagainya.

Dan jika ditampilkan pada browser, maka yang akan anda dapatkan adalah seperti berikut :



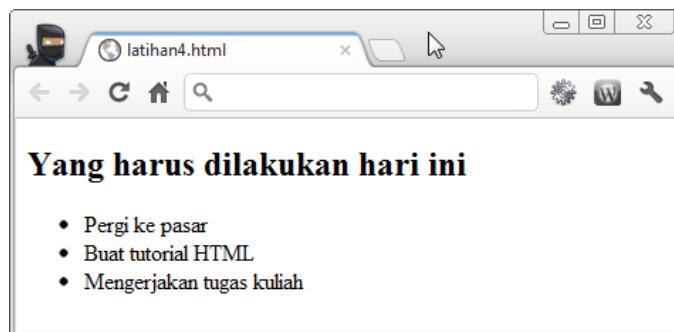
Gambar 25 Daftar berurutan

6.2. Unordered List

Berbeda dengan daftar berurut, *unordered list* akan menandai setiap item dengan simbol. Baik berupa lingkaran atau persegi (anda masih bisa merubahnya dengan CSS). Untuk membuat daftar tidak berurut kita gunakan tag `` dan seperti tag ``, item yang terdapat di dalamnya harus diapit dengan tag ``.

Jika kita modifikasi contoh sebelumnya dengan merubah `` menjadi `` maka yang akan kita dapat adalah seperti berikut :

```
1. <h2>Yang harus dilakukan hari ini</h2>
2. <ul>
3.     <li>Pergi ke pasar</li>
4.     <li>Buat tutorial HTML</li>
5.     <li>Mengerjakan tugas kuliah</li>
6. </ul>
```



Gambar 26 Daftar tidak berurutan

Secara default, item akan ditandai dengan lingkaran.

6.3. Definition List

Berbeda dengan Ordered List dan Unordered List yang memiliki struktur sama, Data list memiliki struktur tersendiri. Biasanya data list ini digunakan untuk membuat daftar istilah beserta definisinya seperti halnya dalam kamus.

```
<dl>
  <dt>Test</dt>
  <dd>Test</dd>
</dl>
```

Tag `dt` (definition term) digunakan untuk menampung istilah yang akan didefinisikan, dan tag `dd` digunakan untuk menuliskan definisi dari `dt` sebelumnya. Berikut contoh penggunaan dari Definition List :

```
<dl>
  <dt>Manga</dt>
  <dd>Manga (baca: man-ga, atau ma-ng-ga) merupakan kata komik dalam bahasa Jepang; di luar Jepang, kata tersebut digunakan khusus untuk membicarakan tentang komik Jepang. </dd>

  <dt>Mangaka</dt>
  <dd>Mangaka (baca: man-ga-ka, atau ma-ng-ga-ka) adalah orang yang menggambar manga</dd>
</dl>
```

6.4. List/Daftar bersarang

Sebuah daftar bisa saja memiliki daftar lagi di dalam itemnya, atau biasa kita sebut dengan daftar/list bersarang (*nested list*). Contohnya seperti pada latihan yang akan kita buat berikut.

Buat file baru dengan nama file latihan5.html lalu ketikkan kode HTML berikut

```
1. <!DOCTYPE HTML>
2. <HTML>
3. <head>
4.   <title>List/Daftar</title>
5. </head>
6. <body>
7.   <h2>Yang harus dilakukan hari ini</h2>
8.   <ol>
9.     <li>Pergi ke pasar
10.    <ul>
11.      <li>Beli sayuran</li>
12.      <li>Beli kertas
13.        <ol>
14.          <li>A4</li>
15.          <li>Legal</li>
```

```
16.                                     </ol>
17.                                         </li>
18.                                         <li>Beli tinta</li>
19.                                     </ul>
20.                                 </li>
21.                                 <li>Buat tutorial HTML</li>
22.                                 <li>Mengerjakan tugas kuliah</li>
23.                             </ol>
24. </body>
25. </HTML>
```

Dan berikut tampilan pada browser.



Gambar 27 List bersarang

Yang perlu anda perhatikan adalah di mana anda meletakkan tag `` atau `` jika anda ingin menempatkannya pada suatu item, yaitu sebelum penutup tag list item (``).

Daftar seperti ini biasa digunakan untuk pembuatan menu website, atau keperluan lainnya yang memang membutuhkan penomoran atau sesuatu yang memiliki poin-poin.

Bab 7

Memahami Tag div (Division)

Anda dapat membuat suatu group dari tag-tag HTML yang anda buat dengan menggunakan tag <div>, seperti pada contoh kasus berikut :

Setiap website/weblog yang anda buka, jika diperhatikan pasti memiliki empat bagian/group. Keempat bagian/group ini antara lain :

1. Header

Bagian kepala website yang berisi logo website, nama website, slogan website, menu website dan lainnya.

2. Content

Berisi isi dari website itu sendiri, jika website berupa weblog, maka *contentnya* berisi postingan terbaru atau informasi lainnya.

3. Sidebar

Berada di samping Content dan biasanya berisi iklan, kategori artikel dan widget atau hiasan situs lainnya.

4. Footer

Bagian kaki dari website yang pada umumnya diisi dengan tag website tersebut, misalnya *copyright © 2012 by someone*.

Lebih sederhananya, bayangkanlah Surat resmi yang memiliki Kop surat, badan surat dan penutup surat.

Dalam HTML, kita dapat membagi bagian-bagian tersebut dengan menggunakan tag <div> (division/bagian) dan untuk menamai setiap div kita gunakan atribut id atau class⁵. Berikut contoh penggunaan tag div jika kita terapkan pada bagian-bagian website yang telah kita sebutkan sebelumnya:

⁵ Atribut ID dan Class tidak hanya digunakan dalam tag div saja melainkan dapat digunakan di setiap tag HTML untuk memberikan penamaan sebagai referensi jika diperlukan.

```
1. <div id=,header>
2. </div>
3.
4. <div id=,content>
5. </div>
6.
7. <div id=,sidebar>
8. </div>
9.
10. <div id=,footer>
11. </div>
```

Jika anda menampilkannya pada browser, anda tidak akan melihat apa-apa karena tag div tidak akan menampilkan efek visual. Tag ini hanya digunakan untuk membuat dokumen HTML lebih terstruktur dengan membagi-bagi dokumen ke dalam bagian-bagian yang lebih spesifik.

7.1. Pakai ID atau Class?

Bagi seorang pemula, termasuk saya ketika masih belajar HTML & CSS pasti menanyakan hal yang sama. Pasalnya atribut ID dan Class digunakan untuk menamai tag HTML. Lalu apa perbedaan dari keduanya?

ID

Atribut ID digunakan untuk penamaan elemen HTML yang memiliki karakteristik unik/berbeda. Tidak boleh ada dua atau lebih elemen yang mempunyai ID yang sama.

Perhatikan contoh berikut :

```
1. <div id=,menu>
2. <ul id=,menu>
3.     <li>Beranda</li>
4.     <li>Tutorial</li>
5. </ul>
6. </div>
```

Penggunaan atribut ID pada contoh di atas adalah **SALAH**, karena terdapat dua ID yang sama, yaitu “menu” pada tag `<div>` dan ``.

Perhatikan contoh sebelumnya, bahwa kita menggunakan ID yang berbeda untuk setiap div yakni, header, content, sidebar dan footer karena semuanya memiliki struktur dan fungsi yang berbeda dalam suatu dokumen HTML.

Class

Class digunakan untuk penamaan elemen yang memiliki karakteristik/struktur sama dan dapat digunakan berulang kali dalam markup (Kode HTML). Sebagai contoh, perhatikan kode HTML berikut :

```
1. <ul id=,menu>
2.     <li class=,merah>Beranda</li>
3.     <li>Tutorial</li>
4.     <li class=,merah>Berita</li>
5.     <li>Video</li>
6. </ul>
```

Pada Kode HTML di atas, saya menggunakan class Merah pada beberapa list item, karena nantinya list item yang memiliki class merah akan diberi warna background merah.

Kesimpulannya adalah, ketika anda memiliki beberapa elemen dengan karakter/format yang sama, gunakan Class sebagai penamaannya dan Gunakan ID untuk elemen yang berbeda dan membutuhkan tanda pengenal lebih spesifik.

7.2. Memahami hubungan Child, Parent dan Siblings

Ketika suatu tag memiliki tag/konten didalamnya maka hubungan antar tag-tag tersebut disebut dengan child-parent dan sibling. Perhatikan contoh berikut :

```
1. <div id=,wrapper>
2.     <div id=,content>
3.         <div id=,article></div>
4.         <div id=,sidebar></div>
5.     </div>
6. </div>
```

Dalam contoh kode HTML tersebut :

- ✓ <div id=,wrapper> disebut **Parent**, dan tag HTML yang ada di dalamnya disebut dengan **Child**(<div id=,content>).
- ✓ Begitu juga dengan <div id=,content> disebut **parent** untuk <div id=,article> dan <div id=,sidebar>,
- ✓ dan hubungan antara<div id=,article> dan <div id=,sidebar> disebut dengan **sibling**, atau saudara.

Bab 8

Form

Dalam sebuah website biasanya terdapat satu atau lebih form, seperti form pencarian, registrasi dan lain sebagainya. Form ini biasa digunakan untuk mengumpulkan data dari pengunjung website.

Sebuah form, boleh jadi memiliki beragam kontrol, mulai dari text input, Combo box, Button dan lain sebagainya. Kita akan mempelajari sebagian kontrol yang disebutkan diatas karena kontrol di atas adalah yang paling sering digunakan dalam halaman web.

Pertama-tama, pembuatan sebuah form diawali dengan tag <form>, dan setiap kontrol-kontrol yang dibutuhkan ditempatkan di dalam tag ini.

```
<form>
  <h1>Formulir Pendaftaran</h1>
  ...
</form>
```

The screenshot shows the Yahoo! Indonesia sign-up page. At the top is the Yahoo! logo. Below it is a promotional message: "Dengan Account Yahoo!, dapatkan email gratis dan layanan web terkemuka lainnya." The form itself has several fields: "Nama Saya" with "Nama Depan" and "Nama Belakang" sub-fields; "Jenis Kelamin" with a dropdown menu showing "- Pilih Satu -"; "Tanggal Lahir" with "Tanggal", "Pilih Bulan", and "Tahun" dropdowns; and "Saya tinggal di" with a dropdown menu showing "Indonesia". Below the form is a section titled "Pilih ID dan kata sandi" with fields for "ID Yahoo! dan Email" (containing "test@yahoo.co.id" with an '@' symbol), "Kata Sandi", and "Ketik ulang Kata Sandi". To the right of these fields is a "Periksa" button and a "Kekuatan Kata Sandi" link with a password strength meter. At the bottom is a section titled "Jika Anda lupa kata sandi atau ID Anda..." with two sets of "Pertanyaan Rahasia" and "Jawaban Anda" fields.

Gambar 28 Formulir (FORM) pendaftaran Yahoo!



*ika anda merasa asing dengan istilah **FORM**, bayangkan saja sebuah **FORMULIR** yang sering ditemui di sekitar kita seperti formulir pendaftaran siswa baru, formulir pembuatan KTP dan lain sebagainya.*

Control-control Form

Setiap control pada form dapat dibuat dengan menggunakan tag <input>. Dan yang membedakan tipe dari control tersebut berada pada atribut type (tipe kontrol). Berikut ini adalah sebagian tipe kontrol yang biasa ditemui :

Label

Label digunakan untuk memberikan keterangan pada setiap input yang ada. Jika anda perhatikan formulir pendaftaran situs yahoo, yang disebut dengan label adalah yang ditandai berikut ini :

The screenshot shows a registration form with several input fields. On the left, there is a vertical list of labels: "Nama Saya", "Jenis Kelamin", "Tanggal Lahir", and "Saya tinggal di". The "Nama Saya" label is highlighted with a blue border. To the right of these labels are the corresponding input fields: "Nama Depan" and "Nama Belakang" (two separate text input boxes), a dropdown menu labeled "- Pilih Satu -", and a date input field consisting of three dropdown menus for "Tanggal", "Pilih Bulan", and "Tahun". Below the form, a blue bracket labeled "Label" points to the "Nama Saya" label.

Gambar 29 Contoh Label pada form

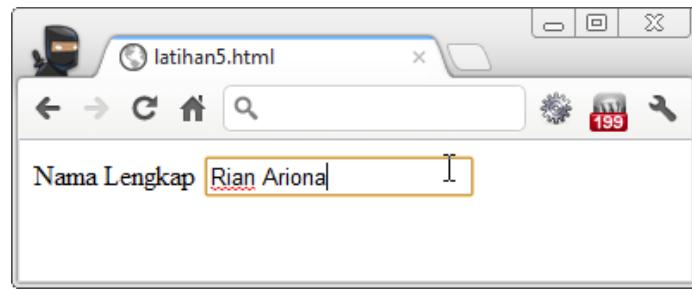
```
<label for="Name"> Keterangan Input </label>
```

Atribut for diisi dengan isi dari atribut name pada kontrol yang ingin diberi label.

Text

Control input ini dapat diisi dengan teks yang memiliki kata yang tidak terlalu panjang/hanya satu baris, biasa digunakan dalam form pencarian, nama dan lain sebagainya.

```
<label for="nama">Nama Lengkap</label>
<input type="text" name="nama" />
```



Gambar 30 Text Input

Jika text input yang akan ditampilkan ingin memiliki nilai, maka tuliskan nilai tersebut di dalam atribut value.

```
<input type='text' name='nama' value='Rian Ariona' />
```

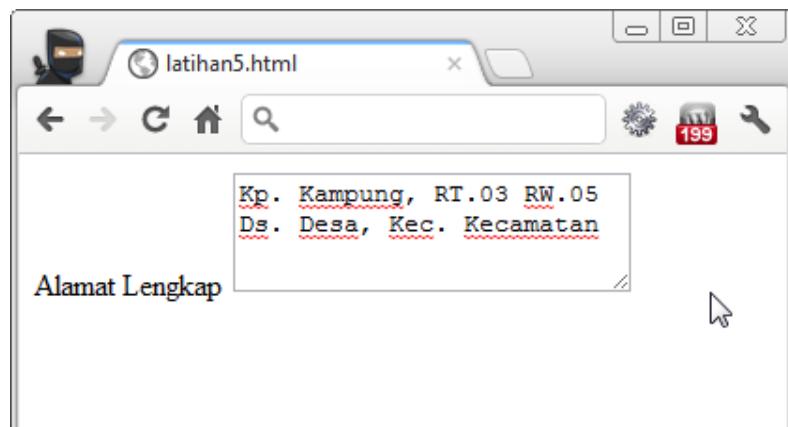
Dengan begitu, ketika halaman dibuka, text input ini secara otomatis akan terisi dengan nilai dari atribut value.

Text Area

Sama halnya dengan Input Text, namun textarea dapat diisi lebih dari satu baris, cocok digunakan untuk isian yang panjang, seperti alamat, deskripsi, atau biodata.

Berbeda dengan kontrol lainnya yang menggunakan tag <input>, text area memiliki tag sendiri yaitu <textarea></textarea>. Dan apa yang terdapat di dalam tag ini adalah value dari kontrol tersebut

```
<label for='alamat'>Alamat Lengkap</label>
<textarea name='alamat'></textarea>
```

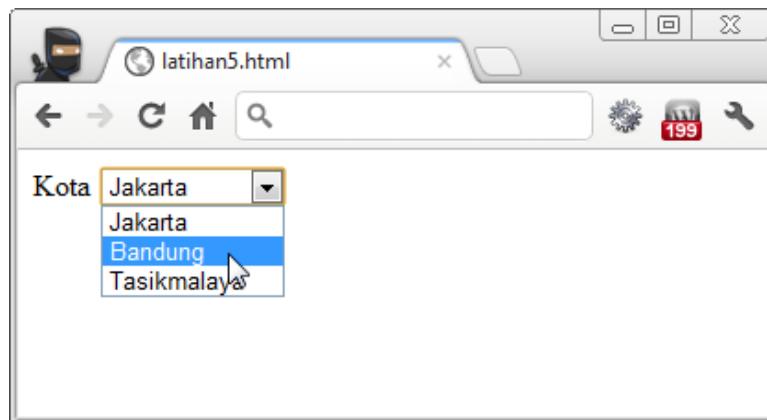


Gambar 31 Text area

Combo Box

Combo Box adalah kontrol yang memiliki pilihan ketika diklik. Pembuatannya sama seperti pembuatan Daftar/List namun dengan tag yang berbeda.

```
<label for=,kota>Kota</label>  
  
<select name=,kota>  
    <option>Jakarta</option>  
    <option>Bandung</option>  
    <option>Tasikmalaya</option>  
</select>
```

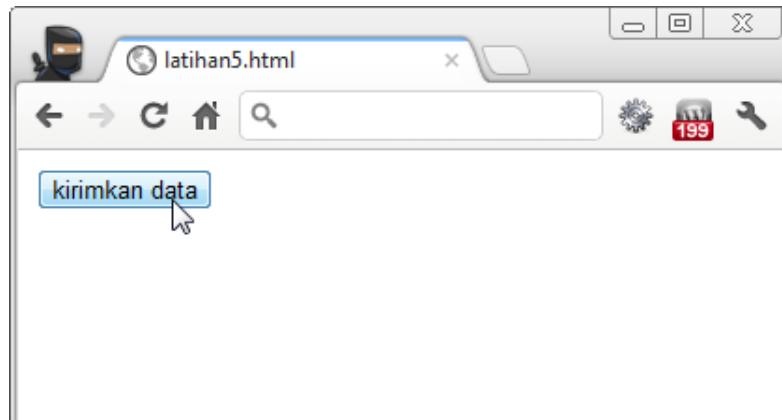


Gambar 32 Combo Box

Submit/Button

Submit atau Button, berupa tombol yang dapat di klik. Penggunaan atribut value pada kontrol ini akan merubah text yang ada di dalamnya.

```
<input type=,submit' value=, kirimkan data' />
```



Gambar 33 Button

Sebagai latihan, kita coba menggabungkan kontrol-kontrol yang telah anda pelajari sebelumnya menjadi satu form utuh.

Buatlah file baru, dengan nama file latihan6.html, lalu ketikkan kode HTML berikut pada file latihan6.html

```
1. <!DOCTYPE HTML>
2. <HTML>
3. <head>
4.     <title>Form</title>
5. </head>
6. <body>
7.     <form>
8.         <label for=,nama>Nama</label>
9.         <input type=,text' name=,nama'><br>
10.        <label for=,alamat>Alamat</label>
11.        <textarea name=,alamat'></textarea><br>
12.
13.        <label for=,kota>Kota</label>
14.        <select name=,kota'>
15.            <option>Jakarta</option>
16.            <option>Bandung</option>
17.            <option>Tasikmalaya</option>
18.        </select><br>
19.
20.        <input type=,submit' value=,kirimkan data'>
21.    </form>
22. </body>
23. </HTML>
```

Berikut ini adalah tampilan yang anda dapatkan jika file latihan6.html dibuka di dalam browser.



Gambar 34 Form utuh

Dalam buku ini, saya tidak akan membahas bagaimana cara memroses data yang dikumpulkan dari form tersebut, karena dalam HTML kita tidak bisa melakukan proses pengolahan data. Ingat bahwa HTML hanya digunakan untuk memformat dokumen, bukan untuk memroses data.

Untuk memroses data, kita bisa menggunakan bahasa pemrograman web, seperti Javascript atau PHP atau bahasa pemrograman web lainnya.

Dengan bahasa pemrograman seperti Javascript atau PHP anda dapat mengolah data-data yang dikumpulkan dari Form HTML untuk diproses dan dikeluarkan kembali menjadi suatu informasi baru.

Bab 9

Tabel

Untuk menampilkan data dengan tipe tabel dalam HTML, kita bisa menggunakan tag `<table>`.

```
<table></table>
```

Untuk membuat baris tabel, kita gunakan tag `<tr>` – singkatan dari *table row* (ditulis di dalam tag table)

```
<table>
  <tr></tr>
</table>
```

Sedangkan untuk menentukan banyaknya kolom, tergantung dari banyaknya tag `<td>` (*table data*) yang digunakan di dalam tag `<tr>`.

```
1. <table>
2.   <tr>
3.     <td>No</td>
4.     <td>Nama</td>
5.     <td>Alamat</td>
6.   </tr>
7. </table>
```

Untuk menyatukan kolom atau baris (*merge-cell*) kita tambahkan atribut colspan (untuk *merge horizontally*-secara horizontal) atau rowspan (untuk *merge vertically*).

```
1. <table>
2.   <tr>
3.     <td rowspan=2>No</td>
4.     <td colspan=2>Jenis Kelamin</td>
5.     <td rowspan=2>Nama</td>
6.   </tr>
7.   <tr>
8.     <td>L</td>
9.     <td>P</td>
10.  </tr>
11.  <tr>
12.    <td>1</td>
13.    <td>L</td>
14.    <td></td>
```

```
15.           <td>Rian</td>
16.     </tr>
17. </table>
```

Jika anda tampilkan dalam browser, maka tabel tersebut tidak akan memiliki garis, untuk memberi garis pada tabel anda dapat menambahkan atribut border=1 pada tag `<table>` atau anda dapat menambahkan border lewat CSS nantinya.

No	Jenis Kelamin		Nama
	L	P	
1	L		Rian

Gambar 35 Tabel setelah pemberian atribut border

Sebagai latihan, anda tulis kode tabel di atas dan tambahkan beberapa data lagi, lalu simpan dalam file latihan7.html.

9.1. Struktur tabel yang dianjurkan

Agar format tabel sesuai dengan ketentuan, Anda dapat menggunakan tag `<thead>` untuk baris yang menjadi Judul tabel, `<tbody>` untuk data/isi dari tabel dan `<tfooter>` untuk kaki tabel (bisa diisi sama dengan thead atau lainnya).

```
1. <table border=1>
2.   <thead>
3.     <tr>
4.       <td rowspan=2>No</td>
5.       <td colspan=2>Jenis Kelamin</td>
6.       <td rowspan=2>Nama</td>
7.     </tr>
8.     <tr>
9.       <td>L</td>
10.      <td>P</td>
11.    </tr>
12.  </thead>
13.
14.  <tbody>
15.    <tr>
16.      <td>1</td>
17.      <td>L</td>
18.      <td></td>
19.      <td>Rian</td>
20.    </tr>
21.  </tbody>
22. </table>
```

9.2. Kesalahan dalam penggunaan <table>

Tag table digunakan untuk memformat data tabular/data yang membutuhkan format tabel dalam menampilkannya, seperti contoh penggunaan tabel untuk menampilkan data nilai berikut :

No	Nama	Nilai Akhir	Index
1	Jono	90	A
2	Joni	75	B
3	Dedi	82	A

Masih banyak yang menggunakan tabel untuk keperluan layouting sebuah website, misalnya menggunakan tag table untuk membuat layout 2 kolom, 3 kolom, atau 4 kolom seperti contoh berikut :

Website Keren Saya		
- Home - Artikel - About	Selamat Datang! Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna. Nunc viverra imperdiet enim. Fusce est.	Address Jl. Sasab 34 4023XX
Copyright © 2012 by Saya		

Gambar 36 Penggunaan table untuk layout adalah menyalahi aturan

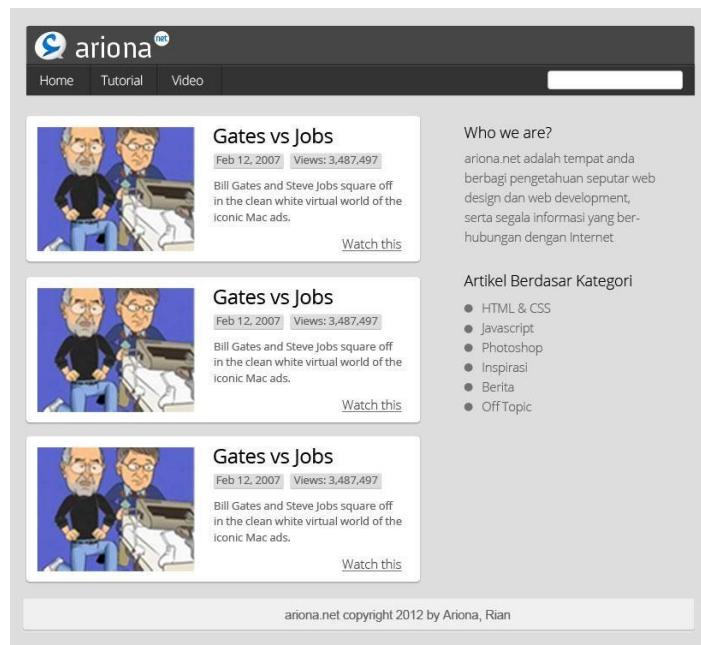
Penggunaan tag <table> untuk keperluan layout adalah menyalahi aturan/fungsi dari tag table tersebut.

Layout table biasanya dibuat secara otomatis oleh aplikasi seperti Dreamweaver atau Photoshop (di-generate oleh slicing tool).

Bab 10

Studi kasus 1- Konversi desain ke HTML

Sebagai seorang web designer, Anda harus mampu menerjemahkan suatu desain web ke dalam bahasa HTML dan CSS. Proses penerjemahan dari desain ke dalam kode HTML ini disebut dengan Pembuatan *Markup*, yang dilanjutkan dengan proses Styling (Penambahan CSS yang akan kita pelajari pada bab selanjutnya).



Gambar 37 Desain web yang akan kita konversi

Pada bab ini anda akan belajar membuat *markup* dari sebuah desain web yang telah disiapkan. Saya tidak akan memberikan desain web yang rumit, sebagai latihan kita gunakan desain yang sederhana dulu.

10.1. Pesiapan

Sebelum kita mulai proses pembuatan markup untuk desain web tersebut, kita buat folder baru terlebih dahulu, agar pekerjaan kita rapih dan terstruktur.

Buatlah folder baru dengan nama “**Latihan Markup**” dan buatlah file HTML baru dengan nama “**latihan-markup.html**”.

Dalam folder tersebut buat lagi satu buah folder dengan nama “**images**” untuk menyimpan seluruh gambar yang diperlukan dalam file HTML yang kita buat. Lalu copy gambar yang terdapat di dalam folder resources/images yang disertakan bersama ebook ini.

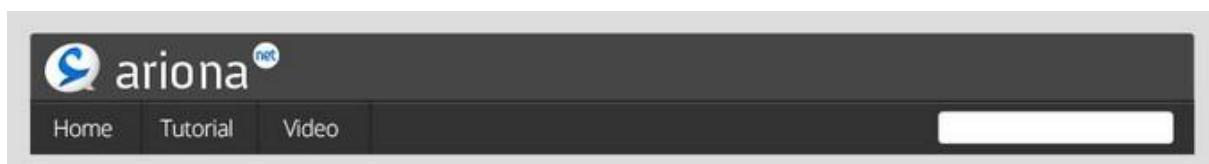
10.2. Wrapper

Kita mulai dari pembuatan sebuah div untuk menampung seluruh konten situs yang akan dibuat. Saya akan menggunakan div dengan id wrapper. Nantinya div ini akan kita buat rata-tengah, jadi setiap browser di zoom-in atau di zoom-out, konten situs tetap berada di tengah.

```
1. <!DOCTYPE HTML>
2. <HTML>
3.     <head>
4.         <title>Latihan Markup</title>
5.     </head>
6.
7.     <body>
8.         <div id=,wrapper>
9.
10.            </div>
11.        </body>
12. </HTML>
```

10.3. Header

Kita mulai dari pembuatan header, perhatikan gambar berikut :



Gambar 38 Bagian header web

Sudah ada bayangan?, pertama kita buat sebuah div dengan nama id header (atau terserah anda). Dalam gambar di atas bisa anda perhatikan terdapat gambar logo, lalu Menu (kita buat dengan unordered list) dan terakhir sebuah text input untuk pencarian.

Semuanya telah anda pelajari pada bab-bab sebelumnya. Jadi, Markup untuk bagian header adalah seperti berikut :

```
8. <div id=,wrapper>
9.     <div id=,header>
10.         <img src=,logo.png alt=,logo />
11.
12.         <ul id=,menu-utama>
13.             <li><a href=,,>Home</a></li>
14.             <li><a href=,,>Tutorial</a></li>
15.             <li><a href=,,>Video</a></li>
16.         </ul>
17.         <form>
18.             <input type=,text name=,search />
19.         </form>
20.     </div>
21.
22.
23. </div>
```

Mudah bukan?, semua tag-tag tersebut telah saya jelaskan sebelumnya. Jika anda masih tidak mengerti atau lupa anda dapat membuka kembali bab sebelumnya.

10.4. Content

Kita lanjutkan dengan proses markup Content/isi Situs.



Gambar 39 Bagian Konten Web

Jika anda perhatikan, disana terdapat tiga Artikel yang memiliki penampilan dan struktur yang sama, untuk itu kita akan menggunakan penamaan dengan Class pada div ini. Namun sebelumnya, kita tulis sebuah div untuk menampung artikel yang akan kita buat dengan nama “**daftar-artikel**”.

```
22. <div id=,daftar-artikel'>
23.     <div class=,artikel'>
24.
25.     </div>
26. </div>
```



Gambar 40 Detail setiap post

Kita masuk ke detail setiap artikel. Di dalam artikel ini terdapat sebuah gambar, judul artikel, tanggal, ringkasan dan sebuah link “watch this”.

Sudah terbayang bagaimana kode HTML nya?

```
22. <div id=,daftar-artikel'>
23.     <div class=,artikel'>
24.         <img src=,images/gambar-artikel.jpg' alt= ,Gambar Artikel' />
25.         <h1> Gates vs Jobs</h1>
26.         <b>Feb 12, 2007</b><b>Views : 3,487,497</b>
27.         <p>Bill Gates and Steve Jobs square off in the clean white virtual world of
        the iconic Mac ads</p>
28.         <a href='''>Watch This</a>
29.     </div>
30. </div>
```

Selanjutnya anda tinggal Copy div class **artikel** ini sebanyak tiga kali. Sehingga menjadi seperti berikut :

```
22. <div id=,daftar-artikel'>
23.     <div class=,artikel'>
24.         <img src=,images/gambar-artikel.jpg' alt= ,Gambar Artikel' />
25.         <h1> Gates vs Jobs</h1>
26.         <b>Feb 12, 2007</b><b>Views : 3,487,497</b>
27.         <p>Bill Gates and Steve Jobs square off in the clean white virtual world of
        the iconic Mac ads</p>
28.         <a href='''>Watch This</a>
29.     </div>
30.     <div class=,artikel'>
```

```

31.      <img src=„images/gambar-artikel.jpg“ alt=„Gambar Artikel“ />
32.      <h1> Gates vs Jobs</h1>
33.      <b>Feb 12, 2007</b><b>Views : 3,487,497</b>
34.      <p>Bill Gates and Steve Jobs square off in the clean white virtual world of
the iconic Mac ads</p>
35.      <a href=„,›Watch This</a>
36.    </div>
37.    <div class=„artikel“>
38.      <img src=„images/gambar-artikel.jpg“ alt=„Gambar Artikel“ />
39.      <h1> Gates vs Jobs</h1>
40.      <b>Feb 12, 2007</b><b>Views : 3,487,497</b>
41.      <p>Bill Gates and Steve Jobs square off in the clean white virtual world of
the iconic Mac ads</p>
42.      <a href=„,›Watch This</a>
43.    </div>
44.  </div>
45.
46.

```

10.5. Sidebar

Kita lanjutkan dengan pembuatan sidebar, perhatikan apa yang terdapat di dalam sidebar.



Gambar 41 Sidebar

Setiap item dalam sidebar memiliki judul. Pertama terdapat deskripsi singkat tentang situs, dan sebuah menu navigasi berdasarkan kategori. Sama seperti menu sebelumnya, kita buat dengan menggunakan unordered list.

```

46.  <div id=„sidebar“>
47.    <h2>Who we are?</h2>
48.    <p>Arionanet adalah tempat anda berbagi pengetahuan seputar web design dan web
development serta segala informasi yang berhubungan dengan internet</p>
49.
50.    <h2>Artikel berdasarkan kategori</h2>
51.    <ul>
52.      <li><a href=„,›HTML dan CSS</a></li>
53.      <li><a href=„,›Javascript</a></li>

```

```
54.      <li><a href=,,>Photoshop</a></li>
55.      <li><a href=,,>Inspirasi</a></li>
56.      <li><a href=,,>Berita</a></li>
57.      <li><a href=,,>Off Topic</a></li>
58.    </ul>
59.  </div>
60.
61.
```

10.6. Footer

Terakhir, kita buat markup untuk bagian footer.



ariona.net copyright 2012 by Ariona, Rian

Gambar 42 Bagian Footer Web

Bagian ini hanya mempunya text copyright, dan saya yakin anda dapat membuat markup dari bagian footer ini :

```
61. <div id=,footer'>
62.   <p>Ariona.net copyright 2012 by Ariona, Rian</p>
63. </div>
```

Mudah bukan? Jika anda merasa bingung bagaimana penulisan kode selengkapnya, berikut ini kode lengkap dari proses markup kita sebelumnya.

```
1.  <!DOCTYPE HTML>
2.  <HTML>
3.    <head>
4.      <title>Latihan Markup</title>
5.    </head>
6.    <body>
7.      <div id=,wrapper'>
8.        <div id=,header'>
9.          <img src=,logo.png' alt=,logo' />
10.         <ul id=,menu-utama'>
11.           <li><a href=,,>Home</a></li>
12.           <li><a href=,,>Tutorial</a></li>
13.           <li><a href=,,>Video</a></li>
14.         </ul>
15.         <form>
16.           <input type=,text' name=,search' />
17.         </form>
18.       </div><!-- Penutup div ,header' -->
19.
20.       <div id=,daftar-artikel'>
21.         <div class=,artikel'>
22.           <img src=,images/gambar-artikel.jpg'
23.             alt=,Gambar Artikel' />
24.           <h1> Gates vs Jobs</h1>
25.           <b>Feb 12, 2007</b><b>Views : 3,487,497</b>
```

```

25.          <p>Bill Gates and Steve Jobs square off in the clean
26.          white virtual world of the iconic Mac ads</p>
27.          <a href=,,>Watch This</a>
28.          </div><!-- Penutup div ,artikel' -->
29.          <div class=,artikel'>
30.              <img src=,images/gambar-artikel.jpg'
31.              alt=,Gambar Artikel' />
32.              <h1> Gates vs Jobs</h1>
33.              <b>Feb 12, 2007</b><b>Views : 3,487,497</b>
34.              <p>Bill Gates and Steve Jobs square off in the clean
35.              white virtual world of the iconic Mac ads</p>
36.              <a href=,,>Watch This</a>
37.              </div><!-- Penutup div ,artikel' -->
38.              <div class=,artikel'>
39.                  <img src=,images/gambar-artikel.jpg'
40.                  alt=,Gambar Artikel' />
41.                  <h1> Gates vs Jobs</h1>
42.                  <b>Feb 12, 2007</b><b>Views : 3,487,497</b>
43.                  <p>Bill Gates and Steve Jobs square off in the clean
44.                  white virtual world of the iconic Mac ads</p>
45.                  <a href=,,>Watch This</a>
46.                  </div><!-- Penutup div ,artikel' -->
47.                  <div id=,sidebar'>
48.                      <h2>Who we are?</h2>
49.                      <p>Ariona.net adalah tempat anda berbagi pengetahuan seputar
50.                      webdesign dan web development serta segala informasi yang
51.                      berhubungan dengan internet</p>
52.                      <h2>Artikel berdasarkan kategori</h2>
53.                      <ul>
54.                          <li><a href=,,>HTML dan CSS</a></li>
55.                          <li><a href=,,>Javascript</a></li>
56.                          <li><a href=,,>Photoshop</a></li>
57.                          <li><a href=,,>Inspirasi</a></li>
58.                          <li><a href=,,>Berita</a></li>
59.                          <li><a href=,,>Off Topic</a></li>
60.                      </ul>
61.                  </div><!-- Penutup div ,sidebar' -->
62.                  <div id=,footer'>
63.                      <p>ariona.net copyright 2012 by Ariona,Rian</p>
64.                  </div>
65.          </div><!-- Penutup div ,wrapper' -->
66.      </body>
67.  </HTML>

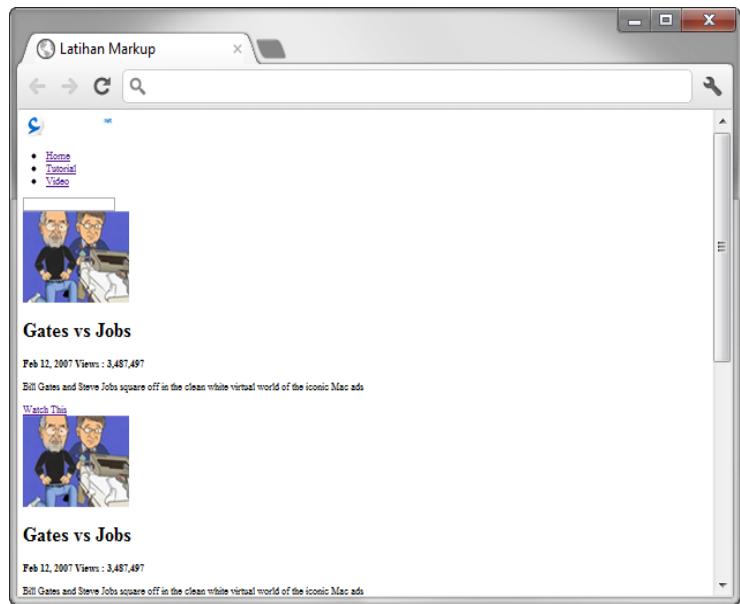
```

Agar tidak bingung membedakan penutup div yang satu dengan yang lainnya, saya menambahkan komentar di setiap penutup div. Komentar ini tidak akan ditampilkan pada browser.

Pembuatan komentar dapat dilakukan dengan menambahkan `<!--` dan ditutup dengan `-->` contohnya :

```
<!-- Komentar dituliskan di sini -->
```

Jika anda buka file HTML tersebut, maka yang akan anda lihat adalah seperti berikut :



Gambar 43 Tampilan dari Studi Kasus

File HTML yang ditampilkan akan terlihat *plain/polos*, tidak sama seperti pada desain web yang diberikan. Tapi inilah tugas dari HTML yakni *Markup/Formatting*. Adapun cara kita menampilkan bagian-bagian kode HTML agar terlihat seperti pada desain web yang diberikan, kita akan menggunakan CSS dan akan kita pelajari pada Bab CSS.

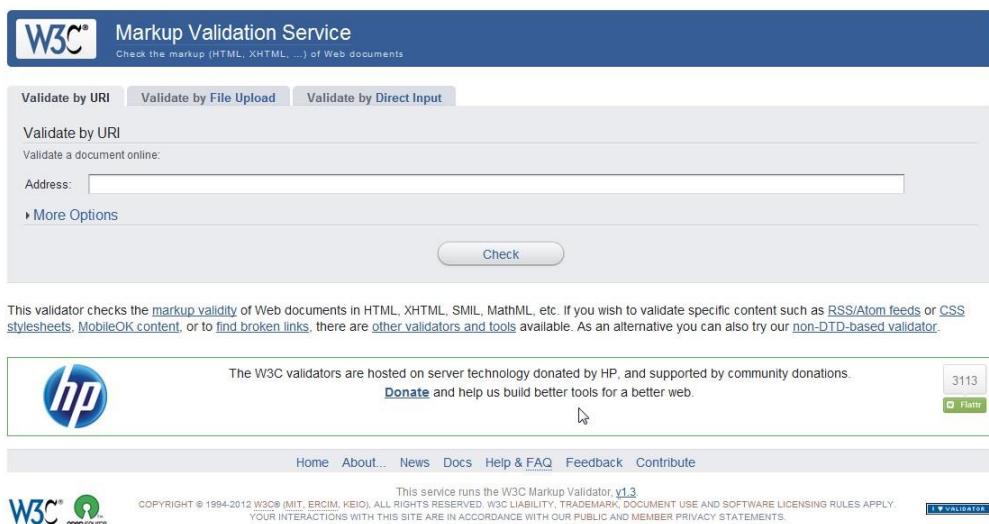
Bab 11

Validasi Markup HTML

Validasi markup adalah proses dimana kode HTML kita divalidasi apakah sudah benar atau belum oleh aplikasi validator dari W3C⁶. Validasi ini sangatlah diperlukan untuk memeriksa apakah ada yang salah dengan markup HTML yang telah kita buat atau ada beberapa bagian yang terlewat seperti penutup tag, atribut dan lain sebagainya.

Sebagian orang berpendapat bahwa markup HTML yang valid akan mempercepat proses loading website dan lebih bersahabat dengan Search Engine⁷.

Untuk melakukan validasi, silahkan buka alamat berikut: <http://validator.w3.org/>. Anda akan dibawa ke halaman validator HTML dari W3C seperti berikut :



Gambar 44 Validator HTML dari W3C

Ada tiga cara pengujian Markup HTML, yakni dengan mengetikkan alamat file HTML yang akan diuji (Validate by URL) opsi ini adalah untuk memvalidasi halaman HTML yang

⁶ W3C (World Wide Web Consortium) adalah komunitas internasional yang bekerja untuk membangun standar web seperti HTML dan CSS.<http://www.w3.org/>

⁷ Search Engine atau mesin pencari adalah aplikasi untuk mencari berbagai informasi di internet, seperti Google.com, bing.com dan lain sebagainya.

sudah “online” atau website yang sudah online. **Cara kedua** adalah dengan mengupload file HTML anda (Validate by file upload), dan validator akan memvalidasi kode HTML yang terdapat dalam file tersebut, dan **cara terakhir** adalah dengan mengetikkan langsung kode HTML dan memvalidasinya (Validate by direct input).

Sebagai contoh, kita coba memvalidasi latihan markup yang kita buat sebelumnya dengan cara “Validate by file upload”. Klik tab “Validate by file upload”, lalu klik Choose file untuk memilih file HTML dan pilihlah latihan Markup yang telah kita buat pada study kasus dan Klik tombol Check untuk mulai memvalidasi file HTML kita.

The screenshot shows the W3C Markup Validation Service interface. At the top, there are three tabs: "Validate by URI", "Validate by File Upload" (which is highlighted in blue), and "Validate by Direct Input". Below the tabs, the "Validate by File Upload" section is active. It contains a label "Upload a document for validation:" followed by a file input field with the placeholder "Choose File No file chosen". There is also a link "More Options". A large "Check" button is centered below the input field. At the bottom, a note states: "Note: file upload may not work with Internet Explorer on some versions of Windows XP Service Pack 2, see our [information page](#) on the W3C QA Website."

Gambar 45 Validate by File Upload

Jika pengujian sukses, maka akan muncul pesan lulus uji markup HTML seperti berikut:

This document was <i>Tentatively</i> checked as HTML5	
Result:	Tentatively passed, 4 warning(s)
File:	<input type="button" value="Choose File"/> No file chosen Use the file selection box above if you wish to re-validate the uploaded file latihan markup.html
Encoding:	utf-8 <input type="button" value="detect automatically"/>
Datatype:	HTML5 <input type="button" value="detect automatically"/>
Root Element:	html

Gambar 46 Hasil validasi markup HTML

Perhatikan, bahwa kita mendapatkan 4 peringatan, coba scroll ke bawah dan anda akan menemukan peringatan tersebut (begitupula dengan error – jika ada).

Validation Output: 4 Warnings

Below is a list of the warning message(s) produced when checking your document.

⚠ The character encoding of the document was not declared.

Gambar 47 Peringatan validasi

Kesimpulan dari peringatan di atas adalah kita belum menyertakan informasi character encoding. Untuk memperbaikinya tambahkan kode berikut di dalam tag head dari file latihan markup kita.

```
<head>
    <title>Latihan Markup</title>
    <meta charset='UTF-8'>
</head>
```

Maksud dari tag tersebut adalah menentukan tipe karakter encoding yang digunakan dalam file HTML, dan kita gunakan tipe “UTF-8”. Sekarang kita coba Revalidate (Memvalidasi ulang) file HTML kita.

This document was successfully checked as HTML5!	
Result:	Passed, 1 warning(s)
File:	<input type="button" value="Choose File"/> No file chosen <small>Use the file selection box above if you wish to re-validate the uploaded file latihan_markup.html</small>
Encoding:	utf-8 <input type="button" value="(detect automatically)"/>
Doxtype:	HTML5 <input type="button" value="(detect automatically)"/>
Root Element:	html

Gambar 48 Lulus uji validasi HTML

Sekarang kita telah lolos uji validasi, 1 peringatan yang tersisa adalah karena validator file HTML versi 5 dari W3C ini masih berupa eksperimen, so.. *it's not problem and you are ready for the next level!*.

Bab 12

Semantik

Sebuah dokumen HTML tidak lain adalah kumpulan dari tag-tag HTML dan seperti yang telah anda ketahui bahwa HTML adalah bahasa untuk *me-markup* atau *mem-format* suatu dokumen. HTML **BUKAN** untuk membuat bagaimana dokumen tersebut ditampilkan.

Sama halnya dengan aplikasi pengolah kata, formating yang dilakukan pada suatu dokumen ditujukan untuk memberikan makna-makna tersendiri pada setiap kata-kata yang diolahnya.

Misalnya, ketika suatu teks ditampilkan miring, biasanya ada makna tertentu pada teks tersebut, seperti pada pertanyaan berikut :

, Berikut ini adalah tag-tag HTML, *kecuali* :

Kata “kecuali” di sana diberi format miring karena penulis ingin menegaskan bahwa jawaban yang benar adalah yang bukan tag HTML. Jika kalimat pertanyaan tersebut dibaca, biasanya akan ada perbedaan intonasi pada kata “kecuali”.

Dalam HTML, terdapat dua tag yang dapat membuat suatu teks miring yakni tag ** dan *<i>*. Lalu apa perbedaan kedua tag tersebut?

Jika kita sangkutpautkan pertanyaan ini dengan contoh kasus di atas, penggunaan tag ** untuk memiringkan kata “kecuali” adalah cara yang benar. Mengingat tag ** ini memang berfungsi untuk memberikan tekanan pada suatu teks (*emphasis*). Berbeda dengan tag *<i>* yang hanya berfungsi untuk menampilkan teks menjadi miring tanpa ada makna sama sekali.

12.1. Maksud Semantik dalam HTML

Semantik berarti makna yang terkandung dari suatu kata. Dalam HTML, semantik berarti makna yang terkandung dari suatu tag HTML. Seperti pada contoh kasus sebelumnya, penggunaan tag lebih memiliki nilai semantik dari pada tag <i>.

Pada dasarnya, Gunakanlah tag-tag HTML sesuai dengan maksud/fungsi dari tag HTML tersebut, Jangan menggunakan suatu tag HTML karena tag tersebut memiliki penampilan berbeda.

Misalnya, penggunaan tag <table>. Tag ini berfungsi untuk membuat suatu tabel atau data tabular, penggunaan diluar itu adalah salah, seperti penggunaan tag table untuk pembuatan layout web.

Untuk mengetahui fungsi dari setiap tag HTML anda dapat membacanya di <http://www.w3schools.com/tags/default.asp>.

12.2. Pentingkah Semantik?

Penulisan dokumen HTML yang semantik ditujukan agar dokumen HTML dapat dipahami oleh komputer, karena pada dasarnya cara berfikir komputer berbeda dengan cara berfikir manusia.

Manusia bisa memahami suatu dokumen dengan adanya tanda baca, atau formatting (seperti teks miring tebal). Berbeda dengan komputer, dan pada kasus ini tag-tag HTML lah yang menjadi tanda baca yang akan membantu komputer untuk memahami dokumen HTML.

Kembali kepada penting tidaknya semantik, boleh jadi dokumen yang semantik dapat mempermudah sistem komputer untuk mengolah informasi dalam dokumen untuk keperluan tertentu, misalnya untuk keperluan Mesin Pencarian (Search Engine) agar hasil pencarian lebih akurat dan sesuai dengan yang diinginkan.

Bagian II

CSS

Bab 13

Cascading Style Sheet (CSS)

13.1. Pengenalan CSS

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

Perhatikan contoh kode CSS berikut :



Gambar 49 Contoh Kode CSS

Selector

Selector adalah elemen/tag HTML yang ingin diberi style. Anda dapat menuliskan langsung **nama tag** yang ingin diberi style tanpa perlu menambahkan tanda <>. Pada contoh kode CSS di atas, kita akan memberi style pada seluruh tag h1 yang terdapat dalam file HTML.

Jika tag HTML yang ingin diberi style memiliki ID, anda dapat menuliskan nama ID tersebut dengan diawali tanda kress (#).

```
#header
```

Dan jika tag yang diberi style memiliki Class, maka penulisan selector bisa dilakukan dengan tanda titik (.) diikuti dengan nama class.

.artikel

Jika anda hanya menuliskan satu selector, seperti contoh kode CSS di atas, maka seluruh tag h1 yang terdapat dalam file HTML akan memiliki style yang sama. Bagaimana jika kita hanya ingin memberi style pada tag h1 yang hanya terdapat di dalam Class artikel. Maka penulisan selectornya seperti berikut :

.artikel h1

Kode tersebut akan memerintahkan pada browser untuk memberi style pada tag h1 yang hanya terdapat di dalam **class artikel** (atau - h1 yang merupakan child dari class artikel).

Anda-pun dapat memilih lebih dari satu tag untuk penghematan kode CSS. Misalnya ketika anda memiliki dua atau lebih tag dengan warna background yang sama, dari pada menuliskan kode seperti ini :

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

anda dapat menggabungkan selector dengan menambahkan tanda koma pada nama tag yang ingin diberi style.

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

Property dan Value

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

Untuk memberikan nilai/value pada property kita gunakan tanda titik dua (:). Setiap property diakhiri dengan titik koma (;), jika anda tidak mengakhirinya maka browser tidak akan mengetahui maksud dari property tersebut.

Property-property pada CSS sangat mudah dimengerti karena lebih mirip bahasa kita sehari-hari. Misalnya untuk merubah warna text kita gunakan property color, untuk

merubah warna background kita gunakan property background-color, untuk merubah ukuran huruf kita gunakan property font-size. Mudah dimengerti bukan?

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

Menurut saya, Kode CSS sangat mudah dimengerti, karena kata-kata yang digunakan sudah tidak asing lagi, perhatikan contoh kode CSS sebelumnya, jika kita ubah ke bahasa kita sehari-hari, kira-kira menjadi seperti berikut :

“Hey Brow (Browser) Cari tag h1 yang terdapat di dalam class artikel, jika ditemukan/ada, ubah warnanya menjadi merah, warna background menjadi biru dan ukuran huruf menjadi 20 pixel.”

Cukup perkenalan dengan CSS-nya, lalu yang menjadi pertanyaan besar anda saat ini adalah :

“Dimana saya mengetikkan kode tersebut?”

13.2. Penulisan CSS

Ada tiga cara penulisan kode CSS, yaitu inline, internal dan external. Ketiganya bisa anda lakukan sesuai dengan kebutuhan. Berikut contoh penggunaan dari metode-metode tersebut:

Inline

Penulisan kode CSS dengan metode inline ini bisa dilakukan langsung pada tag yang ingin diberi style dengan menggunakan atribut style.

```
<h1 style="color : red; > Judul Situs </h1>
```

Pada metode ini, anda tidak perlu menuliskan selector. Karena anda menuliskan CSS langsung pada tag yang ingin diberi style.

Cara ini sangat tidak dianjurkan, karena Anda akan mencampurkan antara “Format” dan “Presentasi”. Cara ini juga tidak efektif ketika anda akan melakukan perubahan pada CSS.

Internal

Metode CSS internal ditulis di dalam tag style yang ditempatkan pada tag head.

```
<HTML>
    <head>
        <title>Judul HTML</title>
        <style>
            h1 {
                color : red;
            }
        </style>
    </head>
...
...
```

Metode kedua ini sangat dianjurkan untuk pengujian style, atau ketika anda hanya memiliki satu halaman web.

External

Metode yang terakhir adalah dengan membuat file CSS dan dipanggil di dalam tag head. File CSS memiliki ekstensi (akhiran) .CSS misanya **namafile.CSS**.

Pemanggilan file CSS dilakukan dengan menggunakan tag link:

```
<HTML>
    <head>
        <title>Judul HTML</title>
        <link rel='stylesheet' href='fileCSS.CSS' />
    </head>
...
...
```

Atribut rel adalah informasi hubungan (relationship) dari tag link tersebut, yaitu sebagai stylesheet. Href diisi dengan lokasi file CSS yang ingin dimuat. Pemanggilannya sama dengan pemanggilan gambar atau link.

Bab 14

Box-Model

14.1. Margin

Property margin digunakan untuk memberikan jarak antar elemen/tag HTML. Misalnya jarak antar <div> dan tag-tag HTML lainnya.

Sebagai contoh, buatlah file HTML baru dengan nama `latihan8.html`, lalu tambahkan kode berikut:

```
<div class="box">  
    Box ke satu  
</div>  
  
<div class="box">  
    Box ke dua  
</div>
```

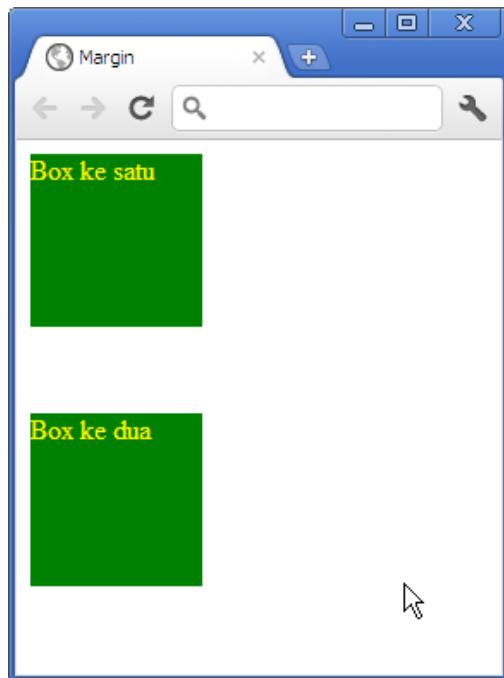


Ingat, tambahkan kode tersebut pada struktur HTML standar yang memiliki <!DOCTYPE HTML>, <HTML>, <head> dan seterusnya.

Lalu tambahkan tag style dalam tag head, dan ketikkan kode CSS berikut :

```
.box{  
    background-color:green;  
    color: yellow;  
    width:100px;  
    height:100px;  
    margin-bottom:50px;  
}
```

Berikut adalah tampilan yang akan anda dapatkan :



Gambar 50Penambahan margin-bottom pada .box

Jarak antar box ke satu dan box ke dua sedikit berjauhan, karena kita menambahkan margin bawah (margin-bottom) sebesar 50px.

Untuk melihat perbedaan ketika kita memberikan margin atau tidak, anda dapat menghapus property margin-bottom dan perhatikan perbedaannya.

Anda juga dapat memberikan jarak dari kanan (right), kiri (left) dan atas (top). Misalnya :

```
margin-top:20px;  
margin-right:30px;  
margin-bottom:50px;  
margin-left:40px;
```

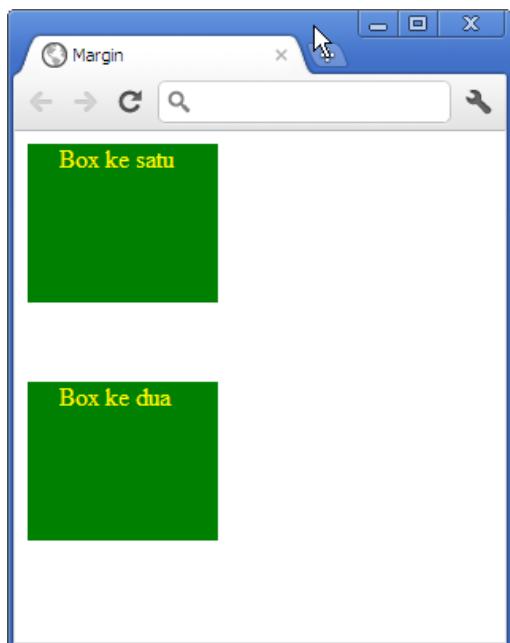
14.2. Padding

Pada latihan sebelumnya, kita telah memberikan jarak antar elemen HTML dalam contoh sebelumnya jarak antar <div>, namun coba perhatikan antara text (Box ke x) dengan garis pinggir kotak, tidak ada jarak sama sekali, merapat ke sisi box tersebut.

Untuk itu *property padding*lah yang akan memberikan jarak antara konten elemen dengan elemen tersebut.

Sebagai contoh tambahkan kode CSS berikut, pada class box yang kita buat sebelumnya:

```
.box{  
background-color:green;  
color: yellow;  
width:100px;  
height:100px;  
margin-bottom:50px;  
padding-left:20px;  
}
```



Gambar 51 Pemberian padding pada .box

Dan sekarang, anda dapat melihat jarak antara text yang berada di dalam box.

Perlu diperhatikan juga, bahwa ketika kita menambahkan padding, ukuran .box juga akan berubah.

Misalnya kita menentukan ukuran lebar dari .box sebesar 100px. Ketika kita menambahkan padding-left sebesar 20px, maka total lebar dari .box tersebut menjadi 120px.

Jadi jika anda ingin ukuran .box tetap 100px, kita harus menghitung ulang width dari box tersebut. Misalnya kita ubah menjadi 80px. Dengan begitu, 80px + 20px menjadi 100px.

14.3. CSS-Shorthand

Pada pembahasan sebelumnya, bisa kita simpulkan bahwa setiap property dapat memiliki lebih dari satu jenis, misalnya margin dapat memiliki 4 jenis. margin-top, margin-right, margin-bottom, dan margin-left.

```
margin-top:20px;  
margin-right:30px;  
margin-bottom:50px;  
margin-left:40px;
```

Daripada menuliskan seluruh property tersebut, kita bisa menggunakan *CSS-Shorthand* yaitu menggabungkan seluruh value (top, right, bottom, dan left) ke dalam satu property.

Sebagai contoh, kita akan menggunakan *CSS-shorthand* untuk Property margin menjadi seperti berikut :

```
margin : 20px 30px 50px 40px;
```

Masing-masing dari nilai tersebut adalah :

```
margin : top right bottom left;
```

bisa kita simpulkan bahwa nilai dari CSS-shorthand margin ini memiliki urutan searah jarum jam dan dimulai dari margin-top.

Kita juga bisa bermain-main dengan property margin ini, perhatikan dengan seksama contoh berikut:

- ✓ Ketika kita memberikan **satu nilai** pada property margin, ini artinya kita memberikan nilai tersebut pada seluruh sisi.

```
margin: 20px;
```

Dengan penulisan seperti ini, maka nilai dari margin-top, right, bottom dan left adalah 20px;

- ✓ Ketika kita memberikan **dua nilai** pada property margin, berarti nilai selanjutnya mengikuti nilai sebelumnya. Contohnya seperti ini :

```
margin: 20px 30px;
```

Pada contoh di atas berarti kita memberikan nilai margin-top dan right bukan?, nah nilai dari margin-bottom akan disamakan dengan nilai margin-top dan margin-left sama dengan margin-right.

- Dan ketika kita menuliskan **tiga nilai**, maka nilai yang tidak diisi akan mengikuti nilai yang diisi, sama halnya ketika kita memberi 2 nilai.

```
margin: 20px 30px 40px;
```

Pada contoh di atas, kita memberikan nilai untuk top, right dan bottom. Sedangkan untuk left akan disamakan dengan right.

Penggunaan nilai seperti ini juga berlaku pada padding, dan property-property sejenis lainnya.

14.4. Border

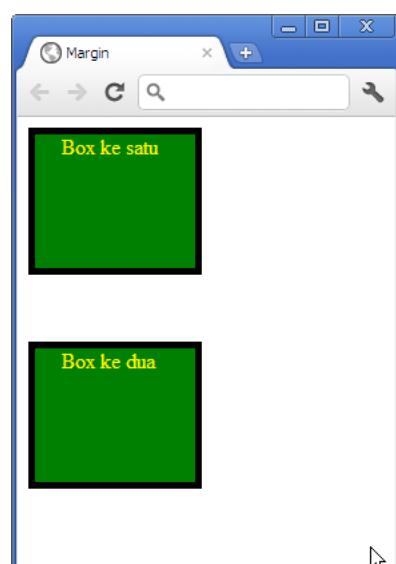
Property border digunakan untuk memberikan garis pinggir pada elemen, setidaknya ada tiga property yang harus anda set, yaitu color, style dan width.

```
border-color : black;
border-style : solid;
border-width : 5px;
```

namun anda juga dapat menggunakan *CSS shorthand* sehingga menjadi seperti berikut :

```
border : width style color;
```

jadi jika kita ubah contoh kode di atas menjadi CSS-shorthand, kodenya menjadi seperti berikut :



```
border : 5px solid black;
```

Cobalah tambahkan kode tersebut pada latihan sebelumnya dan yang akan anda dapatkan adalah seperti berikut :

Sama halnya dengan Padding, penambahan property border juga akan merubah ukuran dari .box.

Jadi, seperti yang saya jelaskan sebelumnya. Ketika anda

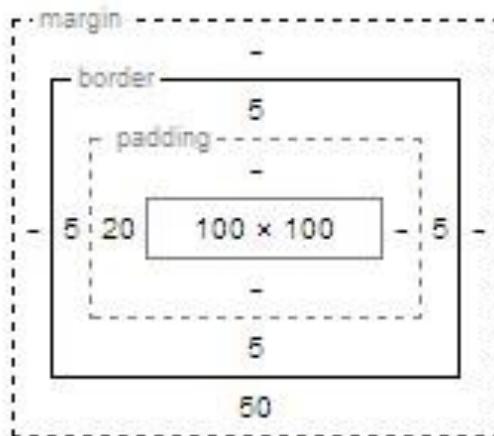
Gambar 52 Penambahan border

ingin membuat ukuran lebar .box tetap 100px, maka anda harus menghitung ulang width dari .box tersebut.

Kali ini perhatikan bahwa kita memberikan border pada seluruh sisi .box, artinya jika kita tentukan ukuran border sebesar 5px, maka 10px (untuk width, karena 5px untuk left, dan 5px untuk right) akan ditambahkan pada ukuran .box tersebut. Sehingga ukuran .box berubah kembali menjadi 110px.

Untuk itu kita kurangi width dari .box tersebut menjadi 70px sehingga : 70px (width) + 20px (padding) + 10px (border) = 100px.

Dalam dunia CSS, kasus seperti ini disebut dengan box-model. Terdapat dua property yang mempengaruhi ukuran dari sebuah box yaitu padding dan border. Perhatikan skema box model berikut untuk contoh kasus di atas :



Gambar 53 skema box-model



Tahukah anda, bahwa setiap elemen/tag HTML ditampilkan dalam bentuk box/kotak, jadi aturan box-model berlaku untuk seluruh elemen HTML

Bab 15

Typography

Secara *default* (bawaan), halaman web yang kita buat akan menggunakan font standar yaitu *Times New Roman*. Namun anda dapat merubahnya dengan property font-family. Berikut contoh penggunaan property font-family :

```
font-family: calibri;
```

Namun perlu diperhatikan bahwa tidak semua PC memiliki font yang kita tentukan, misalnya ketika kita menentukan font-family dengan calibri, maka pada PC dengan sistem operasi Linux, tidak akan ditemukan font tersebut.

Untuk itu anda dapat memberikan lebih dari satu jenis font, font-font yang disebutkan selanjutnya akan menjadi pengganti ketika font sebelumnya tidak ditemukan. Contohnya :

```
font-family: calibri, arial, tahoma, verdana, sans-serif;
```

Ketika font jenis calibri tidak ditemukan, maka yang akan digunakan adalah arial, ketika tidak ditemukan font arial maka tahoma yang akan digunakan, begitu seterusnya sampai kita tentukan jenis font yang dipakai, sans-serif atau serif dan browser akan menggunakan font yang sesuai dengan jenis yang disebutkan (serif atau sans-serif).



Ketika nama font memiliki spasi, maka anda harus memberikan tanda petik/kutip pada nama font tersebut. Misalnya "segoe ui".

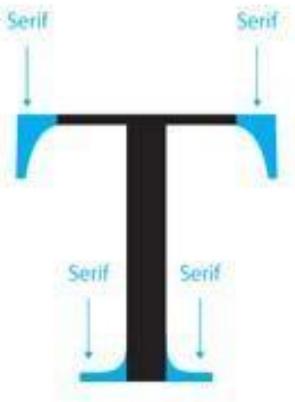
Apa itu Sans-serif dan Serif

Dalam typography jenis font dibagi menjadi dua yakni serif dan sans-serif. Bagian ini akan menjelaskan secara singkat perbedaan dari kedua jenis font ini.

Serif

Kata Serif bisa diartikan sebagai kait, jadi jenis huruf serif memiliki kait disetiap ujung hurufnya. Font jenis serif yang paling terkenal adalah Times New Roman. Perhatikan detail hurufnya.

Font jenis ini bisa kita gunakan untuk teks dengan konteks formal atau gaya klasik. Namun anda dapat memadupadankan jenis font ini sehingga memberikan kesan tertentu pada web anda.



Gambar 54 Font jenis Serif memiliki kait disetiap ujungnya

Sans-serif

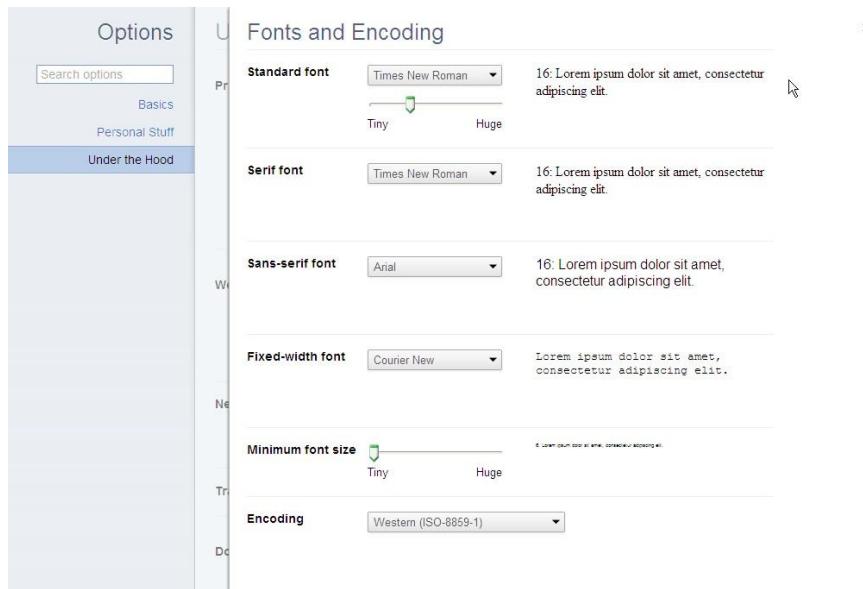
Arti dari sans, adalah “tanpa”. Jadi sans-serif bisa diartikan sebagai jenis font tanpa kait. Yang paling umum digunakan untuk jenis sans-serif adalah arial.

Jenis ini memberikan kesan dynamis dan luwes. Seperti saya sebutkan sebelumnya anda dapat membuat kesan tersendiri ketika memadupadankannya.

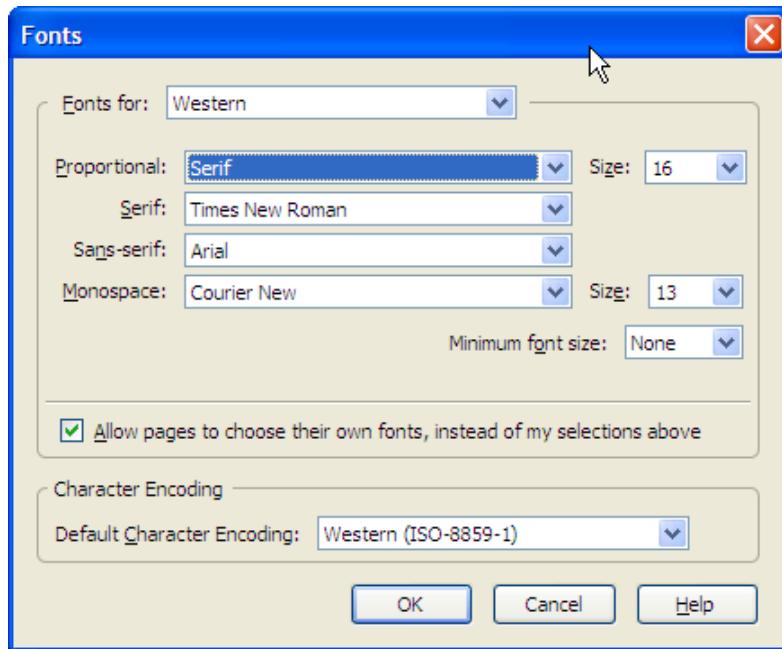


Gambar 55 Font jenis Sans Serif

Dalam pengaturan browser, kita bisa memilih font untuk serif dan sans-serif. Jadi ketika font-family menggunakan serif, maka font yang telah kita pilihlah yang akan digunakan.



Gambar 56 pengaturan font pada browser Google Chrome



Gambar 57 Pengaturan font pada firefox

Untuk menentukan **ukuran** font, kita gunakan property `font-size`:

```
font-size : 12px;
```

anda juga dapat menggunakan satuan pt (point) untuk ukuran font :

```
font-size : 12pt;
```

font yang kita gunakan pun dapat ditentukan **ketebalannya** dengan menggunakan property font-weight. Nilai yang bisa dipakai pada property ini adalah normal, bold, bolder.

```
font-weight : bold;
```

Bukan tidak mungkin untuk mengatur kerning atau jarak antar huruf. Dalam CSS kita bisa merubah jarak antar huruf dengan property letter-spacing.

```
letter-spacing : 15px;
```

Semakin besar nilai pixel yang diberikan semakin jauh jarak antar huruf. Andapun dapat menggunakan nilai negatif untuk memperdekat jarak antar huruf.

```
letter-spacing : -2px;
```

Sedangkan untuk memberikan jarak antar baris, kita gunakan property line-height.

```
line-height : 30px;
```

Dan untuk membuat text menjadi rata kiri, tengah atau kanan, kita gunakan property text-align. Nilai yang dapat digunakan adalah left, center, right, dan justify untuk rata kiri-kanan.

```
text-align : center;
```

Cobalah untuk bemain-main dengan property tersebut dan perhatikan perubahan yang didapatkan. Buatlah file latihan9.html, isi dengan beberapa paragraf beserta headingnya dan terapkan property-property yang telah anda pelajari di atas.

Bab 16

CSS-Image

Sebenarnya, berbicara tentang image dalam CSS, maka kita sedang membicarakan background-image. Bagaimana menggunakan sebuah gambar menjadi background untuk suatu elemen. Pada bagian ini anda akan mempelajari penggunaan background-image, teknik sprite, dan background-repeat.

16.1. background-image

Untuk memanggil gambar sebagai background dari elemen, kita gunakan property background-image. Perhatikan contoh penggunaan property ini :

```
background-image : url ('images/background.jpg');
```

Ketika anda menggunakan gambar dengan ukuran kecil maka anda akan melihat background ditampilkan secara berulang untuk memenuhi seluruh element tersebut.

Untuk percobaan buatlah file `latihan10.html` dan selanjutnya buatlah sebuah div, dan menentukan ukuran width dan heightnya.

```
<div id = 'box'></div>
```

Dan tambahkan style/CSS seperti berikut :

```
#box{  
    width : 300px;  
    height :300px;  
    background : url(logo2.png);  
}
```

Gambar 58 Background akan terus diulang untuk memenuhi ukuran elemen

16.2. background-repeat

Pengulangan gambar background tersebut bisa kita atur dengan menggunakan property `background-repeat`. Anda bisa mengulang gambar secara horizontal (sesuai sumbu x) atau secara vertikal (y) atau anda pun dapat tidak mengulangi background tersebut.

```
background-repeat : repeat-x;
```

Gambar 59 Repeat X akan mengulang background secara horizontal

```
background-repeat : repeat-y;
```

Gambar 60 Pengulangan background secara vertical dengan repeat-y

```
background-repeat : no-repeat;
```

Gambar 61 Background tidak diulang

16.3. background-position

Posisi gambar background dapat kita tentukan letaknya dengan menentukan jarak x (horizontal) dan y (vertical).

Sebagai contoh, kode berikut akan menggeser background-image sebesar 50px dari kiri.

```
background-position-x : 50px;
```

Gambar 62 background bergeser selebar 50px dari kiri

Dan contoh berikut akan menggeser background-image sebesar 50px ke bawah.

```
background-position-y : 50px;
```

Gambar 63 Background turun 50px dari posisi sebelumnya

Andapun dapat menggunakan CSS-shorthand untuk background-position :

```
background-position : nilaiX nilaiY;
```

misalnya

```
background-position: 50px 50px;
```

Gambar 64 posisi background ada pada 50px dari kiri dan atas

16.4. Image Sprite

Dengan adanya property `background-position` untuk menentukan lokasi penempatan gambar, kita dapat bermain-main dengan image-sprite. Image Sprite adalah sekumpulan gambar berbeda yang disatukan dalam satu file gambar.

Gambar 65 contoh Image sprite

Teknik ini dilakukan untuk mempercepat loading website tanpa harus memanggil beberapa file gambar yang banyak. Cukup satu gambar dan kita tentukan lokasi gambar yang ingin ditampilkan.

Buatlah file baru dengan nama file `latihan11.html` dan ketikkan markup HTML seperti berikut :

```
<div id=,button'></div>
```

Lalu tambahkan style CSS berikut :

```
#button {  
    width:240px;  
    height:83px;  
    background-image : url(sprite.jpg);  
    background-position-y : 0;  
}  
#button:hover{  
    background-position-y : -83px;  
}
```

Style tersebut akan mengambil gambar yang terdapat di posisi x, 0 dan posisi y turun 83px. Sehingga gambar yang ditampilkan adalah bagian berikut :

Gambar 66 Hanya sebagian gambar yang ditampilkan

Lalu apa maksud dari selector `#button:hover`? `:hover` adalah salah satu Pseudo Class, saya akan membahasnya di bab selanjutnya.

Contoh penggunaan Image sprite ini banyak diimplementasikan pada beberapa situs besar seperti Google atau facebook, berikut ini adalah image sprite untuk icon-icon yang ada dalam google plus.



Gambar 67 Image Sprite Google Plus

Bab 17

Floating

Jika sebelumnya untuk mengatur text/paragraf rata kanan atau kiri kita menggunakan property text-align, namun bagaimana jika yang ingin kita atur rata kiri atau rata kanan adalah elemen HTML?

Untuk itu kita dapat menggunakan property float. Nilai yang digunakan yaitu left dan right. Perhatikan contoh pada `latihan12.html` berikut dan perhatikan apa yang terjadi :

Buatlah markup HTML seperti berikut :

```
  
<p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.</p>
```

Lalu tambahkan CSS berikut :

```
img {  
    float: left;  
}
```

Style diatas akan meratakan elemen gambar dan membuat paragraf naik keatas.

Gambar 68 Penambahan Float akan mempengaruhi element disekitarnya.

17.1. Permasalahan pada floating

Ada satu permasalahan ketika anda menambahkan floating pada suatu elemen, permasalahan ini bisa anda temui pada contoh kasus dalam file latihan13.html berikut.

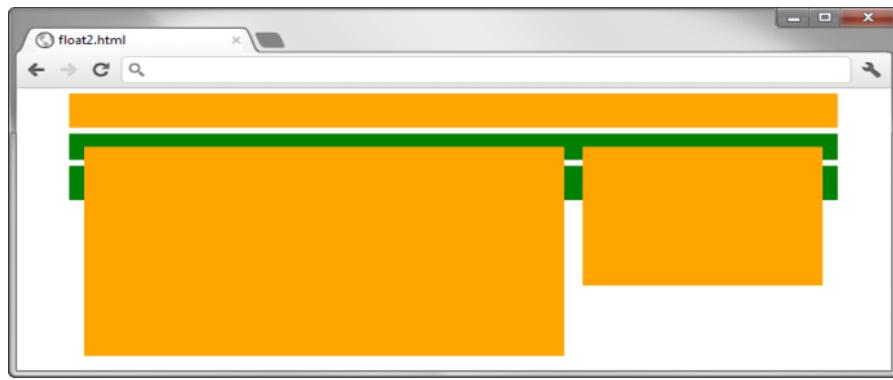
Buatlah markup HTML seperti berikut :

```
1. <div id=,wrapper>
2.     <div id=,header></div>
3.
4.     <div id=,content>
5.         <div id=,artikel></div>
6.         <div id=,sidebar></div>
7.     </div>
8.
9.     <div id=,footer></div>
10. </div>
```

Lalu berilah style berikut :

```
1. #wrapper{
2.     width : 960px;
3.     margin : 0 auto; /*agar wrapper tetap berada ditengah */
4. }
5. #header {
6.     width:100%;
7.     height:50px;
8.     background: orange;
9.     margin-bottom: 10px;
10. }
11. #content{
12.     padding:20px;
13.     background : green;
14.     margin-bottom : 10px;
15. }
16. #artikel{
17.     width : 600px;
18.     height : 700px;
19.     float : left;
20.     background: orange;
21. }
22. #sidebar {
23.     width : 300px;
24.     height: 200px;
25.     float : right;
26.     background: orange;
27. }
28. footer{
29.     width : 100%;
30.     height: 50px;
31.     background: green;
32. }
```

Jika anda buka dalam browser, maka tampilan yang akan dapat adalah seperti ini.



Gambar 69 Permasalahan ketika anda menerapkan Floating

Ada yang janggal? Ya.. #content tidak memenuhi isi dari #artikel dan #sidebar. Seharusnya tinggi #content otomatis sama dengan tinggi dari elemen tertinggi di dalamnya, dalam kasus ini adalah artikel dengan tinggi 700px. Dengan begitu seharusnya tinggi dari #content adalah 740px (ditambah dengan padding-top dan padding-bottom). Jika anda tidak mengerti cobalah untuk membuang property float yang ada, dan perhatikan bahwa #artikel dan #sidebar berada di dalam #content.

Hal ini dikarenakan kita menambahkan floating pada #artikel dan #sidebar, sehingga #content menganggap tidak ada element didalamnya. Ada dua cara untuk menyelesaikan kasus ini :

1. Menambahkan style overflow:hidden pada #content

```
#content{  
    ...  
    overflow:hidden;  
}
```

2. Menambahkan div kosong dengan style clear:both; sebelum penutup content

```
    ...  
    <div id="sidebar"></div>  
    <div style="clear:both;"></div>  
</div>
```



Float berarti melayang. Ketika kita memberikan float pada elemen maka elemen tersebut seolah-olah melayang dan elemen di atasnya (parent) menganggap tidak ada elemen didalamnya (child).



Gambar 70 Penyelesaian masalah Float

17.2. CSS Reset

Pernahkah anda bertanya darimana warna link dari tag `<a>` berasal, atau mengapa tag `` bisa menebalkan teks, atau kenapa tag `<h1>` sampai tag `<h6>` memiliki ukuran font yang berbeda-beda dan ketebalan yang berbeda? Padahal kita sama sekali belum menambahkan CSS pada markup yang kita buat?

Setiap browser memiliki Style atau File CSS yang akan diterapkan pada setiap halaman web yang dibuka. Keberadaan style yang dimiliki browser ini sangat membantu namun terkadang juga membingungkan bagi sebagian orang.

CSS Reset berfungsi untuk mereset atau mengatur ulang seluruh property-property yang telah disetting oleh browser, seperti margin, padding, warna huruf (link) ukuran huruf heading dan lain sebagainya. Perhatikan CSS contoh berikut :

```
* {  
    margin:0;  
    padding:0;  
}
```

Selector bintang (*) digunakan untuk memberikan style pada Seluruh elemen/Tag HTML. Dalam kasus ini, kita akan mengatur ulang margin dan padding menjadi 0 pada seluruh elemen HTML yang ada.

Bab 18

Positioning

Dalam CSS kita pun dapat memposisikan (*Positioning*) elemen sesuai kehendak kita. Hal ini membuat desain web dengan CSS lebih fleksibel karena kita bisa menempatkan elemen-elemen HTML sesuai dengan keinginan kita atau tuntutan desain web.

18.1. Static

Secara default, seluruh elemen/tag HTML memiliki positioning static. Artinya anda belum bisa/tidak dapat menentukan posisi elemen tersebut. Ketika anda ingin merubah posisi dari elemen tersebut maka anda harus memberikan position:relative dan selanjutnya anda dapat menggunakan property top, right, bottom, dan left untuk menentukan posisi elemen yang bersangkutan.

Sebagai contoh perhatikan kode HTML dan CSS berikut :

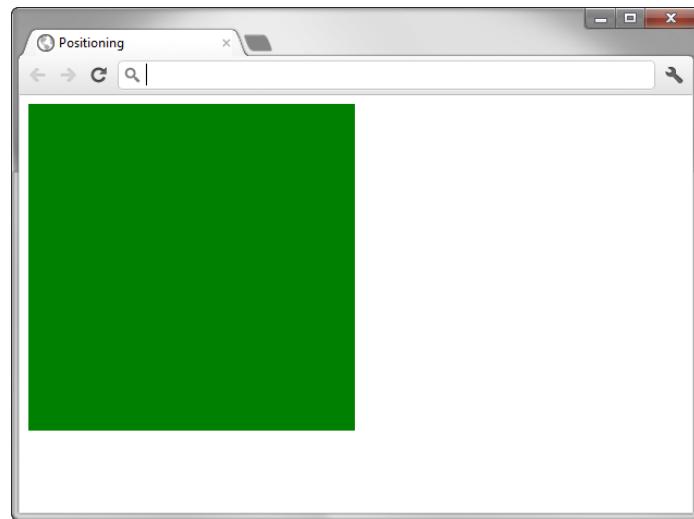
```
<div id="box"></div>
```

Berikut kode CSSnya.

```
#box{  
    width : 300px;  
    height : 300px;  
}
```

Ketika anda tidak memberikan nilai pada property position, maka secara default, #box memiliki position:static, jadi anda tidak dapat menggunakan property top, right, bottom dan left untuk menentukan posisi #box tersebut :

```
#box{  
    background : green;  
    width : 300px;  
    height : 300px;  
    top : 100px;  
    left : 200px;  
}
```

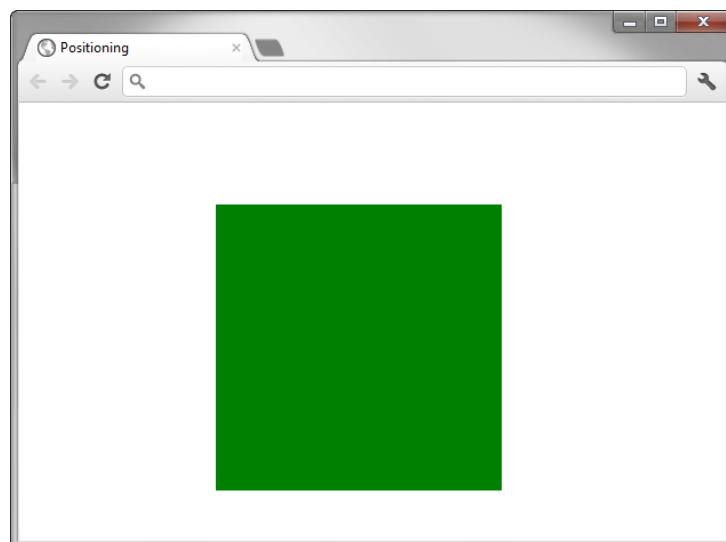


Gambar 71 Box static tidak dapat ditentukan posisinya

18.2. Relative

Sekarang cobalah untuk menambahkan `position:relative` pada `#box` dan perhatikan bahwa box tersebut bergeser 100px dari atas dan 200px dari kiri.

```
#box{  
    background : green;  
    width : 300px;  
    height : 300px;  
    top : 100px; left  
    : 200px;  
    position:relative;  
}
```



Gambar 72 Penambahan `position relative` memungkinkan penempatan box

18.3. Absolute

Penambahan position:absolute akan membuat elemen tersebut terlepas dari aturan penempatan elemen dan dibatasi oleh parent yang memiliki positioning relative. Sulit memahami apa yang saya tulis diatas, kita langsung ke contoh kasus.

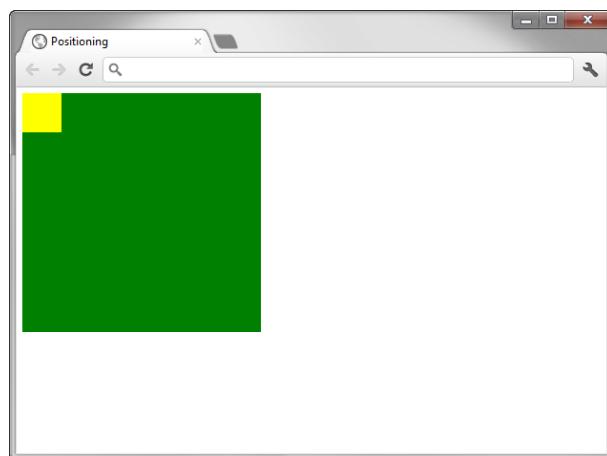
Buatlah markup HTML seperti berikut :

```
<div id="box">
  <div id="box2"></div>
</div>
```

Lalu berilah style seperti berikut :

```
#box{
    width:300px;
    height:300px;
    background:green;
}

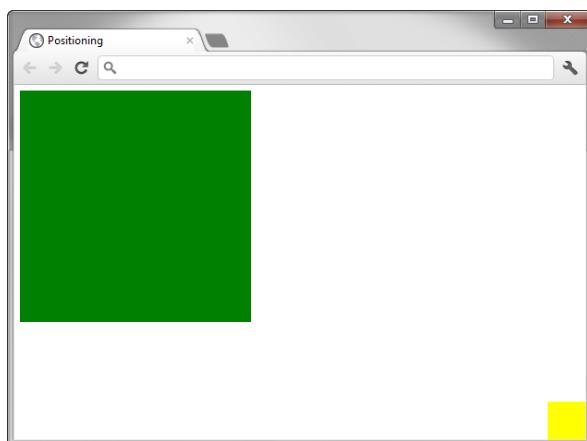
#box2{
    width:50px;
    height:50px;
    background:yellow;
}
```



Gambar 73 Child elemen akan berada di dalam parentnya

#box2 secara aturan ditampilkan berada di dalam div #box, anda dapat terlepas dari aturan tersebut dengan menambahkan positioning absolute. Sekarang tambahkan position:absolute dan tentukan property bottom dan right.

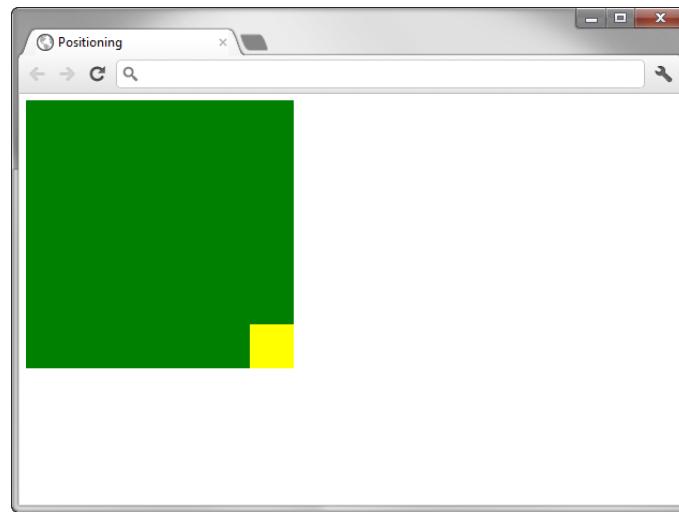
```
#box2{  
    width:50px;  
    height:50px;  
    background:yellow;  
    position:absolute;  
    bottom:0;  
    right:0;  
}
```



Gambar 74 #box2 akan terlepas dari parent ketika ditambahkan position absolute

#box2 akan terlepas dari aturan penempatan elemen (keluar dari #box) dan berada di kanan bawah browser. Untuk membatasi pergerakan dari #box2 yang memiliki positioning absolute, maka kita harus menambahkan positioning relative pada parentnya yakni #box.

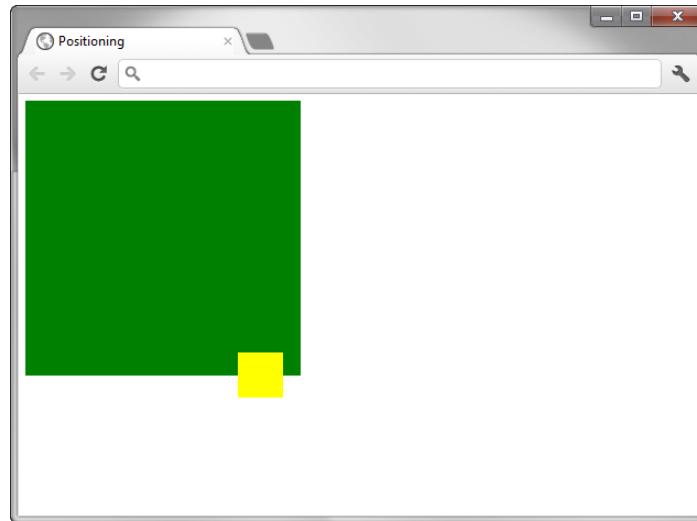
```
#box{  
    width:300px;  
    height:300px;  
    background:green;  
    position:relative;  
}
```



Gambar 75 Posisi absolute position dapat dibatasi oleh elemen dengan posisi relative

Sekarang #box2 berada di kanan bawah parentnya #box. Andapun dapat bermain dengan nilai negative untuk mendapat efek tertentu.

```
#box2{  
    width:50px;  
    height:50px;  
    background:yellow;  
    position:absolute;  
    bottom:-25px;  
    right:20px;  
}
```



Gambar 76 Pengaturan posisi dengan absolute

Perhatikan contoh penggunaan absolute position yang sering ditemukan di internet berikut :



Gambar 77 Contoh Kasus penggunaan Absolute Position

Perhatikan contoh berikut tombol untuk sharing via facebook atau twitter dan nomor posting ditampilkan berjajar seacara vertical dan setengah keluar dari kotak artikel.



Sulit memahami Relative-Absolute? Bayangkan "Relative" sebagai sangkar bagi "Absolute", oleh karena itu pergerakan "Absolute" dibatasi oleh "Relative".

18.4. Fixed

Sifat dari position fixed hampir sama dengan absolute, namun posisi dari elemen fixed akan tetap berada posisi yang ditentukan walaupun browser dizoom-in/out atau scroll ke atas atau ke bawah.

Perhatikan contoh kode HTML dan CSS berikut :

```
<div id="social"></div>
```

CSS :

```
#social{ width:30px;
          height:100px;
          position:fixed;
          left:0;
          top:100px;
        }
```

Berikut ini adalah contoh penggunaan position:fixed yang sering ditemui di internet.



Gambar 78 Contoh Kasus “Jadul” penggunaan fixed position

Bagian Sign Up di situs kaskus.co.id menggunakan position fixed sehingga ketika anda men-scroll halaman web, bagian tersebut tetap berada disana.

Bab 19

Pseudo-Class

Pseudo Class digunakan untuk memberikan efek-efek tertentu pada selector. Misalnya efek ketika cursor berada di atas suatu elemen (hover), ketika cursor meng-klik suatu elemen dan lain sebagainya.

Berikut adalah cara penulisan pseudo class :

```
selector:pseudo-class { property:value; }
```

Misalnya kita akan menambahkan pseudo-class :hover pada sebuah link

```
a:hover {  
    color : red;  
}
```

Style tersebut akan memberikan efek perubahan warna link menjadi merah **ketika** cursor berada di atasnya.



Gambar 79 Pseudo-Class :hover memberikan efek ketika cursor berada di atas elemen

19.1. Pseudo-Class untuk Link/Anchor

Ada beberapa pseudo-class untuk anchor/link yakni :

```
/* untuk link yang belum di klik */  
a:link { color: blue; }  
  
/* untuk link yang sudah di klik */  
a:visited { color:green; }  
  
/* ketika cursor berada di atas link */  
a:hover { color:red; }  
  
/* ketika link dalam keadaan aktif atau terpilih */
```

```
a:active { color: orange; }
```

Perhatikan juga urutan ketika penulisan semua pseudo class tersebut (Jika anda bermaksud menuliskan seluruh pseudo class untuk link):

- ✓ :hover harus ditulis sesudah :link dan :visited
- ✓ :active harus ditulis sesudah :hover.

Jika tidak demikian maka efek tersebut tidak akan berjalan dengan baik.

Pseudo Class :hover dan :active tidak hanya bisa digunakan untuk link saja melainkan bisa diterapkan pada elemen/tag-tag HTML lainnya. Masih ingat dengan demonstrasi Image Sprite pada bab sebelumnya?

19.2. Pseudo-Class :first-child dan :last-child

Selain pseudo-class untuk link/anchor ada juga pseudo class :first-child dan :last-child. Keduanya digunakan untuk memilih elemen pertama dan terakhir dari suatu selector.

Misalnya kita mempunyai sebuah unordered list (ul), dengan beberapa list-item di dalamnya. Kita bisa memberikan style khusus untuk list-item yang pertama dan yang terakhir dengan menggunakan :first-child dan :last-child.

```
<ul>
  <li>Item ke 1</li>
  <li>Item ke 2</li>
  <li>Item ke 3</li>
  <li>Item ke 4</li>
  <li>Item ke 5</li>
</ul>
```

Dan jika kita tambahkan style berikut :

```
ul li:first-child{
  color:green;
}

ul li:last-child{
  color:orange;
}
```

Maka yang akan kita dapatkan adalah tampilan seperti berikut :



Gambar 80 Pemberian pseudo class :first-child dan :last-child

List-item yang pertama akan memiliki warna huruf hijau dan yang terakhir memiliki warna orange. Tidak hanya pada unordered list, andapun dapat menerapkannya pada deretan elemen lainnya. Misalnya elemen p yang terdapat dalam suatu div.

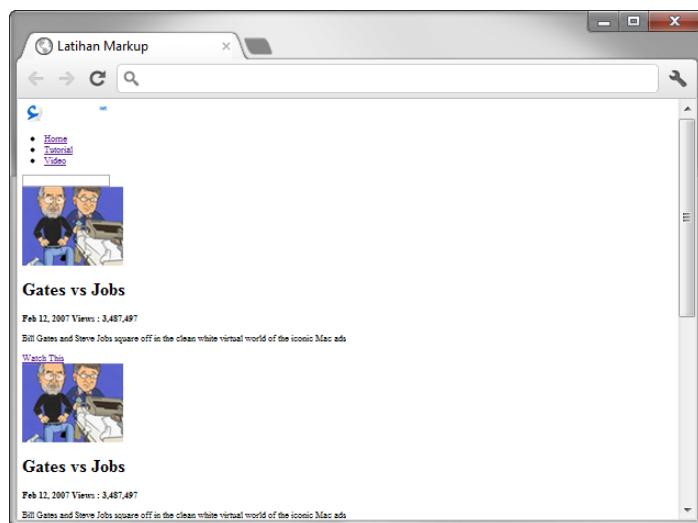
```
<div>
  <p> Paragraf pertama </p>
  <p> Paragraf kedua </p>
  <p> Paragraf ketiga </p>
  <p> Paragraf keempat </p>
</div>
```

Bab 20

StudiKasus 2 – Pemberian Style/CSS

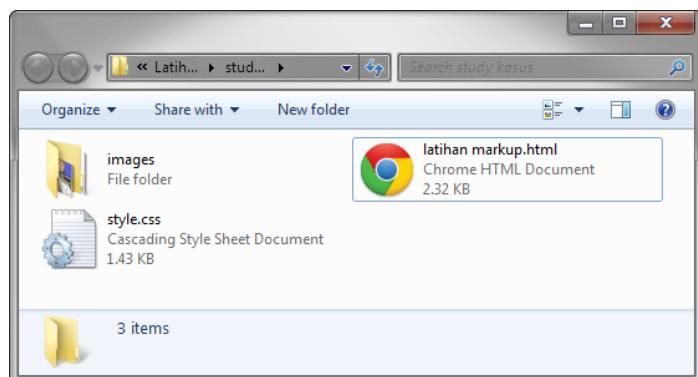
Pada studi kasus sebelumnya, anda telah membuat markup HTML dari sebuah desain web sederhana. Hasil yang didapatpun terasa polos, tidak menarik dan yang terpenting tidak sama dengan desain web yang diinginkan.

Pada bagian ini kita akan belajar menambahkan style/CSS pada file studi kasus yang kita buat sebelumnya.



Gambar 81 Markup HTML tanpa CSS

Pertama-tama, buatlah file CSS baru dengan nama style.css, simpan di folder yang sama dengan lokasi file HTML.



Gambar 82 Penempatan file latihan

Lalu tambahkan tag link untuk memanggil file style.CSS pada bagian header.

```
<head>
    <title>Latihan Markup</title>
    <link rel='stylesheet' href='style.css' />
</head>
```

20.1. CSS Reset

Pertama-tama kita atur ulang margin dan padding agar tidak menggunakan style bawaan dari browser, dan kita tentukan jenis font standar yang akan kita gunakan.

```
*{
    margin:0;
    padding:0;
    font-family:'calibri', 'segoe ui', arial, tahoma, sans-serif;
}
```

20.2. Body

Dasar background dari desain web berwarna abu-abu, jadi kita tentukan warna tersebut pada tag body:

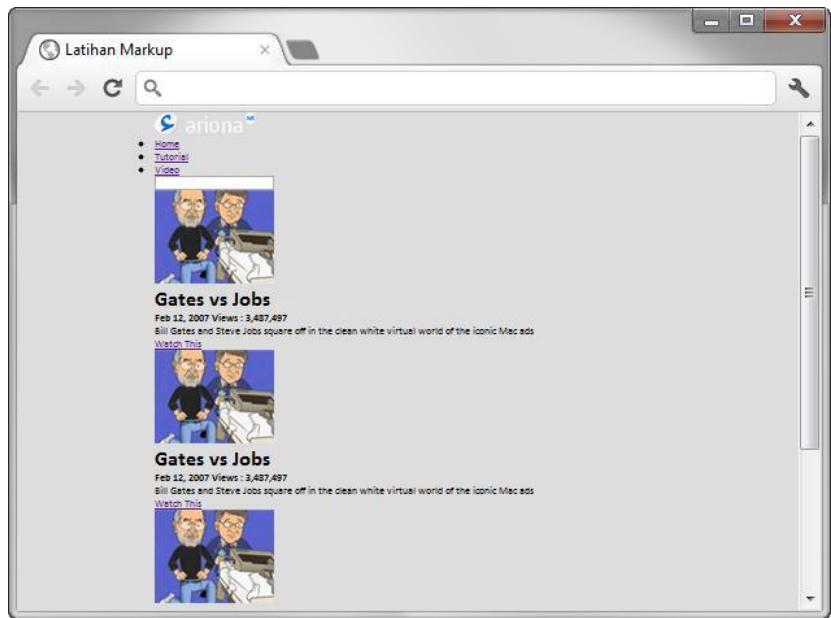
```
body{
    background-color:#DDDDDD;
}
```

20.3. Wrapper

Kita mulai dari wrapper. Seperti saya jelaskan sebelumnya, kita membutuhkan sebuah div untuk menampung seluruh isi situs karena kita akan membuatnya menjadi rata tengah. Jadi meskipun browser di zoom-in atau di zoom-out, konten tetap berada di tengah

```
#wrapper{
    width : 780px;
    margin: 0 auto;
    overflow: hidden;
}
```

Margin: 0 auto, akan memberikan jarak dari atas sebesar 0, dan untuk sisi kiri-kanannya kita beri nilai auto sehingga membuat div ini berada di tengah.



Gambar 83 Konten tetap berada ditengah meski di-zoom-in atau zoom-out

Pemberian nilai 780px untuk width bukan tanpa alasan, ini dikarenakan rata-rata pengguna internet didunia menggunakan layar dengan resolusi minimal 1024x768px jadi lebar dari situs kita berada pada ukuran aman meskipun resolusi layar yang digunakan minimal.

Sebenarnya ukuran lebar minimal standar dari sebuah desain web adalah 960px namun karena pada desain lebar dari web sebesar 780px maka kita gunakan nilai ini.

Penambahan overflow:hidden karena kita akan menambahkan floating pada daftar-artikel dan sidebar. (Baca kembali : Floating)

20.4. Header

Pada bagian header, kita akan menggunakan gambar yang telah saya siapkan, ukuran lebar gambar hanya 10px namun dengan metode pengulangan/repeat-x kita akan mendapatkan tampilan yang sama dengan yang ada di dalam desain.

Gambar 84 Hasil akhir dari Head

```

#header{
    background: url(images/header.jpg) repeat-x #555;
    height : 70px;
    width : 100%;
    padding:5px;
    margin : 20px 0;
    overflow : hidden;
}
#header img{ width:130px ; height:35px ; }

```

Kita menggunakan nilai 100% agar lebar dari header sama dengan lebar #wrapper. Dan untuk background anda bisa tebak apa yang saya lakukan. Ya, saya menggunakan CSS-shorthand untuk memanggil gambar.

Pertama kita tentukan lokasi gambar (background-image) lalu jenis pengulangan (background-repeat) dan terakhir kita tentukan background-color sebagai fallback ketika gambar tidak berhasil dimuat.

Kita juga tambahkan overflow : hidden karena di dalam header nantinya akan ada elemen yang memiliki floating (menu dan form).



Gambar 85 tampilan Header setelah diberi style

20.5. Menu

Tidak ada style yang perlu anda tambahkan pada logo, jadi kita lanjut ke bagian menu. Menu yang kita buat adalah sebuah daftar/list yang secara-default ditampilkan menurun.

Kita dapat membuat daftar tersebut menjadi rata-kanan dengan teknik floating.

```

#menu-utama{
    margin-top:3px;
    list-style:none;
}

```

Kita beri jarak antara menu dengan logo sebesar 3px, dan property list-style:none berfungsi untuk menghilangkan lingkaran didepan setiap list.

Selanjutnya kita akan membuat list/daftar menjadi rata-kiri/sejajar dengan menggunakan properti display:inline :

```
#menu-utama li{  
    float:left;  
    padding:8px 20px;  
    border-right:1px solid #555555;  
}
```

Warna teks pada menu (biru) dan garis bawah berasal dari tag <a> jadi kita ubah warna dan menghilangkan garis bawahnya.

```
#menu-utama li a{  
    color : #ffffff;  
    text-decoration:none;  
}
```



Gambar 86 Penampilan Menu utama setelah diberi style

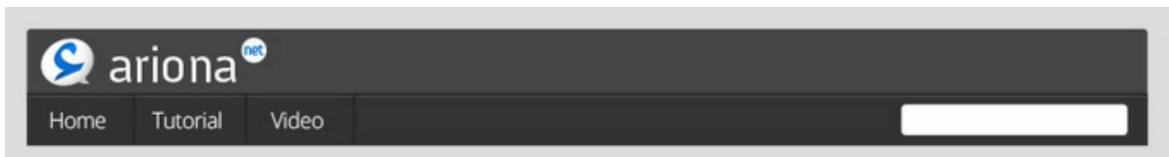
20.6. Form

Bagian form pencarian ini sangat sederhana, kita hanya akan menentukan tinggi dan meratakanankan form ini.

```
#header form{  
    float:right;  
    margin-top:7px;  
    margin-right: 10px;  
}
```

Selanjutnya kita hilangkan border bawaan dari tag input, dan kita tentukan tinggi dari input ini.

```
#header form input{  
    border:none;  
    height:20px;  
}
```



Gambar 87 Penampilan akhir header

20.7. Daftar Artikel

Selanjutnya kita beri style pada #daftar-artikel. Yang akan kita tentukan adalah lebar/width dari div ini.

```
#daftar-artikel{  
    width:470px;  
    float:left;  
}
```

The screenshot shows a news article titled "Gates vs Jobs" from February 12, 2007, with 3,487,497 views. The article features a cartoon illustration of Bill Gates and Steve Jobs. To the right of the article, there is a sidebar with the heading "Who we are?" containing a brief description of Ariona.net. Below this is a section titled "Artikel berdasarkan kategori" with a list of categories: HTML dan CSS, Javascript, Photoshop, Inspirasi, Berita, and Off Topic. At the bottom of the sidebar, it says "Ariona.net copyright 2012 by Ariona, Rian".

Gambar 88 Pengaturan Width dan rata kiri pada Daftar Artikel

20.8. Artikel

Jika anda perhatikan, setiap div artikel memiliki background putih dan antar artikel memiliki jarak.

```
.artikel{  
    background:#ffffff;  
    padding:10px;  
    margin-bottom:20px;  
    overflow:hidden;  
}
```

Perhatikan juga, gambar dari setiap artikel bisa kita setting dengan floating agar sama dengan desain yang diberikan (karena gambar memiliki nilai float, maka kita tambahkan overflow:hidden pada .artikel)

```
.artikel img{  
    float:left;  
    margin-right:10px;  
    width:200px;  
    height:165px  
}
```

Tak ada yang perlu dirubah untuk judul artikel. Yang perlu kita beri style adalah tanggal dan view. Kita akan menambahkan warna background pada kedua bagian tersebut.

```
.artikel b{ color:#616060;  
background:#DDDDDD;  
padding:3px;  
font-size:12px;  
font-weight:bold;  
margin-right:10px;  
}
```

Karena kedua elemen tersebut menggunakan tag b, maka kita tinggal menambahkan selector tersebut. Kita juga mengatur ulang ukuran font dan ketebalan dari font tersebut.

Beri jarak antar sinopsis dengan tanggal dan link di bawahnya.

```
.artikel p{  
    margin:10px 0;  
}
```

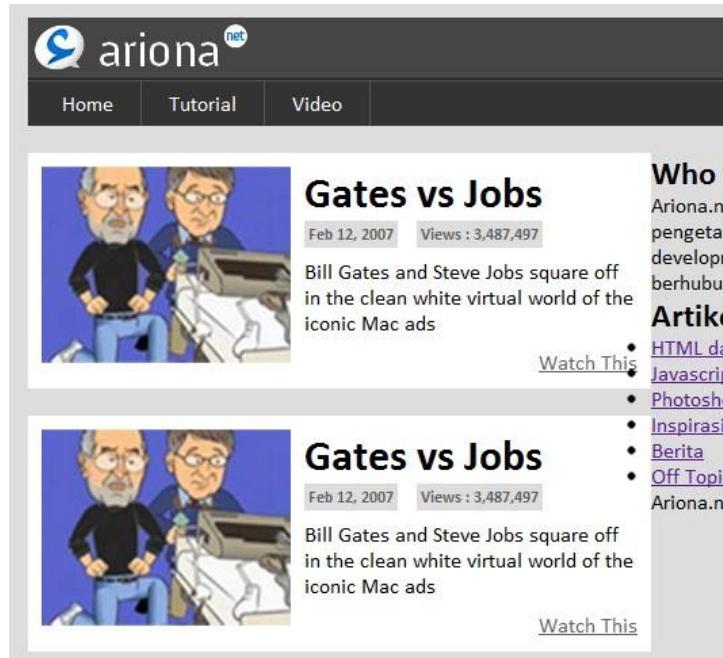
Terakhir dari bagian artikel ini, adalah penembatan link “watch this” ke sebelah kiri.

```
.artikel a{  
    display:block;  
    text-align:right;  
    color:#636363;  
}
```

Kita menggunakan metode lain, selain menggunakan floating. Yaitu dengan text-align. Namun pertama-tama kita harus mengatur elemen tersebut menjadi block.



Ukuran lebar dari elemen adalah lebar dari konten/text yang terdapat di dalamnya. Dengan menambahkan `display:block` maka elemen tersebut akan ditampilkan secara block/full width.



Gambar 89 Pengaturan link Watch This

20.9. Sidebar

Kita lanjut ke bagian Sidebar, pertama-tama kita tentukan lebar dari sidebar ini dan mengatur posisinya menjadi rata-kanan.

```
#sidebar{  
    width:270px;  
    float:right;  
}
```

Isi dari sidebar sangat mudah untuk disesuaikan dengan desain, kita mulai dari judul setiap item dalam sidebar.

```
#sidebar h2{  
    font-size:18px;  
    margin:10px 0;  
}
```

Lalu kita tentukan jarak perbaris dari text.

```
#sidebar p{  
    line-height:20px;  
    color : #636363;  
}
```



Gambar 90 Pemberian style pada sidebar

Yang perlu kita beri style selanjutnya adalah daftar kategori yang terlalu menjorok ke kiri. Kita dapat merubahnya dengan menambahkan margin-left pada tag ul.

```
#sidebar ul{  
    margin-left:10px;  
}
```

Sama halnya dengan menu utama, warna biru dan garis bawah berasal dari tag link. Jadi style yang diberikan tidak jauh berbeda.

```
#sidebar ul li a{  
    color:#636363;  
    text-decoration:none;  
}
```

20.10. Footer

Terakhir kita akan menggarap bagian footer. Perhatikan bahwa lokasi footer berada pada sidebar. Ini disebabkan oleh floating yang kita tambahkan pada daftar-artikel dan sidebar.

Untuk memperbaikinya kita bisa menggunakan clear:both; pada footer.

```
#footer{ clear:both;  
    margin:20px 0;  
    text-align:center;  
    background-color: #f0f0f0;  
    padding:5px;  
    color:#555555;  
}
```

Saya yakin anda sudah mengerti mengapa saya menambahkan property-property tersebut, jika masih bingung silahkan baca kembali bab-bab sebelumnya.

Home Tutorial Video



Gates vs Jobs

Feb 12, 2007 Views : 3,487,497

Bill Gates and Steve Jobs square off in the clean white virtual world of the iconic Mac ads

[Watch This](#)



Gates vs Jobs

Feb 12, 2007 Views : 3,487,497

Bill Gates and Steve Jobs square off in the clean white virtual world of the iconic Mac ads

[Watch This](#)



Gates vs Jobs

Feb 12, 2007 Views : 3,487,497

Bill Gates and Steve Jobs square off in the clean white virtual world of the iconic Mac ads

[Watch This](#)

Who we are?

Ariona.net adalah tempat anda berbagi pengetahuan seputar web design dan web development serta segala informasi yang berhubungan dengan internet

Artikel berdasarkan kategori

- HTML dan CSS
- Javascript
- Photoshop
- Inspirasi
- Berita
- Off Topic

Ariona.net copyright 2012 by Ariona, Rian

Gambar 91Hasil Akhir Latihan 1

Bab 21

Studi Kasus 3 – Slicing

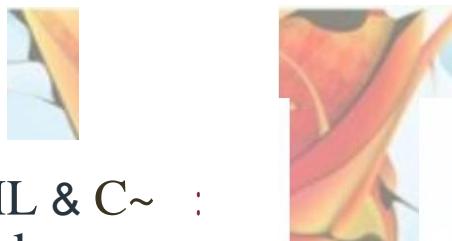
Pada bab ini kita kembali melakukan latihan Konversi desain web ke dalam HTML & CSS, namun dengan desain web yang berbeda, kita juga akan mempelajari bagaimana teknik slicing gambar-gambar yang diperlukan dalam desain web tersebut untuk bagian-bagian tertentu seperti background, logo, dan slide.

Yang akan kita konversi kali ini adalah desain web bertema minimalis. Jika diperhatikan, desain kali ini lebih sederhana dibandingkan desain sebelumnya.

Gambar 92 Desain Web untuk Latihan Slicing

Mini ml

Home Portfolio Blog Contact



Belajar HTML & C~ :
Tutorial fundamen
mempelajari HTML & CSS

B galman c r m nj d1 seorang w bd gn r,

Belajar HTML

Pelajari terlebih dahulu apa itu HTML, bagaimana cara pembuatan file HTML, Tag-tag HTML yang ada beserta fungsinya

BelajarCSS

Pelajari apa itu CSS, hubungan antara CSS dengan HTML, bagaimana cara penggunaan CSS untuk mengatur layout HTML.

BuatW bstt

Setelah mempelajari HTML & CSS saatnya untuk menerapkan keduanya dalam kasus konversi design web menjadi file HTML & CSS

Agar lebih mudah dalam proses markup, saya buatkan versi wireframe untuk desain web tersebut :



Gambar 93 Wireframe dari Desain Web

Dengan begitu kita bisa menentukan bagian-bagian yang akan kita buat div. Selanjutnya saya juga sudah menambahkan beberapa keterangan tentang nama-nama div yang akan digunakan:



Gambar 94 Rencana pemberian nama elemen

Dengan begitu pasti anda sudah memiliki bayangan akan apa yang anda tulis bukan? Ya, mari kita mulai ke dalam proses markup HTML.

21.1. Penulisan Markup HTML

Sebelum memulai proses konversi, buat folder baru untuk latihan konversi kita kali ini dengan nama “Latihan Slicing” atau terserah anda. Buat lagi satu folder dengan nama “images” untuk menyimpan gambar-gambar pendukung hasil slicing nanti.

wrapper

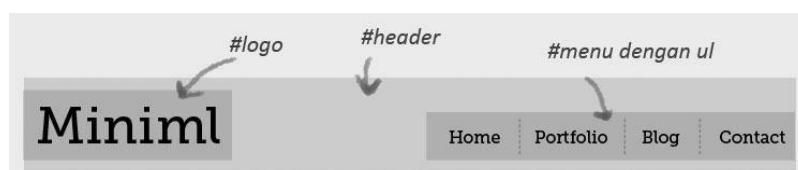
Seperti kasus sebelumnya, seluruh konten website yang ada ditampung dalam sebuah div dengan id wrapper, jadi sebagai permulaan kita buat markup standar HTML beserta sebuah div dengan id wrapper:

```
1. <!doctype HTML>
2. <HTML>
3.   <head>
4.     <title>Miniml</title>
5.   </head>
6.   <body>
7.     <div id=,wrapper'>
8.
9.     </div>
10.   </body>
11. </HTML>
```

Kode-kode selanjutnya ditulis di dalam div wrapper ini.

Header

Jika anda perhatikan, bagian header memiliki dua elemen, logo dan menu. Untuk logo kita bisa gunakan heading level 1 (h1) atau dengan gambar, namun kali ini kita coba dengan menggunakan Heading level 1 saja. Dan untuk menu kita gunakan unordered list (ul).



Gambar 95 Header

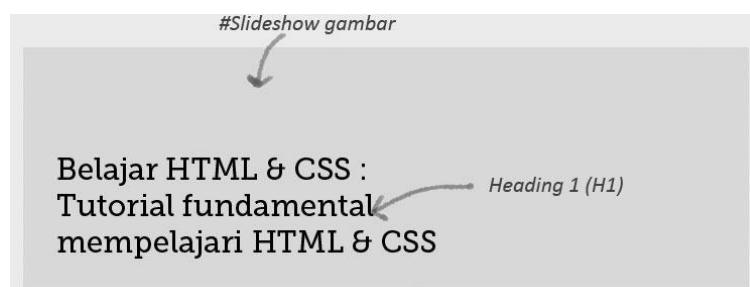
```

1. <div id="header">
2.   <h1 id="logo">Miniml</h1>
3.   <ul id="menu">
4.     <li><a href="#">Home</a></li>
5.     <li><a href="#">Portfolio</a></li>
6.     <li><a href="#">Blog</a></li>
7.     <li><a href="#">Contact</a></li>
8.   </ul>
9. </div>

```

Slideshow

Bagian selanjutnya adalah slideshow, bagian ini akan menampilkan gambar-gambar secara bergantian (slideshow). Elemen di dalamnya antara lain gambar-gambar slide dan sebuah heading untuk judul setiap slide disertai dengan link.



Gambar 96 Slideshow

```

1. <div id="slideshow">
2.   
3.   <h1><a href="#">Belajar HTML & CSS : Tutorial fundamental mempelajari HTML dan CSS</a></h1>
4. </div>

```

Untuk sementara kita gunakan gambar statis terlebih dahulu, untuk membuat slideshow ini berfungsi sebagai mestinya kita akan bahas di bagian jQuery selanjutnya.

Langkah



Gambar 97 Layout Kolom untuk bagian #Langkah

Bagian ini berisi langkah-langkah (sesuai dengan desain) secara konsep, bagian ini memiliki layout tiga kolom. Untuk itu kita gunakan div dengan class kolom. Jadi kira-kira kodennya menjadi seperti berikut :

```
1. <div id=,langkah'>
2.     <h2>Bagaimana menjadi seorang Web Designer?</h2>

3.     <div class=,kolom'>
4.         <h3>Belajar HTML</h3>
5.         <p>Pelajari terlebih dahulu apa itu HTML, bagaimana cara pembuatan file HTML,
   Tag-tag HTML yang ada beserta fungsinya</p>
6.     </div>

7.     <div class=,kolom'>
8.         <h3>Belajar CSS</h3>
9.         <p>Pelajari apa itu CSS, hubungan antara CSS dengan HTML, bagaimana cara
   penggunaan CSS untuk mengatur layout HTML</p>
10.    </div>

11.    <div class=,kolom'>
12.        <h3>Buat Website</h3>
13.        <p>Setelah mempelajari HTML & CSS saatnya untuk menerapkan keduanya dalam
   kasus konversi design web menjadi file HTML & CSS</p>
14.    </div>
15. </div>
```

Footer

Bagian terakhir adalah footer, pada bagian ini terdapat dua element, yang pertama adalah text copyright (seperti biasa) dan menu footer yang akan kita buat dengan unordered list – ul (lagi).



Gambar 98 Footer

```
<div id =,footer'>
    <p>&copy; copyright 2012 by ariona.net</p>
    <ul id=,footer-menu'>
        <li><a href=,#'>Tentang</a></li>
        <li><a href=,#'>Blog</a></li>
        <li><a href=,#'>twitter</a></li>
        <li><a href=,#'>kontak</a></li>
    </ul>
</div>
```

Tidak sulit bukan untuk membuat markup HTML dari desain yang diberikan? Yang perlu anda lakukan adalah membayangkan layout dan pembagian masing-masing elemen yang akan digunakan lalu mengkodekannya menjadi HTML.

Kode lengkap dari proses Markup yang telah kita lakukan adalah seperti berikut :

```
1.  <!doctypeHTML>
2.  <HTML>
3.      <head>
4.          <title>Miniml</title>
5.      </head>
6.      <body>
7.          <div id=,wrapper>
8.              <div id=,header>
9.                  <h1 id=,logo>Miniml</h1>
10.                 <ul id=,menu>
11.                     <li><a href=,#>Home</a></li>
12.                     <li><a href=,#>Portfolio</a></li>
13.                     <li><a href=,#>Blog</a></li>
14.                     <li><a href=,#>Contact</a></li>
15.                 </ul>
16.             </div>
17.             <div id=,slideshow>
18.                 <img src=,images/slide.jpg alt=,slide />
19.                 <h1><a href=,,>Belajar HTML & CSS : Tutorial fundamental
mempelajari HTML dan CSS</a></h1>
20.             </div>
21.             <div id=,langkah>
22.                 <h2>Bagaimana menjadi seorang Web Designer?</h2>
23.                 <div class=,kolom>
24.                     <h3>Belajar HTML</h3>
25.                     <p>Pelajari terlebih dahulu apa itu HTML,bagaimana
cara pembuatan file HTML, Tag-tag HTML yang ada beserta
fungsiyanya</p>
26.                 </div>
27.                 <div class=,kolom>
28.                     <h3>Belajar CSS</h3>
29.                     <p>Pelajari apa itu CSS, hubungan antara CSS dengan
HTML,bagaimana cara penggunaan CSS untuk mengatur
layout HTML</p>
30.                 </div>
31.                 <div class=,kolom>
32.                     <h3>Buat Website</h3>
33.                     <p>Setelah mempelajari HTML & CSS saatnya untuk
menerapkan keduanya dalam kasus konversi design web
menjadi file HTML & CSS</p>
34.                 </div>
35.             </div>
36.             <div id =,footer>
37.                 <p>&copy; copyright 2012 by ariona.net</p>
38.                 <ul id=,footer-menu>
39.                     <li><a href=,#>Tentang</a></li>
40.                     <li><a href=,#>Blog</a></li>
41.                     <li><a href=,#>twitter</a></li>
42.                     <li><a href=,#>kontak</a></li>
43.                 </ul>
44.             </div>
45.         </div>
46.     </body>
47. </HTML>
```

Simpan dengan nama “index.html” di dalam folder yang kita buat sebelumnya “Latihan Slicing”.

21.2. Slicing

Slicing adalah proses pemotongan gambar-gambar yang akan digunakan dalam website agar sesuai dengan desain yang diinginkan. Gambar-gambar tersebut meliputi background, logo, slide, icon dan gambar pendukung lainnya.

Desain Website tersebut saya sertakan dalam File penyerta buku ini, untuk membukanya anda membutuhkan software Adobe Photoshop. Buka file **miniml-website-design.psd** dan kita mulai proses slicing.

Ada tiga cara untuk melakukan slicing dengan photoshop:

1. Slice Tool : Cara ini lebih efektif untuk melakukan proses slicing secara batch/sekaligus. Kita tinggal menyeleksi bagian-bagian yang akan di slice dengan tool ini kemudian pilih menu save for web & device.



Gambar 99 Contoh Penggunaan Slice Tool

2. Cropping Tool : Buat seleksi ada bagian yang ingin diambil, crop, save menjadi gambar, lalu kembalikan ke kondisi awal dengan CTRL+ALT+Z. Cukup ribet namun terkadang menguntungkan.



Gambar 100 Slicing dengan teknik Cropping

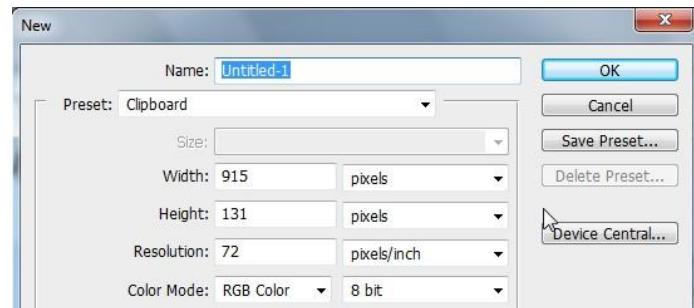
3. Manual Slicing : Cara ini adalah cara manual namun lebih efektif. Mengingat hanya bagian-bagian tertentu yang akan kita ambil, jadi kita cukup memilih layer yang dibutuhkan, copy, buat file baru, paste dan save. Dan Cara ini yang akan kita gunakan untuk melakukan slicing pada desain website yang sedang kita kerjakan.

Pertama-tama kita akan mengambil background, tepatnya di bagian atas gambar. Pada window layer pilih layer dengan nama background-grunge, lalu **tekan CTRL** dan **Klik** pada **Layer Thumbnail**. Langkah ini akan menyeleksi bagian yang terdapat dalam layer tersebut.



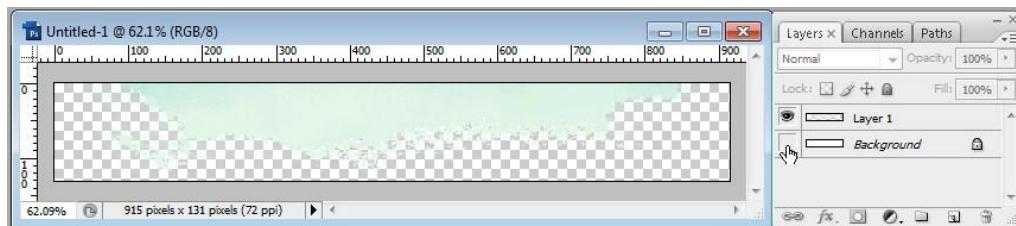
Gambar 101 Ctrl+Click untuk menyeleksi konten dari layer

Selanjutnya tekan Kombinasi tombol **CTRL+C** untuk meng-*copy* isi dari layer tersebut. Buat dokumen/canvas baru dengan menekan **CTRL+N (File > New)**. Ukuran dari canvas baru akan sesuai dengan isi dari seleksi yang kita *copy*.



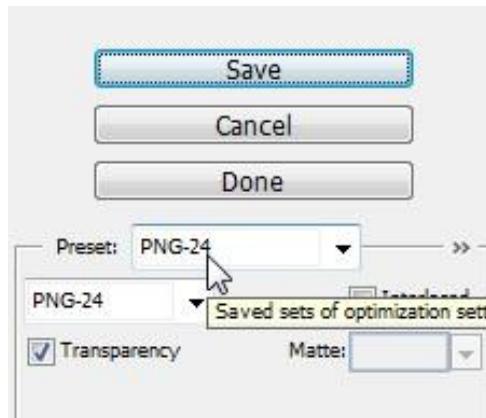
Gambar 102 Membuat file baru untuk hasil slice

Tekan **OK**, lalu tekan tombol **CTRL+V** untuk mem-*paste* apa yang kita *copy* sebelumnya. Hilangkan tanda “mata” pada layer background untuk menyembunyikan layer ini karena kita akan membuatnya transparan.



Gambar 103 Hilangkan icon mata untuk menyembunyikan layer

Selanjutnya kita tinggal menekan tombol **CTRL+ALT+SHIFT+S** untuk menyimpan sebagai gambar web/device atau pilih menu “**File > Save for Web & Devices**”. Pilih format PNG agar gambar yang dihasilkan dapat memiliki area transparan. Tekan **Save** untuk menyimpan gambar, simpan di folder Images yang ada di dalam folder latihan yang kita buat dengan nama “**background.png**” (tanpa tanda kutip).

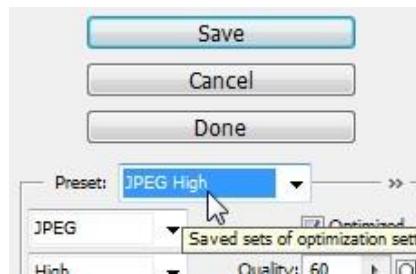


Gambar 104 PNG-24 mendukung transparansi pada gambar



Jika gambar yang dibutuhkan memiliki area transparan, selalu pilih preset **PNG-24**

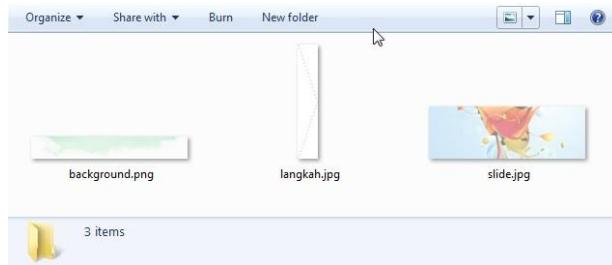
Selanjutnya kita membutuhkan gambar untuk Slideshow. Pada Window Layer, Buka folder Slideshow lalu klik layer Slide. Lakukan proses slicing seperti langkah sebelumnya, **CTRL+Klik, Copy, Buat Dokumen Baru, Paste**. Kali ini pilih *preset JPEG High*, dan simpan di folder yang sama seperti sebelumnya dengan nama “**slide.jpg**”.



Gambar 105 Preset JPG High untuk kualitas gambar tinggi

Gambar terakhir yang kita butuhkan adalah tanda panah pada bagian “Langkah”. Cari Layer “Separator” di dalam folder langkah lalu lakukan langkah slicing seperti sebelumnya. Simpan dengan nama “**langkah.jpg**”. Untuk bagian ini cukup satu saja yang di slice, OK.

Kini dalam folder images, kita mempunyai 3 gambar pendukung, background.png, slide.jpg dan langkah.jpg.



Gambar 106 File-file hasil slicing

21.3. Pemberian Style

Lanjut ke bagian paling menyenangkan (menurut saya), pemberian Style atau CSS. Buat file baru dengan nama style.CSS, simpan di folder yang sama dengan file HTML yang kita buat sebelumnya.

CSS Reset

Seperti biasa, kita buat dulu CSS Reset untuk mereset margin dan padding bawaan dari tiap browser.

```
*{  
    margin:0;  
    padding:0;  
    font-family:'segoe ui',arial,sans-serif;  
}
```

Body

Untuk body kita akan tambahkan warna background dan gambar background yang telah kita *slice*.

```
body{  
    background-color:#f8fbf4;  
    background-image: url('images/background.png');  
    background-repeat: no-repeat;  
    background-position: center top;  
    font-size: 14px;  
}
```

Ingat, kode diatas bisa anda singkat dengan menggunakan CSS-shorthand untuk background seperti berikut :

```
background:url('images/background.png') center top no-repeat #f8fbf4;
```

Saya uraikan agar anda mengerti masing-masing bagianya.

Heading

Untuk heading, kita akan menggunakan jenis font yang berbeda dengan font keseluruhan. Font yang digunakan adalah museo slab. Anda dapat menginstallnya dari folder resources yang disertakan dalam buku ini.

```
h1,h2,h3,h4,h5,h6{  
    font-family:"Museo Slab 500", impact,arial,sans-serif;  
    font-weight: normal;  
}
```

Wrapper

Dari desain, ukuran wrapper adalah 720px. Anda bisa menggunakan Ruler Tool untuk mengukur lebar wrapper dengan patokan gambar slide.

```
#wrapper{  
    width:720px;  
    margin:0 auto;  
}
```

Masih ingat fungsi `margin:0 auto;` ? Ya, property ini berfungsi untuk membuat #wrapper menjadi rata tengah meski di zoom-in atau di zoom-out.

Header

Di dalam header kita akan ada 2 elemen yang memiliki floating. Jadi gunakan solusi floating pada div header.

```
#header{  
    overflow: hidden;  
    margin-bottom: 20px;  
    padding:30px 0  
}
```

Untuk Logo, karena kita menggunakan h1 maka yang kita perlukan adalah merubah font sesuai dengan yang ada dalam desain. Font yang digunakan adalah **Museo Slab 500**(font disertakan di dalam File penyerta buku ini).

```
#logo{  
    font-family:'Museo Slab 500', impact, sans-serif;  
    float:left;  
    font-size:40px;  
}
```

Kemudian untuk menu, kita buat rata kanan, lalu membuat list-item menjadi satu baris serta memberikan sentuhan border sebagai separator :

```
#menu{  
    float:right;  
    list-style:none;  
    margin-top:10px;  
}  
  
#menu li{ display: inline }  
#menu li a{  
    padding:5px 10px;  
    border-right: 1px dashed #aeb59c;  
    text-decoration:none;  
    color:black;  
}  
#menu li:last-child a{ border-right:none }
```

Slideshow

Untuk slideshow kita akan menerapkan positioning pada elemen #slideshow dan h1.

```
#slideshow{  
    position:relative;  
}
```

Kita set position menjadi relative karena h1 akan kita buat menjadi absolute. Sehingga pergerakan posisi h1 tergantung/relative pada #slideshow.

```
#slideshow h1{  
    position:absolute;  
    bottom:20px;  
    left:20px;  
    font-weight:normal;  
}
```

Karena di dalam heading kita menambahkan link/anchor kita juga harus memberikan style pada link tersebut agar tidak ditampilkan dengan style bawaan.

```
#slideshow h1 a{  
    color:black;  
    text-decoration:none;  
    font-family:"Museo Slab 500",impact,sans-serif;  
    font-weight: normal;  
}
```

Layout Kolom untuk #langkah

Seperti yang telah saya singgung, pada bagian #langkah kita akan menerapkan layout kolom. Caranya sangatlah mudah. Kita hanya akan menambahkan floating pada .kolom.

```
#langkah{  
    overflow:hidden;  
    background:white;  
    margin:20px 0;  
}  
#langkah h2{ padding:10px }  
#langkah .kolom{  
    width:220px;  
    margin:10px;  
    padding:0 30px 0 0;  
    float:left;  
    background:url('images/langkah.jpg') right top no-repeat;  
}  
#langkah .kolom:last-child{background:none;}
```

Pertama-tama kita tambahkan overflow:hidden pada #langkah, karena di dalamnya akan ada elemen yang memiliki property float (baca lagi Masalah Floating).

Selanjutnya kita tentukan width/lebar dari .kolom, tentukan padding-right untuk background (tanda siku).

Terakhir kita hilangkan background pada kolom paling akhir dengan menggunakan pseudo-class (.kolom:last-child)

Footer

Untuk bagian #footer kita hanya memerlukan border-top dan membuat menu ditampilkan satu baris serta merata kirikan elemen tersebut.

```
#footer{  
    border-top:1px dashed #aeb59c;  
    padding:10px;  
}  
#footer p{ float:left }  
#footer-menu{  
    float:left;  
    margin-left:30px;  
    list-style:none;  
}  
  
#footer-menu li{  
    display:inline;  
    margin-right:20px;  
}  
#footer-menu li a{  
    color:#777;
```

```
    text-decoration:none;  
}
```

Akhirnya proses Konversi telah selesai, jika anda membukanya di browser anda akan mendapatkan tampilan yang sama dengan desain. Namun masih ada yang kurang, Kita belum membuat Slideshow berfungsi sebagaimana mestinya.

Mengingat HTML & CSS tidak memiliki fungsionalitas untuk membuat sebuah slideshow⁸, maka bersama dengan ini saya akan memperkenalkan Javascript pada Anda.

⁸ Pada kenyataannya kita bisa membuat slideshow dengan menggunakan teknologi CSS3, namun mengingat teknik tersebut adalah teknik lanjut.

Bab 22

Javascript

22.1. Pengenalan Javascript

Javascript adalah bahasa pemrograman web yang berjalan disisi Client/Browser. Javascript biasa digunakan untuk memanipulasi element-element HTML dan menambahkan Style secara otomatis atau lebih sederhananya membuat dokumen HTML menjadi lebih Interaktif.

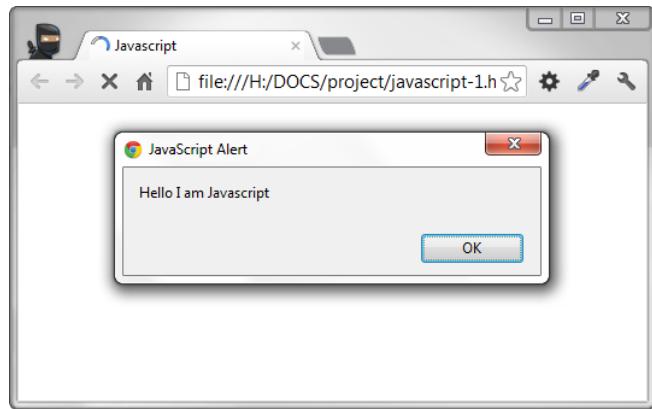
Untuk menambahkan Script javascript pada dokumen HTML, anda dapat menggunakan tag script dan diletakkan di dalam head atau di area manapun di dalam tag body.

```
...
<head>
    <title>Javascript</title>
    <script type='text/javascript'>
        ... Script javascript diketik disini...
</script>
</head>
...
```

Saya akan memberikan contoh sederhana dari javascript ini, tambahkan script berikut pada latihan sebelumnya atau anda dapat membuat file HTML baru dengan nama file latihan14.html.

```
...
<head>
    <title>Javascript</title>
    <script type='text/javascript'>
        alert('Hello I am Javascript');
    </script>
</head>
...
```

Script tersebut akan menampilkan MessageBox dengan teks “Hello I am Javascript” ketika file HTML dibuka.



Gambar 107 Kotak pesan dengan javascript

Berikut contoh lain dari penggunaan Javascript yang bias anda lihat dalam file latihan15.html:

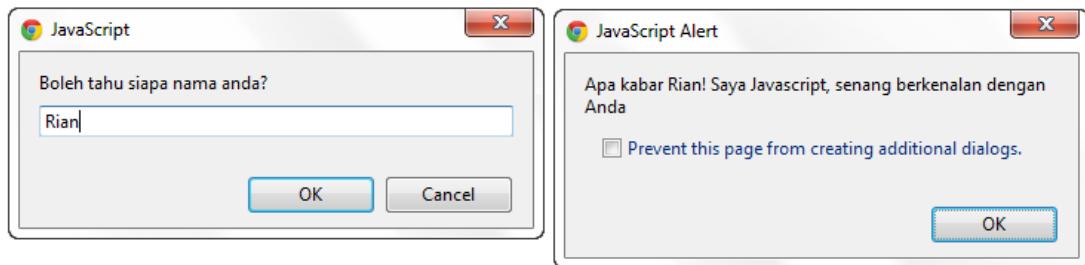
```
1. <script type="text/javascript">
2.     var nama;
3.     nama=prompt('Boleh tahu siapa nama anda?');
4.     if(nama){
5.         alert('Apa kabar '+nama+'! Saya Javascript, senang berkenalan dengan Anda');
6.     }else{
7.         alert('Hhmm sepertinya anda tidak mau berkenalan dengan saya.');
8.     }
9. </script>
```

Penjelasan Script :

Pertama, kita buat sebuah variabel nama. Kemudian kita meminta Javascript untuk menanyakan nama pengunjung lalu menyimpannya di dalam variabel nama.

Selanjutnya, Kita memilih, apakah variabel nama terisi (pengunjung mengetikkan nama), maka akan ditampilkan Message Box dengan teks “Apa kabar (Nama yang dimasukkan)!, Senang berkenalan dengan Anda”.

Jika pengunjung tidak memasukkan nama (langsung mengklik ok atau cancel) maka akan ditampilkan message box “Hhmm sepertinya anda tidak mau berkenalan dengan saya.”.



Gambar 108 Contoh penggunaan kondisi

Script tersebut adalah contoh sederhana dari penggunaan Javascript, Anda dapat mempelajarinya secara lanjut di <http://www.w3schools.com/js/>.

22.2. jQuery

jQuery, adalah library atau sekumpulan kode javascript siap pakai untuk memanipulasi elemen HTML dengan mudah dan cepat. Anda dapat mendownload jQuery di situs resminya <http://www.jquery.com>.

Untuk menggunakan jQuery, Anda diharuskan menyertakan jQuery dengan menggunakan tag Script.

```
<script type='text/javascript' src='jquery.js'></script>
```

Berikut ini adalah contoh sederhana penggunaan jQuery untuk memanipulasi HTML, tambahkan script berikut pada file latihan anda.

```
<script type='text/javascript' src='jquery.js'></script>
<script>
    $(document).ready(function(){
        $('body').css('background-color','red');
    });
</script>
```

Penjelasan Script :

```
$(document).ready(function(){
    ...
});
```

Block kode ini memastikan script dijalankan setelah file HTML selesai dimuat.

```
$(body).css('background-color','red');
```

Script tersebut akan merubah warna body dengan menambahkan atribut style yang memiliki nilai “background-color:red”.

22.3. jQuery Plugin

jQuery Plugin adalah plugin/suplemen untuk menambah fungsionalitas dari jQuery. Plugin ini biasanya dibuat untuk memudahkan pengguna dalam membuat suatu aplikasi. Salah satunya adalah membuat Slider, Slideshow dan lain sebagainya.

Anda tinggal memanggil library plugin jquery tersebut seperti anda memanggil file javascript, selanjutnya tinggal memanggil fungsi yang disediakan oleh library tersebut.

Sebagai contoh, ada sebuah plugin jQuery Slider dan kita akan menerapkannya pada elemen HTML yang memiliki id slider, Plugin tersebut menyediakan fungsi “slider()” untuk membuat markup HTML kita menjadi sebuah slider. jadi anda tinggal mengetikkan script berikut :

```
$(, #slider').slider();
```

Anda dapat mencari plugin-plugin jquery yang anda butuhkan di internet atau anda dapat mencari sebagian plugin jquery di <http://www.jquery4u.com/plugins/>.



Gambar 109 jQuery4u telah merangkum sebagian plugin jQuery dari A-Z

Bab 23

Studi Kasus 4 – Penambahan jQuery Slider

Terkait dengan Latihan Slicing sebelumnya kita akan menambahkan fungsionalitas pada Slider agar berjalan sebagaimana mestinya. Dalam kasus ini kita akan menggunakan Plugin Basic jQuery Slider yang bisa anda download di <http://basic-slider.com/>.



Sebelum menggunakan plugin jQuery, sebagiknya anda Baca Documentation/Panduannya terlebih dahulu

Cara penggunaan basic-slider

Dari Dokumentasi basic-slider yang bisa anda baca di <http://basic-slider.com/documentation/>. Berikut ini adalah cara penggunaan dari plugin tersebut :

1. Download pluginnya lalu sertakan dalam file HTML anda.

```
<!-- Sertakan file jQuery -->
<script src="js/libs/jquery-1.6.2.min.js"></script>

<!-- Sertakan plugin basic-slider SETELAH menyertakan jquery -->
<script src="js/basic-jquery-slider.js"></script>

<!-- Include the basic styles -->
<link type="text/css" rel="stylesheet" href="CSS/basic-jquery-slider.CSS"/>
```

2. Markup HTML untuk slider mengikuti struktur yang ditentukan oleh basic-slider seperti berikut :

```
<div id="slideshow-anda">
    <ul class="bjqs">
        <li><!-- Isi Slide Anda --></li>
        <li><!-- Dapat diisi oleh apapun --></li>
    </ul>
</div>
```

3. Jika sudah melakukan poin 1 dan 2 di atas, baru kita panggil fungsinya seperti berikut :

```
$(document).ready(function(){
    $('#id-slideshow-anda').bjqs({
        'width': 940,
        'height': 340,
        'showMarkers': true,
        'showControls': true,
        'centerMarkers': false
    });
});
```

Di dalam fungsi .bjqs terdapat beberapa parameter untuk pengaturan slider, Anda dapat membaca penjelasan tiap parameternya di link yang saya sebutkan di atas.

Penerapan pada Studi Kasus ke-3

Pada Studi kasus ke-4 ini, kita akan membuat slider yang terdapat pada studi kasus ke-3 berfungsi sebagaimana mestinya yakni sebagai slider.

Perubahan Markup HTML

Sebelum kita mulai membuat slider berfungsi dengan baik, rubah markup HTML berikut:

```
1. <div id=,slideshow>
2.   <img src=,images/slide.jpg` alt=,slide` />
3.   <h1><a href=,,>Belajar HTML & CSS : Tutorial fundamental mempelajari HTML dan
CSS</a></h1>
4. </div>
```

Menjadi seperti di bawah ini :

```
1. <div id=,slideshow>
2.   <ul class=,bjqs>
3.     <li>
4.       <img src=,images/slide.jpg` alt=,slide` />
5.       <h1><a href=,,>Belajar HTML & CSS, :
Tutorial fundamental mempelajariHTML dan
CSS</a></h1>
6.     </li>
7.     <li>
8.       <img src=,images/slide2.jpg` alt=,slide` />
9.       <h1><a href=,,>Belajar HTML & CSS : Tutorial fundamental
mempelajariHTML dan CSS</a></h1>
```

```

10.          </li>
11.          <li>
12.              <img src=„images/slide3.jpg“ alt=„slide“ />
13.              <h1><a href=„,›Belajar HTML & CSS : Tutorial fundamental
mempelajariHTML dan CSS</a></h1>
14.          </li>
15.      </ul>
16.  </div>

```

Pada dasarkan kita akan menampilkan 3 buah slide/gambar. Kita juga akan merubah dan menambah style untuk bagian slideshow ini. Gambar-gambar tambahan bisa anda buat sendiri atau copy gambar yang terdapat di dalam folder “miniml – with slider/images”.

Class bjqs yang terdapat pada unordered list adalah ketentuan yang diberikan oleh library jquery slider yang akan kita gunakan (sudah dijelaskan sebelumnya).

Pemanggilan Library

Selanjutnya kita panggil beberapa file javascript yakni file jQuery, plugin jQuery basic-slider, dan file CSS bawaan dari jquery-basic slider.

Ubah bagian header dan tambahkan kode yang ditebalkan berikut :

```

<!DOCTYPE HTML>
<HTML>
    <head>
        <title>Miniml</title>
        <link rel=„stylesheet“ href=„style.css“>
        <link rel=„stylesheet“ href=„basic-jquery-slider.css“>
        <script type=„text/javascript“ src=„jquery.js“></script>
        <script type=„text/javascript“ src=„basic-jquery-slider.js“></script>

```

Tambahkan juga style berikut ke dalam file style.css yang kita buat sebelumnya :

```

.bjqs-next a, .bjqs-prev a {
    color: orange;
    text-decoration: none;
    background: white;
    padding: 5px 10px;
}

```

Aktivasi Plugin

Langkah terakhir adalah pemanggilan fungsi basic slider yang akan merubah markup HTML kita menjadi Slider “beneran”. Tambahkan kode berikut setelah kode pemanggilan library `basic-jquery-slider.js` di atas.

```
1. ...
2. <script type="text/javascript" src="basic-jquery-slider.js"></script>
3. <script type="text/javascript">
4.     $(document).ready(function(){
5.         $('#slideshow').bjqs({
6.             'width' : 720,
7.             'height' : 238,
8.             'showMarkers' : false,
9.             'showControls' : true
10.        });
11.    });
12. </script>
```

Parameter `width` dan `height` mengacu pada lebar dan tinggi gambar slide. Berbeda dengan contoh pemanggilan basic-slider yang telah saya jelaskan sebelumnya, saya hanya menampilkan parameter “`showControls`” untuk navigasi tiap slider dan menyembunyikan Markers dengan memberi nilai **false**.

Kini, jika anda tampilkan dalam web browser, Bagian slider memiliki link “`prev`” dan “`next`”, klik link tersebut untuk memajumundurkan slide.



Gambar 110 Link Prev dan Next untuk navigasi slider

Bab 24

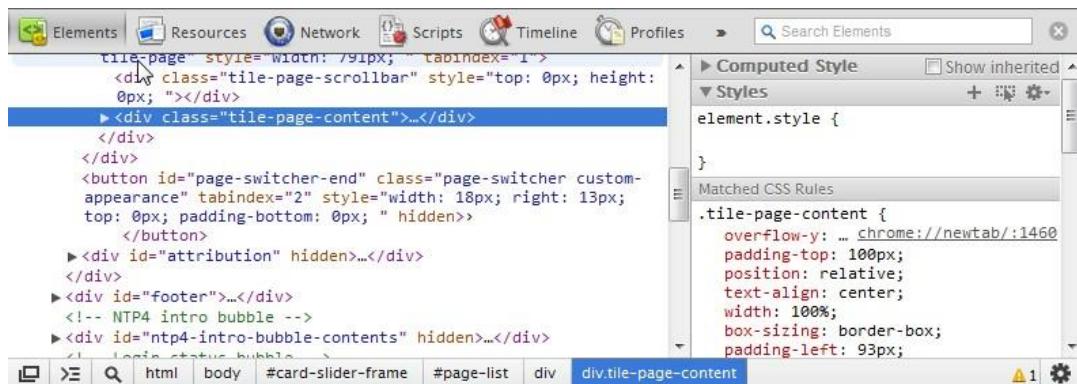
EXTRAS

24.1. Developer Tools dan Firebug

Kedua add-on ini telah saya ceritakan secara singkat di awal buku ini. Fungsi keduanya hampir sama yakni menyediakan berbagai informasi untuk Developer web termasuk informasi tentang dokumen HTML dan CSS yang menjadi konsentrasi dalam buku ini.

Tool ini biasa digunakan untuk mempelajari struktur HTML dan CSS dari web lain, misalnya ketika kita ingin mengetahui bagaimana struktur dari suatu bagian dalam web, anda tinggal klik kanan bagian tersebut, dan pilih menu Inspect Element.

Secara otomatis block HTML bagian tersebut akan ditampilkan lengkap dengan style/CSS yang diterapkan.



Gambar 111 Kode HTML beserta Style yang diterapkan

Dengan begitu anda dapat mempelajari style yang digunakan. Anda-pun dapat menambah, menghapus, dan mengeditnya dengan mengklik dua kali pada kode yang bersangkutan dan anda dapat melihat langsung perubahan yang terjadi dalam browser.

Perubahan yang anda lakukan tidak bersifat permanen, ketika browser anda refresh semuanya akan kembali seperti semula.

24.2. Zen Coding

Zen Coding adalah plugin tambahan yang tersedia untuk beberapa text editor, seperti Notepad++, Sublime text dan masih banyak lagi. Untuk daftar text editor yang didukung oleh zen coding bisa dilihat disitus *officialnya*

<http://code.google.com/p/zen-coding/>.

Untuk instalasi zen coding di Notepad++, anda cukup masuk ke menu Tools, Plugin Manager. Selanjutnya scroll sampai anda melihat Zen Coding. Beri tanda dan klik tombol install untuk mulai menginstall zen coding.

Penggunaan zen coding sangatlah mudah. Plugin ini dibuat untuk mempercepat penulisan kode HTML dan CSS, sebagai contoh perhatikan kode HTML berikut :

```
<div id=,wrapper>
    <div id=,content></div>
    <div id=,sidebar></div>
    <div id=,footer></div>
</div>
```

Untuk ukuran anda yang sudah terbiasa mengetik dengan sepuluh jari mungkin hanya membutuhkan 5 menit untuk menulis kode di atas. Lalu bagaimana jika anda tidak menguasai teknik tersebut? Mungkin akan memakan waktu 10 menit atau bahkan lebih.

Dengan zen coding, kode tersebut bisa disingkat menjadi satu baris seperti berikut :

```
#wrapper>#content+#sidebar+#footer
```

Selanjutnya anda tinggal menekan kombinasi tombol yang telah ditentukan untuk melebarkan (expand) kode tersebut, dan 5 baris kode diatas pun akan anda dapatkan. Sangat cepat bukan?

Lalu apa arti dari satu baris kode zen coding sebelunya?

#wrapper, #content , #sidebar dan #footer

#wrapper, #content , #sidebar dan #footer berarti kita membuat sebuah div dengan nama id wrapper, content, sidebar dan footer. Penambahan tanda pagar mengacu pada

penulisan selector CSS untuk id (baca kembali bab pengenalan CSS). Dengan begitu untuk penulisan sebuah div dengan nama class artikel anda tinggal menuliskan titik sebelum nama class (.artikel).

Ketika anda tidak menambahkan nama tag yang akan dibuat, maka tag yang akan dibuat adalah tag div. Jika Anda ingin membuat tag tertentu anda tinggal mengetikkan nama tag tersebut. Misal :

```
ul#menu
```

Akan menghasilkan

```
<ul id="menu"></ul>
```

Child

Selanjutnya tanda “lebih besar dari” (>) digunakan untuk membuat child dari tag sebelumnya. Pada contoh kasus di atas adalah : #wrapper>

Ini menandakan kode-kode setelah tanda “>”(yakni #content, #sidebar dan #footer) akan berada di dalam #wrapper (Child dari #wrapper).

Perulangan kode

Untuk menuliskan beberapa baris kode yang sama, anda dapat menggunakan operator *n dimana n adalah jumlah yang diinginkan. Contoh kasusnya adalah pembuatan list item menu.

```
<ul id="menu">
    <li><a href=""></a><li>
    <li><a href=""></a><li>
    <li><a href=""></a><li>
    <li><a href=""></a><li>
    <li><a href=""></a><li>
</ul>
```

Penulisan kode zen coding dari beberapa baris kode diatas adalah seperti berikut :

```
ul#menu>li*5>a
```



Zen Coding sangat **dianjurkan** untuk anda yang sudah **memahami** HTML dan CSS, jika anda masih belum mengerti sepenuhnya akan HTML dan CSS saya sarankan untuk mengetik kode secara manual agar anda mengingat apa yang sedang anda pelajari.

Sebenarnya masih banyak lagi tata cara penulisan kode HTML dengan bantuan Zen Coding ini, begitu pula penulisan property-property CSS, anda dapat membaca daftar singkatan zen-coding yang bisa di download dari situs resminya.

24.3. Menambahkan Dummy Text

Ketika anda membutuhkan sejumlah teks/kalimat/paragraf untuk kepentingan visual, Anda akan membutuhkan dummy text. Dummy text adalah sekumpulan kalimat/paragraf tanpa makna seperti :

`Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.`

Paragraf tersebut tidak memiliki makna namun berguna ketika kita membutuhkan konten cepat.

Untuk mendapatkan teks tersebut, jika anda menggunakan text editor sublime text, anda tinggal mengetikkan `lorem` lalu tekan tombol Tab.

Anda juga dapat membuatnya dengan menggunakan Aplikasi pengolah kata Microsoft Word, caranya dengan mengetikkan :

```
=lorem(jumlahParagraf, jumlahKalimat)
```

Lalu tekan Enter untuk mendapatkan text dummy sesuai dengan jumlah kalimat dan paragraf yang ditentukan.

Jika anda menggunakan text editor Sublime text, anda tinggal mengetikkan kata `lorem` dan tekan tab untuk mendapatkan dummy text ini.

24.4. Buat situs anda “online”

Untuk membuat halaman web yang anda buat “online” sehingga dapat diakses oleh semua orang melalui internet, anda harus menyimpan seluruh file yang anda buat di tempat penyimpanan online yang biasa disebut dengan **hosting** dan mengarahkannya dengan **domain**.

Mengenal Hosting dan Domain

Sebelumnya telah dijelaskan secara singkat bahwa Hosting adalah tempat penyimpanan online untuk file-file website anda sehingga semua orang dapat mengaksesnya di internet. sedangkan untuk mengarahkan orang-orang agar membuka halaman web anda, diperlukan sebuah alamat yang biasa disebut dengan **domain** misalnya www.facebook.com.

Untuk menyederhanakan istilah-istilah tersebut, bayangkanlah **Hosting** itu sebuah Rumah, dan domain sebagai **Alamat** dari rumah tersebut.

Lalu dimana saya bisa mendapatkan domain dan hosting ini? Jawabannya, Anda dapat menyewa keduanya. Sudah banyak jasa-jasa Hosting indonesia yang menawarkan jasa sewa Hosting dan domain dengan harga yang murah, anda dapat menyewanya perbulan atau bahkan pertahun dengan berbagai paket sesuai kebutuhan.

Tips memilih Hosting

Murah saja tidaklah cukup, jangan terpengaruh oleh harganya yang miring, tapi lakukan survey terlebih dahulu akan kinerja dari hosting yang akan kita sewa. Karena biasanya kendala dari sewa hosting sangat banyak, mulai dari server yang selalu down sehingga halaman web anda tidak bisa dibuka oleh orang lain.

Sesuaikan pemilihan paket data sesuai dengan ukuran situs anda, semakin kecil ukuran hosting yang akan disewa semakin murah pula harga yang harus anda bayar.

Tips memilih Domain

Gunakanlah nama domain yang pendek atau yang lebih mudah diingat oleh semua orang, misalnya toko-bagus.com, ilmu-komputer.com atau nama lain yang akan mudah diingat.

Harga domain tergantung dari ekstensi yang anda pilih misalnya .com , .net , .org , .co.id dan lain sebagainya.

Pada Contoh yang akan saya berikan, kita akan menyewa hosting dari salah satu jasa hosting lokal. Sebagai contoh saya akan memilih paket domain dan hosting dengan size 150mb, cukup untuk menampung seluruh situs saya. Setelah proses pembayaran selesai, kita tinggal menggunakan hosting tersebut tanpa harus bingung mengatur hosting dan domain karena semuanya sudah dilakukan oleh penyedia jasa.

Small	Medium	Large
Rp. 120.000 Per Tahun	Rp. 140.000 Per Tahun	Rp. 160.000 Per Tahun
50 MB Disk Space	100 MB Disk Space	150 MB Disk Space
4 GB Data Transfer	5 GB Data Transfer	6 GB Data Transfer
1 Addon/Park Domain	1 Addon/Park Domain	1 Addon/Park Domain
10 Email Account	10 Email Account	10 Email Account
10 Sub Domain	10 Sub Domain	10 Sub Domain
10 MySQL Database	10 MySQL Database	10 MySQL Database
Rendah Prioritas Bantuan	Rendah Prioritas Bantuan	Rendah Prioritas Bantuan
Gratis Domain	Gratis Domain	Gratis Domain
Pesan (Server ID)	Pesan (Server ID)	Pesan (Server ID)
Pesan (Server USA)	Pesan (Server USA)	Pesan (Server USA)

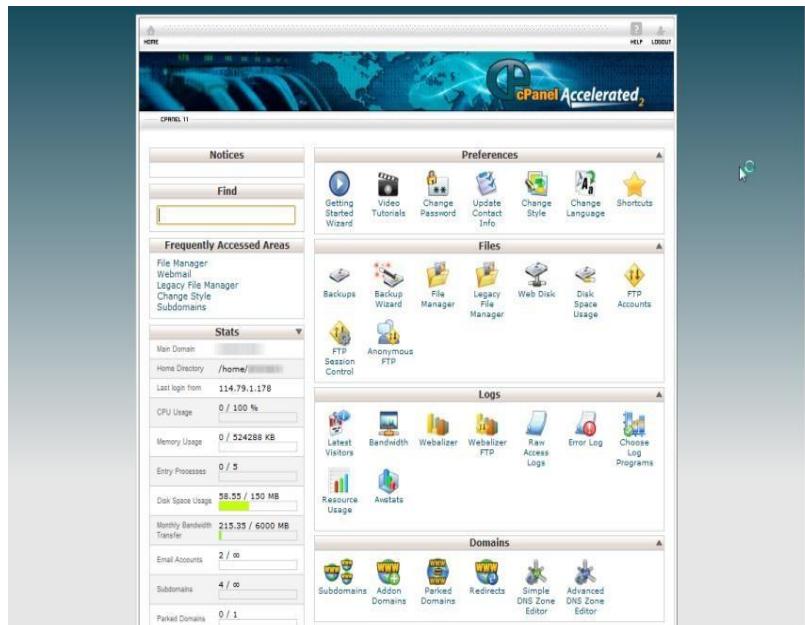
Gambar 112 Tabel harga hosting+domain di salah satu jasa hosting lokal

Pertama-tama, kita masuk ke pusat pengaturan Hosting yang biasa disebut dengan cpanel. Biasanya dapat diakses melalui alamat :www.domain-anda.com/cpanel. Andapun akan dibawa ke halaman berikut.



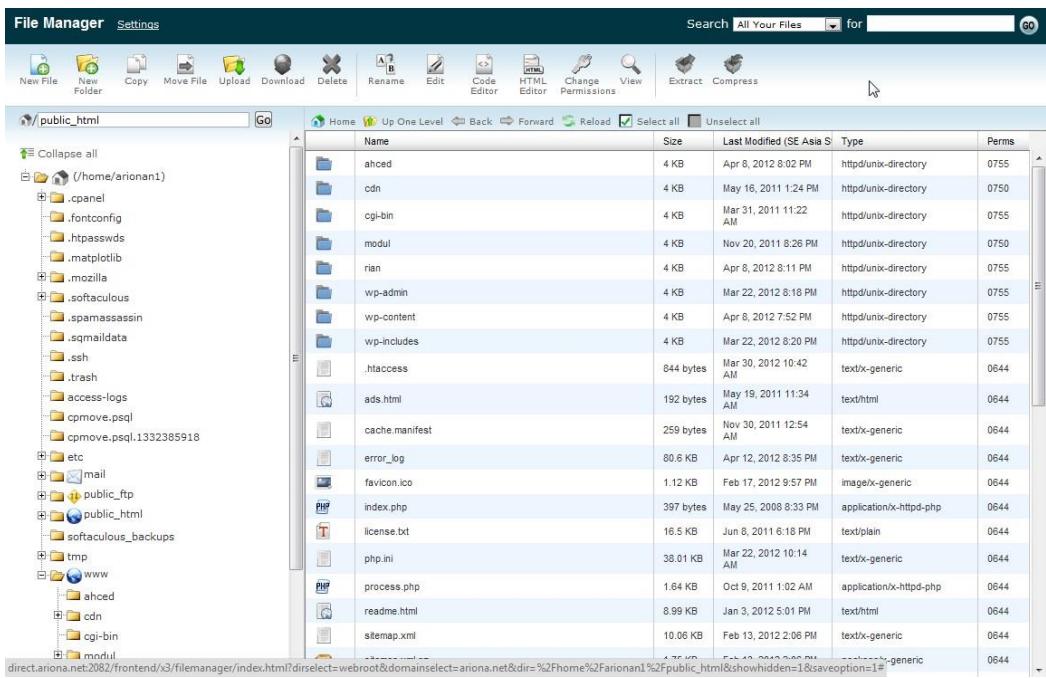
Gambar 113 Halaman login cPanel

Masukkan username dan password yang telah diberikan oleh jasa hosting yang anda gunakan untuk memasuki cpanel. Setelah anda login, berikut ini adalah tampilan halaman web yang akan anda dapatkan.



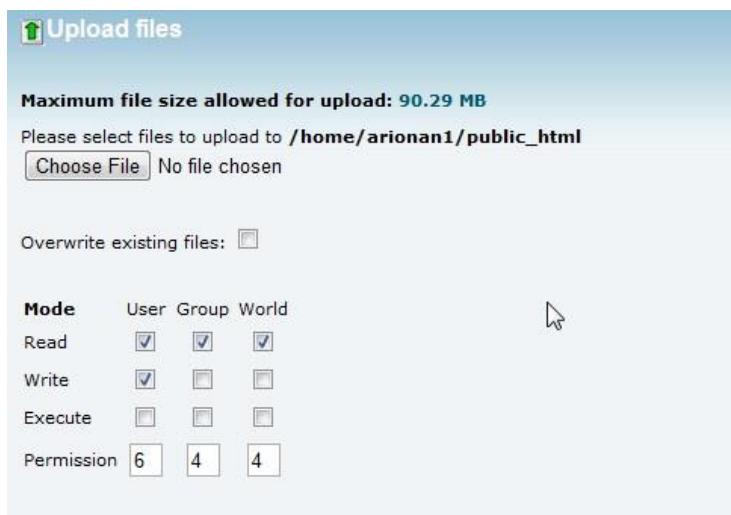
Gambar 114 Halaman cPanel

Kita mulai dengan mengupload seluruh file-file HTML dan file-file yang bersangkutan (CSS, gambar dan lain sebagainya), carilah Icon File Manager, ketika anda klik, berikut tampilan yang akan anda dapatkan, hampir sama dengan explorer atau aplikasi pencari berkas yang terdapat dalam komputer anda.



Gambar 115 File Manager untuk mengatur file di hosting

Lalu dimana kita harus meletakkan file yang kita punya agar dapat diakses oleh orang lain di internet? Anda harus mengupload file-file web anda ke folder **www** atau **public_HTML**. Klik tombol upload, dan uploadlah file anda satu persatu.



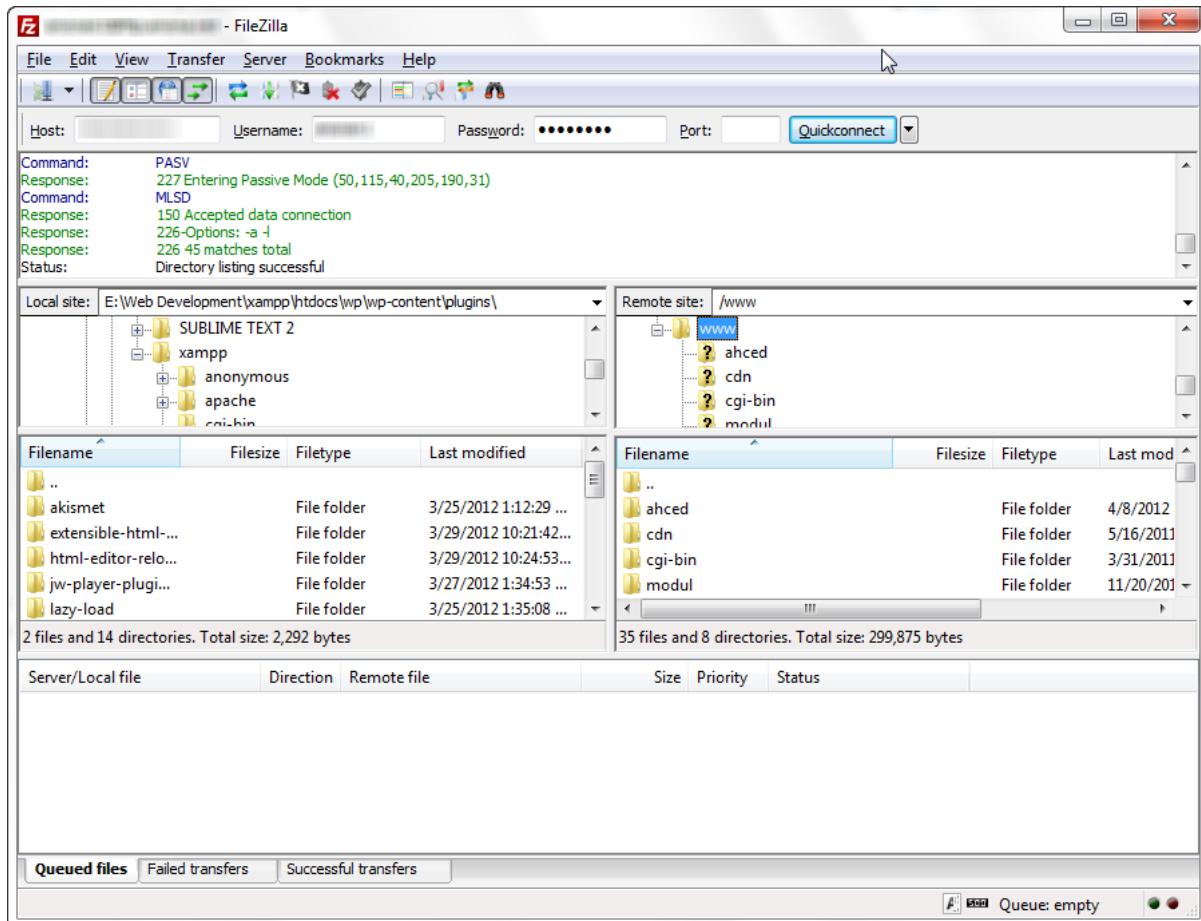
Gambar 116 Halaman untuk mengupload file

Jika semua file-sudah anda upload, andapun dapat mengaksesnya melalui alamat domain anda www.domain-anda.com.

Jika anda membuat folder lagi dalam folder www, anda dapat mengaksesnya lewat alamat : www.domain-anda.com/nama-folder.

24.5. Menggunakan FTP Client

Jika sebelumnya kita mengupload file satu persatu, sekarang kita akan belajar menggunakan FTP⁹ client untuk mengupload file-file web ke hosting kita. FTP Client adalah aplikasi kecil, untuk mengupload file-file dengan mudah ke hosting tanpa harus membuka Cpanel. Salah satu aplikasi yang paling sering digunakan adalah filezilla yang bisa anda download dari filezilla.com.



Gambar 117 FileZilla, Aplikasi FTP client free dan ringan

Masukkan, Host (domain-anda.com), username dan password cpanel anda, pada kolom login. Setelah login, maka dibagian kanan akan ditampilkan seluruh file dan folder dalam hosting anda. Dan di bagian kiri adalah file-dan folder di komputer anda.

⁹ FTP merupakan kependekan dari File Transfer Protocol, FTP digunakan untuk berinteraksi dengan file-file yang terdapat dalam server hosting dengan mudah

Cara mengupload file yang diinginkan sangatlah mudah, anda tinggal menyeleksi file-file yang dinginkan dan klik upload atau drag ke bagian kanan dan secara otomatis file-file tersebut akan diupload oleh FileZilla.

Selain mengupload, andapun dapat mendownload file yang terdapat dalam hosting anda dengan cara yang sama, yaitu memindahkannya dari kolom sebelah kanan ke kiri.

EPILOG

That's it, Sekarang anda telah mempelajari dasar-dasar dari HTML, CSS dan sedikit pengetahuan tentang Javascript. Buku ini hanyalah sebagian kecil dari pembahasan seputar Web Design, HTML, CSS dan Javascript.

Setelah selesai memahami buku ini bukan berarti perjalanan anda dalam mempelajari HTML dan CSS telah usai, masih banyak yang harus anda pelajari seperti Tag-tag baru HTML5, Efek-efek yang bisa diterapkan dengan CSS3, Penggunaan Javascript untuk interaktifitas website, Teknik Mendesain Website dengan Photoshop dan lain sebagainya.

Materi-materi yang saya sebutkan di atas bisa didapatkan dengan mudah di internet ataupun dalam buku-buku tentang Web Design/Web Development yang terdapat di toko-toko buku.

Daftar Pustaka

Buku

Way, Jeffrey Jordan. *From Photoshop to HTML & CSS*. 2009. Rockable Press.

Ta'eed, Collis. How to be Rockstar Wordpress Designer. 2009. Rockable Press.

Website

<http://www.w3schools.com>

<http://learncss.tutsplus.com>

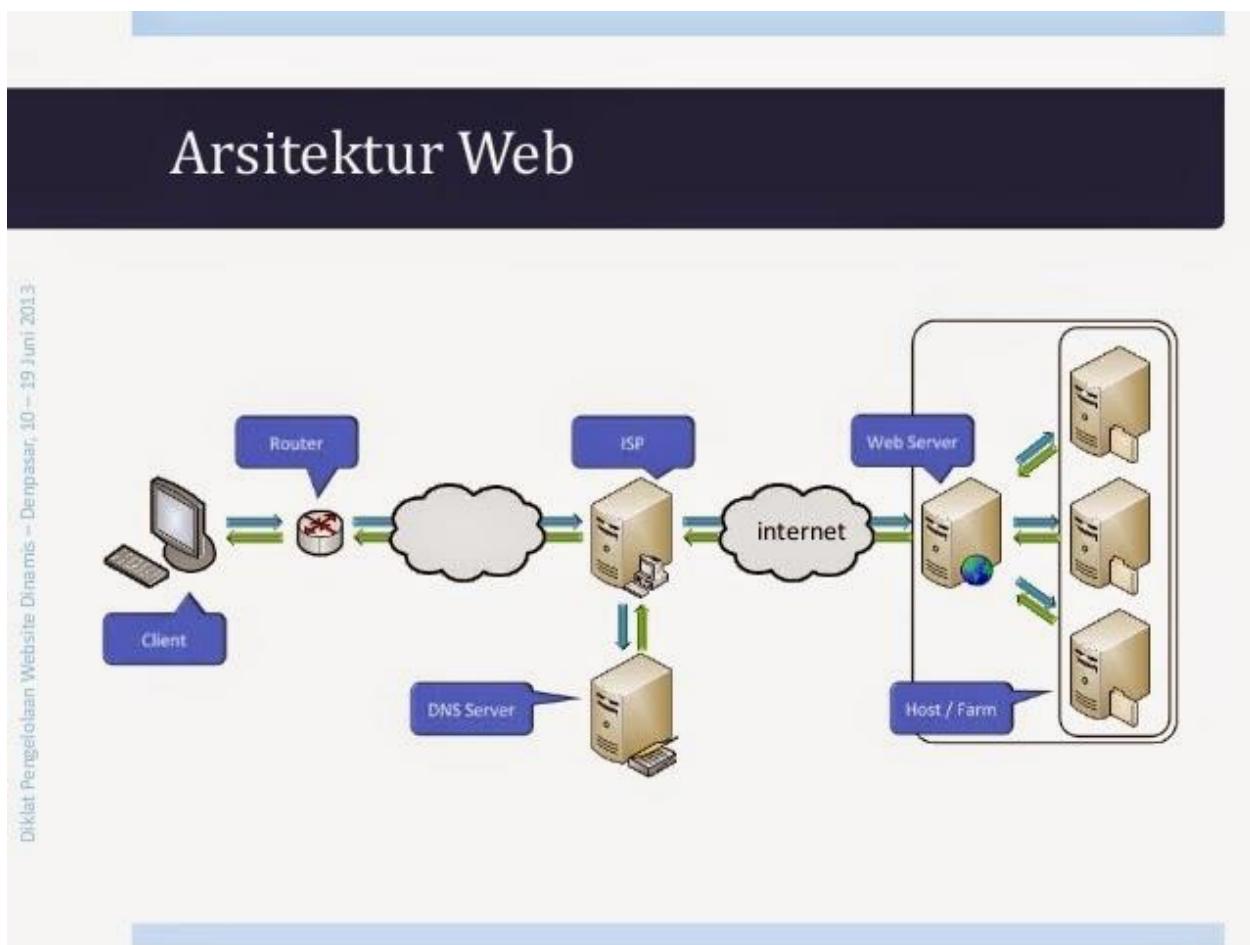
<http://www.ariona.net>

BAB I

Pengenalan Web Server dan Server Side Scripting

A. Web Server

Web Server merupakan sebuah perangkat lunak dalam server yang berfungsi menerima permintaan (request) berupa halaman web melalui HTTP atau HTTPS dari klien yang dikenal dengan browser web dan mengirimkan kembali (response) hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML.



Beberapa Web Server yang banyak digunakan di internet antara lain :

1. Apache Web Server (<http://www.apache.org>)
2. Internet Information Service, IIS (<http://www.microsoft.com/iis>)
3. Xitami Web Server (<http://www.xitami.com>)
4. Sun Java System Web Server
(http://www.sun.com/software/products/web_srvr/home_web_srvr.xml)
5. NginX

B. SKRIP DI SISI SERVER

Server Side Scripting merupakan sebuah teknologi scripting atau pemrograman web dimana script (program) dikompilasi atau diterjemahkan di server. Dengan server side scripting, memungkinkan untuk menghasilkan halaman web yang dinamis.

Beberapa contoh Server Side Scripting (Programming) :

1. ASP (Active Server Page) dan ASP.NET
2. ColdFusion (<http://www.macromedia.com/software/coldfusion>)
3. Java Server Pages (<http://java.sun.com/products/jsp/>)
4. Perl (<http://www.perl.org>)
5. Phyton (<http://www.python.org>)
6. PHP (<http://www.php.net>)

C. INSTALASI WEB SERVER

Proses instalasi Apache, PHP dan MySQL seringkali menjadi kendala terutama bagi pemula yang baru belajar pemrograman web dengan PHP. Hal tersebut disebabkan karena software tersebut harus diinstall dan dikonfigurasikan satu per satu. Bagi belum mengerti benar, tentu sering mengalami kesulitan, apalagi bagi kita yang belajar secara mandiri.

Namun jangan berkecil hati, saat ini banyak tersedia aplikasi paket yang menyatukan ketiga software tersebut (Apache, PHP dan MySQL) ke dalam satu installer. Proses instalasinya pun dapat dilakukan dengan mudah dan cepat, hanya perlu next, next, and finish.

Beberapa aplikasi paket tersebut antara lain:

1. XAMPP (versi Windows) dan LAMPP (versi Linux) yang dapat didownload di <http://apachefriends.org>.
2. WAMP Server.
3. APPServ
4. PHPTriad.
5. EasyPHP

Penulis sendiri menyarankan pemakaian XAMPP atau WAMP Server karena versi software di dalamnya menggunakan versi terbaru.

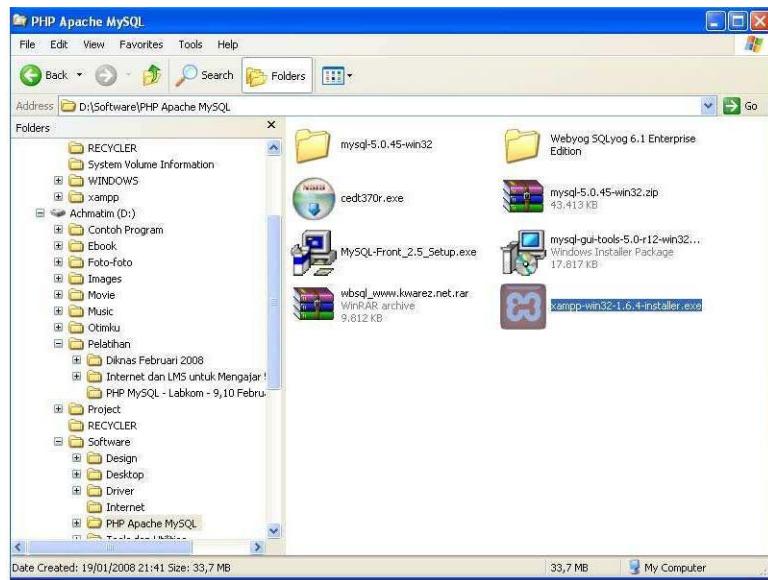
Persiapan Instalasi

1. Pastikan komputer Anda belum terinstall web server lain seperti IIS atau PWS karena dapat menyebabkan bentrok dengan web server Apache. Namun demikian, jika Anda masih tetap mempertahankan program tersebut, setelah instalasi selesai Anda dapat mengkonfigurasikan Apache secara manual dengan mengganti nomor port yang digunakan.
2. Download Source XAMPP versi stabil terbaru di <http://apachefriends.org>. Pilih versi sesuai dengan sistem operasi yang Anda gunakan (tersedia versi untuk Windows, Linux dan Mac).
3. Pastikan komputer yang Anda

gunakan berjalan dengan baik dengan kapasitas memori dan hardisk yang masih mencukupi.

Proses Instalasi XAMPP

1. Jalankan file installer XAMPP xampp-win32-x.x.x-installer.exe.



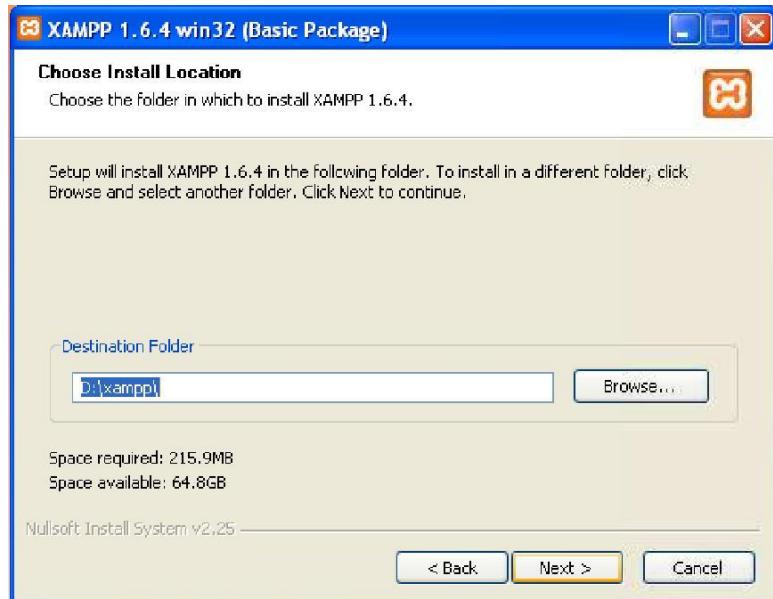
Gambar 1.1. File installer XAMPP

2. Akan ditampilkan window instalasi XAMPP. Pilih **Next >** untuk memulai proses instalasi.



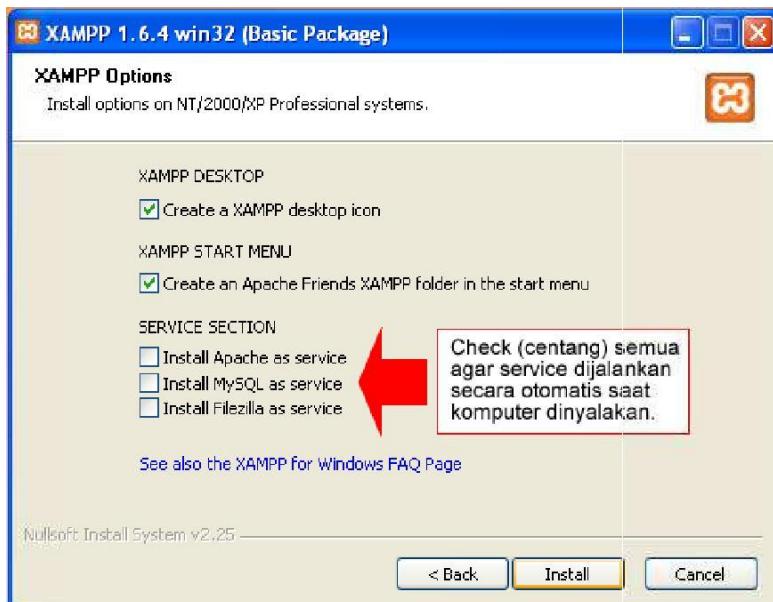
Gambar 1.2. Window Instalasi XAMPP

3. Akan ditampilkan window lokasi tujuan instalasi (*destination folder*), tentukan lokasi (folder) dimana XAMPP akan diinstall dengan menekan tombol **Browse...**. Pilih tombol **Next >** untuk melanjutkan proses instalasi.



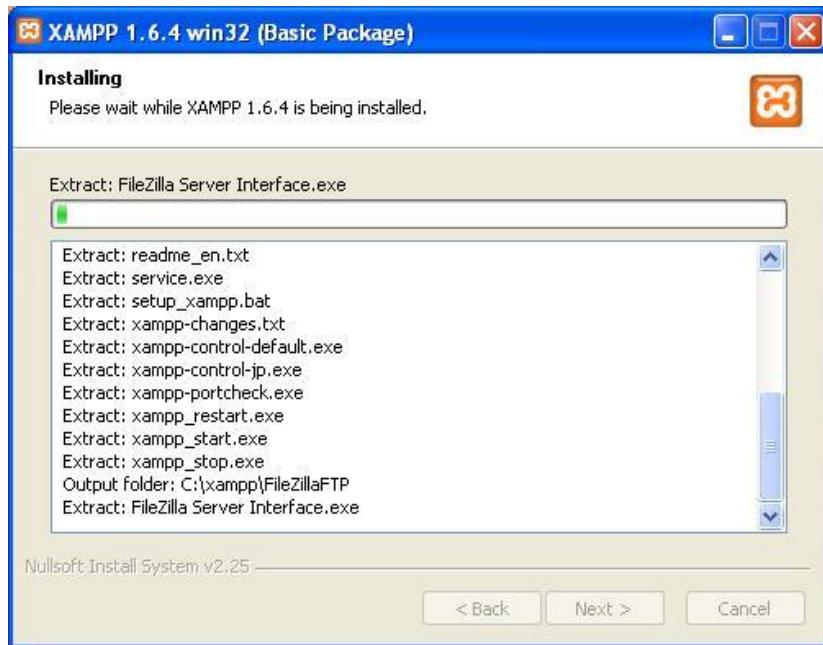
Gambar 1.3. Folder tujuan instalasi XAMPP

4. Selanjutnya akan ditampilkan window pilihan untuk kemudahan mengakses XAMPP nantinya. Tekan tombol **Install** untuk melanjutkan proses instalasi.



Gambar 1.4. Window pilihan XAMPP

-
5. Proses instalasi dimulai.



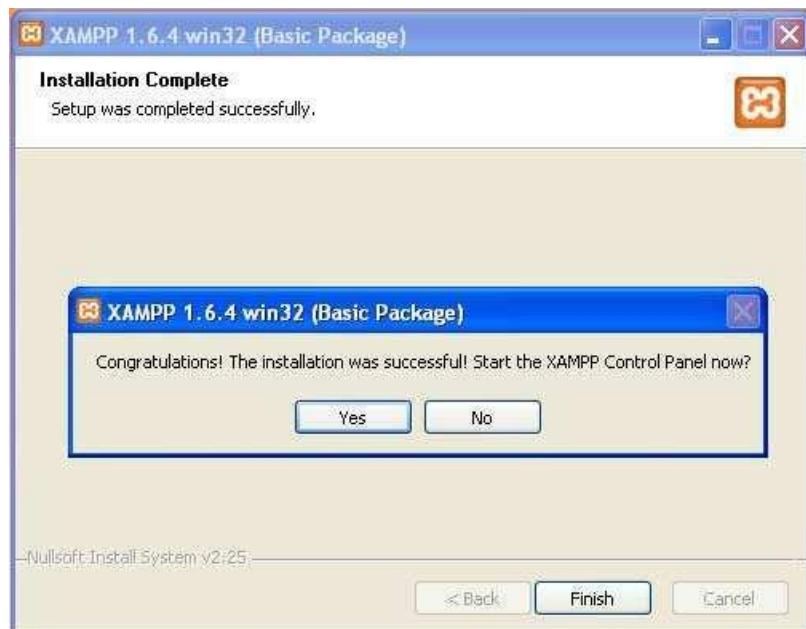
Gambar 1.5. Proses instalasi dimulai

6. Proses instalasi selesai. Klik **Finish** untuk menutup proses instalasi.



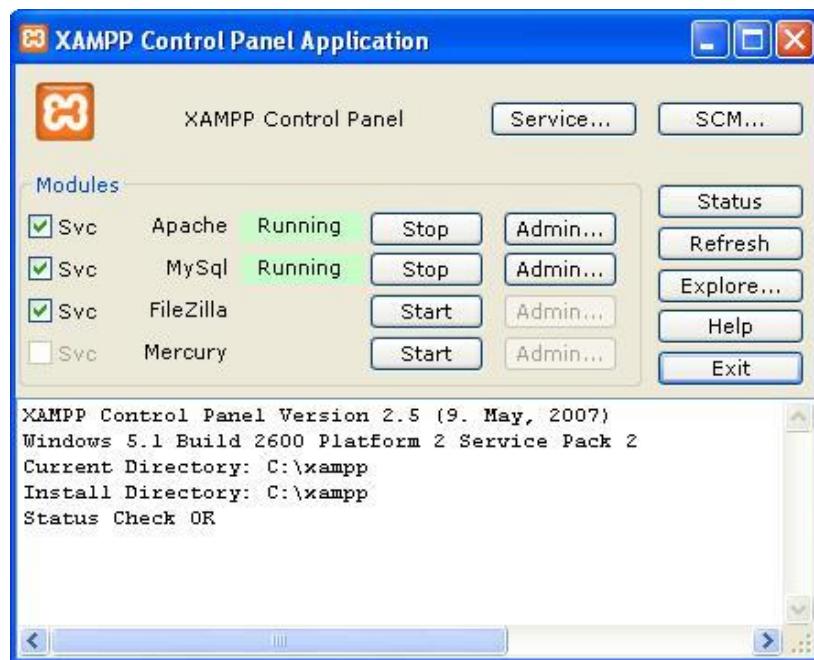
Gambar 1.6. Proses instalasi selesai

7. Jika proses instalasi berjalan dengan lancar maka akan ditampilkan window konfirmasi.



Gambar 1.7. Konfirmasi Instalasi Berhasil

8. Kita dapat mengatur XAMPP dengan mengakses XAMPP Control Panel yang dapat diakses melalui menu di Program Files atau icon di taskbar.



Gambar 1.8. XAMPP Control Panel

Test Instalasi XAMPP di Browser

Untuk memastikan bahwa XAMPP beserta PHP, Apache dan MySQL berjalan dengan baik, bukalah browser dan ketikkan URL sebagai berikut:

http://localhost

Jika semua sudah berjalan dengan baik, maka akan ditampilkan gambar sebagai berikut:



Gambar 1.9. Tampilan XAMPP di Browser

Pilih bahasa yang ingin digunakan dan akan ditampilkan halaman utama XAMPP sebagai berikut:



Gambar 1.9. Tampilan XAMPP di Browser

Bab 2

Struktur Dasar PHP

- TM Sebelum Belajar PHP
- TM Mari Mengenal PHP
- TM Say Hello to PHP
- TM Variabel
- TM Tipe Data
- TM Konstanta
- TM Operator dalam PHP
- TM Komentar Program

Sebelum Belajar PHP

1. Saya asumsikan Anda telah mengenal dengan cukup baik komputer Anda, dan juga bisa mengoperasikannya. Jadi Anda seharusnya sudah mengerti perbedaan file dan direktori, bagaimana menyimpan file atau document, bagaimana meng-copy file, dan seterusnya.
2. Saya asumsikan Anda telah mengerti bagaimana membuka dan menggunakan salah satu atau *salah semua* dari editor seperti Notepad, EditPlus, Crimson Editor, Dreamweaver, atau yang lainnya. Saya asumsikan juga, Anda telah mengerti bagaimana membuat, mengedit, dan menyimpan suatu file dengan editor tersebut.
3. Saya asumsikan bahwa Apache, PHP, dan MySQL di komputer yang Anda pakai sudah berjalan sebagaimana mestinya.
4. Saya asumsikan Anda telah mengerti bagaimana cara menggunakan Web Browser (IE, Mozilla, Opera, dsb)
5. Saya asumsikan Anda telah mengenal dan mengerti dengan cukup baik tentang tag-tag HTML dan juga bagaimana menggunakannya. Anda tentu sudah mengambil matakuliah Pemrograman Web 1 atau Internet kan ?
6. Saya asumsikan bahwa penglihatan Anda masih normal sehingga bisa membedakan antara :
 - f Tulisan NAMA, NAMA dan NAMA
 - q Koma (,), titik-koma (;) dan titik-dua (:)
 - f Kutip-tunggal ('') dan kutip-ganda (")
 - f Tulisan NAMAMHS, NAMA_MHS dan NAMA MHS
7. Saya asumsikan Anda pernah menggunakan www.google.com

Mari Mengenal PHP

Tahukah Anda, PHP itu.....

1. Merupakan singkatan recursive dari **PHP** : **Hypertext Preprocessor**
2. Pertama kali dibuat oleh **Rasmus Lerdorf** pada tahun 1994.
3. Harus ditulis di antara tag :
 - f <? dan ?>
 - q <?php dan ?>
 - q <script language="php"> dan </script>
 - q <% dan %>
4. Setiap satu statement (perintah) biasanya diakhiri dengan titik-koma (;
5. CASE SENSITIVE untuk nama identifier yang dibuat oleh user (berupa variable, konstanta, fungsi dll), namun TIDAK CASE SENSITIVE untuk identifier *built-in* dari PHP. Jadi : \$nama ≠ \$Nama ≠ \$NAMA

```
hitungLuas() ≠ HitungLuas()
echo = ECHO
while = WHILE
6. Mudah dipelajari.
```

Say Hello to PHP

1. Ketikkan perintah di bawah ini di editor :

Program 2.1

Nama File : hello.php

Deskripsi : Program sederhana menampilkan sebuah string di layar.

```
1 <?php
2 echo "Hello, Nama saya PHP ";
3 ?>
```

2. Simpan dengan nama "**hello.php**" di *document-root* komputer Anda.
3. Buka browser dan ketikkan alamat <http://localhost/hello.php>
4. Lihat hasilnya di browser. Akan terlihat tampilan seperti pada gambar 2.1



Gambar 2.1. Tampilan Program 2.1

Variabel

- f Digunakan untuk menyimpan sebuah value, data atau informasi
- f Nama variabel diawali dengan tanda \$
- f Panjang tidak terbatas
- f Setelah tanda \$ diawali oleh huruf atau *under-score* (_). Karakter berikutnya bisa terdiri dari huruf, angka, dan karakter tertentu yang diperbolehkan (karakter ASCII dari 127 – 255).
- f Bersifat case-sensitive.
- f Tidak perlu dideklarasikan.
- f Tidak boleh mengandung spasi.

f Contoh :

Benar

- \$_name
- \$first_name
- \$name3
- \$name_3

Salah

- \$3name
- \$name?
- \$first+name
- \$first.name
- \$first name

Program 2.2

Nama File : variabel.php

Deskripsi : Program mengisi dan menampilkan variabel di layar.

```

1 <?php
2 $nim = "0411500400";
3 $nama = 'Chotimatul Musyarofah';
4
5 echo "NIM : " . $nim . "<br>"; 6 echo "Nama
: $nama";
7 ?>

```



Gambar 2.2. Tampilan Program 2.2

Tipe Data

Pada PHP, tipe data variabel tidak didefinisikan oleh programmer, akan tetapi secara otomatis ditentukan oleh intepreter PHP. Namun demikian, PHP mendukung 8 (delapan) buah tipe data primitif, yaitu :

1. boolean
2. integer
3. float
4. string
5. array
6. object
7. resource
8. NULL

Program 2.3

Nama File : *tipe.php*
 Deskripsi : *Program contoh penanganan beberapa tipe data dasar dalam PHP.*

```

1 <?php
2 $nim = "0411500400";
3 $nama = 'Chotimatul Musyarofah';
4 $umur = 23;
5 $nilai = 82.25;
6 $status = TRUE;
7
8 echo "NIM : " . $nim . "<br>";
9 echo "Nama : $nama<br>";
10 print "Umur : " . $umur; print "<br>";
11 printf ("Nilai : %.3f<br>", $nilai);
12 if ($status)
13     echo "Status : Aktif";
14 else
15     echo "Status : Tidak Aktif";
16 ?>

```



Gambar 2.3. Tampilan Program 2.3

Konstanta

Konstanta merupakan variabel konstan yang nilainya tidak berubah-ubah. Untuk mendefinisikan konstanta dalam PHP, menggunakan fungsi define()

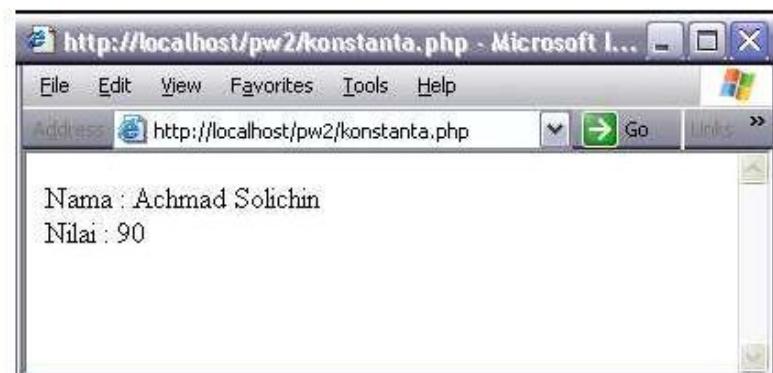
Program 2.4

Nama File : konstanta.php

Deskripsi : Program penggunaan konstanta dalam PHP.

```

1 <?
2 define ("NAMA", "Achmad Solichin");
3 define ("NILAI", 90);
4
5 //NAMA = "Muhammad"; //akan menyebabkan error
6 echo "Nama : " . NAMA;
7 echo "<br>Nilai : " . NILAI;
8 ?>
```



Gambar 2.4. Tampilan Program 2.4

Operator dalam PHP

Jenis Operator	Operator	Contoh	Keterangan
Aritmatika	+	\$a + \$b	Pertambahan
	-	\$a - \$b	Pengurangan
	*	\$a * \$b	Perkalian
	/	\$a / \$b	Pembagian
	%	\$a % \$b	Modulus, sisa pembagian

Penugasan	=	\$a = 4;	\$a diisi dengan 4
Bitwise	&	\$a & \$b	Bitwise AND
		\$a \$b	Bitwise OR
	^	\$a ^ \$b	Bitwise XOR
	~	~\$b	Bitwise NOT
	<<	\$a << \$b	Shift Left
	>>	\$a >> \$b	Shift Right
Perbandingan	==	\$a == \$b	Sama dengan
	==>	\$a === \$b	Identik
	!=	\$a != \$b	Tidak sama dengan
	<>	\$a <> \$b	Tidak sama dengan
	!==	\$a !== \$b	Tidak identik
	<	\$a < \$b	Kurang dari
	>	\$a > \$b	Lebih dari
	<=	\$a <= \$b	Kurang dari sama dengan
	>=	\$a >= \$b	Lebih dari sama dengan
Logika	and	\$a and \$b	TRUE jika \$a dan \$b TRUE
	&&	\$a && \$b	TRUE jika \$a dan \$b TRUE
	or	\$a or \$b	TRUE jika \$a atau \$b TRUE
		\$a \$b	TRUE jika \$a dan/atau \$b TRUE
	xor	\$a xor \$b	TRUE jika \$a atau \$b TRUE, tapi tidak keduanya
	!	!\$a	TRUE jika \$a FALSE
String	.	\$a . \$b	Penggabungan string \$a dan \$b

Program 2.5

Nama File : operator.php

Deskripsi : Program beberapa Operator aritmatika dalam PHP.

```

1 <?
2 $gaji = 1000000;
3 $pajak = 0.1;
4 $thp = $gaji - ($gaji*$pajak); 5
5 echo "Gaji sebelum pajak = Rp. $gaji <br>";
7 echo "Gaji yang dibawa pulang = Rp. $thp";
8 ?>

```

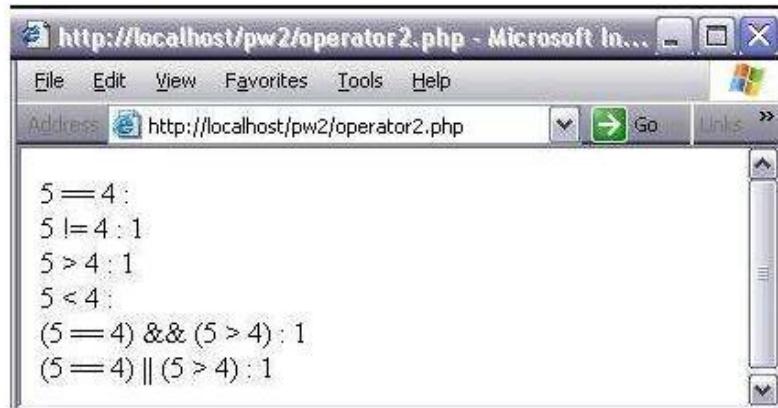


Gambar 2.5. Tampilan Program 2.5

Program 2.6

*Nama File : operator2.php
Deskripsi : Program operator logika dan perbandingan dalam PHP.*

```
1 <?
2 $a = 5;
3 $b = 4;
4
5 echo "$a == $b : ". ($a == $b);
6 echo "<br>$a != $b : ". ($a != $b); 7 echo "<br>$a
> $b : ". ($a > $b);
8 echo "<br>$a < $b : ". ($a < $b);
9 echo "<br>($a == $b) && ($a > $b) : ". (($a != $b) && ($a > $b));
10 echo "<br>($a == $b) || ($a > $b) : ". (($a != $b) || ($a > $b));
11 ?>
```



Gambar 2.6. Tampilan Program 2.6

Komentar Program

Dalam PHP, komentar program bisa menggunakan :

```
f /* dan */
f // dan
f #
```

Program 2.7

*Nama File : komentar.php
Deskripsi : Program cara memberikan komentar dalam program PHP.*

```
1 <?
2 /* Ini komentar tidak akan tercetak di layar
3     yang ini juga komentar
4 */
5
6 // Nah kalo ini komentar satu baris
7
8 # Terus kalo ini juga komentar
9 echo "Ini akan tercetak di layar";
10 ?>
```



Gambar 2.7. Tampilan Program 2.7

Bab 3

Struktur Kondisi dan Perulangan

- ™ Struktur Kondisi
 - + Struktur Kondisi If
 - + Struktur Kondisi If...Else
 - + Struktur Kondisi Khusus ? :
 - + Struktur Kondisi Switch...Case
- ™ Struktur Perulangan
 - + Struktur Kondisi For
 - + Struktur Kondisi While
 - + Struktur Kondisi Do...While
 - + Struktur Kondisi Foreach
- ™ Struktur Break dan Continue

Struktur Kondisi

Struktur Kondisi If

```
if (kondisi) { statement-jika-kondisi-true;  
}
```

Keterangan :

f **kondisi** merupakan statemen atau variabel yang akan diperiksa TRUE atau FALSE-nya.

Struktur Kondisi If ... Else

```
if (kondisi) { statement-jika-kondisi-true;  
} else { statement-jika-kondisi-false;  
}
```

Keterangan :

29 **kondisi** merupakan statemen atau variabel yang akan diperiksa TRUE atau FALSE-nya.
Jika kondisinya TRUE maka statemen yang berada di blok if akan dieksekusi, sebaliknya jika kondisinya FALSE maka statemen yang berada di blok else yang akan dieksekusi.

Struktur Kondisi Khusus ? :

```
(kondisi) ? benar : salah;
```

Keterangan :

f **kondisi** merupakan statemen atau variabel yang akan diperiksa TRUE atau FALSE-nya.
Statement pada blok benar dan salah hanya boleh satu statemen saja

Struktur Kondisi Switch ... Case

```
switch ($var) {  
    case '1' : statement-1; break; case '2' :  
        statement-2; break;  
    ....  
}
```

Keterangan :

- f \$var merupakan variabel yang akan diperiksa isi atau nilainya. Tipe data variabel ini tidak dibatasi.
- f Value pada case juga bisa berupa string, integer, boolean, bahkan bisa berupa conditional-statement. Boleh memakai kutip tunggal maupun kutip ganda.

Program 3.1

Nama File : if.php

Deskripsi : Program Struktur Kondisi If.

```
1 <?php
2 $nilai = 80;
3 if ($nilai >= 60) {
4     echo "Nilai Anda $nilai, Anda LULUS";
5 }
6 ?>
```

Penjelasan Program 3.1

Pada program di atas, terdapat pemeriksaan kondisi apakah isi variabel \$nilai lebih dari 60 (baris 3). Jika kondisi ini bernilai **TRUE** (variabel \$nilai berisi nilai yang lebih besar dari 60) maka statement pada baris ke-4 akan dijalankan. Sebaliknya jika kondisinya **FALSE**, maka statement pada baris ke-4 tidak akan dijalankan.



Gambar 3.1. Tampilan Program 3.1

Program 3.2

Nama File : if_else.php

Deskripsi : Program Struktur Kondisi If..Else.

```
1 <?php
2 $nilai = 50;
3 if ($nilai >= 60) {
4     echo "Nilai Anda $nilai, Anda LULUS";
5 } else {
6     echo "Nilai Anda $nilai, Anda GAGAL";
7 }
8 ?>
```

Penjelasan Program 3.2

Pada program di atas, terdapat pemeriksaan kondisi apakah isi variabel \$nilai lebih dari 60 (baris 3). Jika kondisi ini bernilai **TRUE** (variabel \$nilai berisi nilai yang lebih besar dari 60) maka statement pada baris ke-4 akan dijalankan. Sebaliknya jika kondisinya **FALSE**, maka statement pada baris ke-6 yang akan dijalankan. Pada contoh program di atas, kondisi pada baris ke-3 bernilai **FALSE** karena isi variabel \$nilai adalah 50.



Gambar 3.2. Tampilan Program 3.2

Program 3.3

Nama File : if_else2.php

Deskripsi : Program Memeriksa username dan password dengan If..Else.

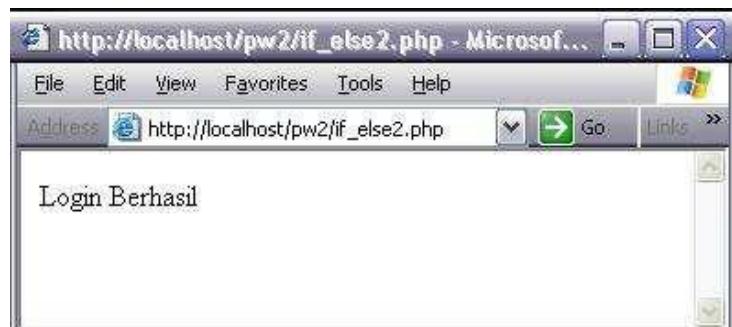
```

1 <?php
2 $user = "achmatim";
3 $pass = "123"
4 if ($user == "achmatim" && $pass == "123") {
5     echo "Login Berhasil";
6 } else {
7     echo "Login Gagal";
8 }
9 ?>

```

Penjelasan Program 3.3

Pada program di atas, terdapat pemeriksaan kondisi apakah isi variabel **\$nilai** sama dengan "achmatim" **dan** apakah isi variabel **\$pass** sama dengan "123" (baris 4). Jika kedua kondisi tersebut bernilai **TRUE** maka statement pada baris ke-5 akan dijalankan. Sebaliknya jika salah satunya bernilai **FALSE**, maka statement pada baris ke-7 yang akan dijalankan.



Gambar 3.3. Tampilan Program 3.3

Program 3.4

Nama File : if_var.php

Deskripsi : Program Memeriksa suatu variabel ada atau tidak.

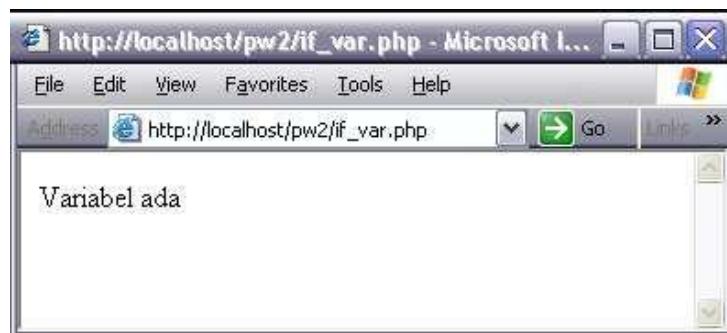
```

1 <?php
2 $user="";
3 if (!isset($user)) {
4     echo "Variabel tidak ada/belum terbentuk";
5 } else {
6     echo "Variabel ada";
7 }
?>

```

Penjelasan Program 3.4

Pada program di atas, terdapat fungsi **isset()** yang merupakan fungsi untuk memeriksa apakah suatu variabel sudah terbentuk (ada) atau belum. Fungsi akan bernilai **TRUE** jika variabelnya ada. Jadi jika baris ke-2 dari program di atas di- *remark* atau dihilangkan maka *statement* pada baris ke-4 yang akan dijalankan.



Gambar 3.4. Tampilan Program 3.4

Program 3.5

Nama File : switch.php
Deskripsi : Program Struktur Switch..Case untuk menampilkan nama hari dalam bahasa Indonesia.

```
1 <?php
2 $day = date ("D");
3 switch ($day) {
4     case 'Sun' : $hari = "Minggu"; break;
5     case 'Mon' : $hari = "Senin"; break;
6     case 'Tue' : $hari = "Selasa"; break;
7     case 'Wed' : $hari = "Rabu"; break;
8     case 'Thu' : $hari = "Kamis"; break;
9     case 'Fri' : $hari = "Jum'at"; break;
10    case 'Sat' : $hari = "Sabtu"; break;
11    default : $hari = "Kiamat";
12 }
13 echo "Hari ini hari <b>$hari</b>";
14 ?>
```

Penjelasan Program 3.5

Pada program di atas, variabel **\$day** (baris ke-2) akan berisi 3 (tiga) digit pertama dari nama hari dalam bahasa Inggris. Dengan fungsi **date()**, kita akan memperoleh informasi tanggal, hari dan waktu sekarang. Fungsi ini akan diperlajari lebih lanjut pada Bab mendatang. Selanjutnya isi variabel **\$day** akan diperiksa dengan switch (baris ke-3), jika isinya adalah 'Sun' maka **\$hari** adalah

"Minggu" dan seterusnya.



Gambar 3.5. Tampilan Program 3.5

Program 3.6

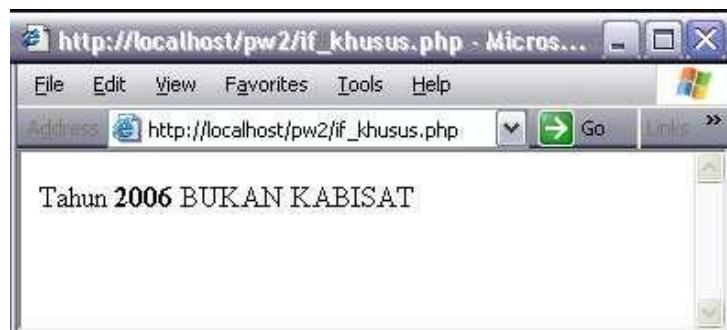
Nama File : if_khusus.php
Deskripsi : Program Struktur Kondisi Khusus ? untuk memeriksa tahun kabisat.

```
1 <?php
2 $tahun = date ("Y");
3 $kabisat = ($tahun%4 == 0) ? "KABISAT" : "BUKAN KABISAT"; 4 echo "Tahun
<b>$tahun</b> $kabisat";
5 ?>
```

Penjelasan Program 3.6

Pada program di atas, fungsi **date()** dengan parameter "Y" (baris ke-2) akan menghasilkan 4 digit tahun sekarang. Pada baris ke-3 akan diperiksa apakah isi variabel \$tahun jika dimodulus dengan 4 akan menghasilkan nilai 0. Jika **TRUE** maka \$kabisat akan berisi "KABISAT" dan jika **FALSE** \$kabisat akan berisi

"BUKAN KABISAT"



Gambar 3.6. Tampilan Program 3.6

Struktur Perulangan

Struktur Perulangan For

```
for (init_awal, kondisi, counter) { statement-
    yang-diulang;
}
```

Keterangan :

- ƒ **init_awal** merupakan inisialisasi atau nilai awal variable.
- ƒ **kondisi** merupakan statemen kondisi yang akan membatasi perulangan.
- ƒ **counter** merupakan pertambahan atau pengurangan nilai variabel sehingga perulangan tetap berjalan.

Struktur Perulangan While

```
init_awal;
while (kondisi) { statement-yang-
    diulang; counter;
}
```

Keterangan :

- ƒ **init_awal** merupakan inisialisasi atau nilai awal variable.
- ƒ **kondisi** merupakan statemen kondisi yang akan membatasi perulangan.
- ƒ **counter** merupakan pertambahan atau pengurangan nilai variabel sehingga perulangan tetap berjalan.

Struktur Perulangan Do ... while

```
init_awal; do
{
    statement-yang-diulang;
    counter;
} while (kondisi);
```

Keterangan :

- f **init_awal** merupakan inisialisasi atau nilai awal variable.
- f **kondisi** merupakan statemen kondisi yang akan membatasi perulangan.
- f **counter** merupakan pertambahan atau pengurangan nilai variabel sehingga perulangan tetap berjalan.
- f Pada struktur do...while, pemeriksaan kondisi ada di bawah, sehingga statement yang berada dalam block do...while setidaknya akan dieksekusi sebanyak satu kali.

Struktur Perulangan Foreach

```
foreach (array_expression as $value) statement;

foreach (array_expression as $key => $value) statement;
```

Keterangan :

- f Struktur foreach biasanya digunakan untuk melakukan perulangan berdasarkan isi suatu array. Perulangan akan berakhir jika isi array telah habis.

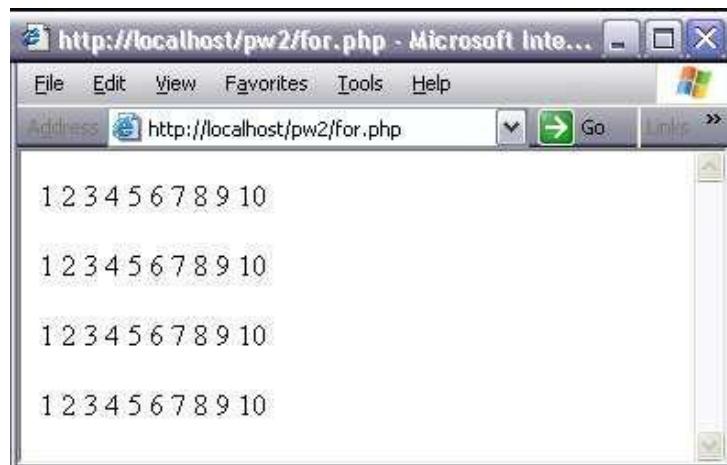
Program 3.7

Nama File : for.php
Deskripsi : Program Struktur Perulangan For dan beberapa variasinya.

```
1 <?php
2
3 /* contoh 1 */
4 for ($i = 1; $i <= 10; $i++) { 5 echo "$i ";
5
6 }
7 echo "<br><br>";
8
9 /* contoh 2 */
10 for ($i = 1; ; $i++) {
11     if ($i > 10) {
12         break;
13     }
14     echo "$i ";
15 }
16 echo "<br><br>";
17
18 /* contoh 3 */
19 $i = 1;
20 for (; ; ) {
21     if ($i > 10) {
22         break;
23     }
24     echo "$i ";
25     $i++;
26 } echo "<br><br>";
27
28 /* contoh 4 */
29 for ($i = 1; $i <= 10; print "$i ", $i++); 30 31 ?>
```

Penjelasan Program 3.7

Program di atas merupakan bentuk-bentuk perulangan dengan menggunakan for. Contoh yang pertama (baris 4-6) merupakan bentuk yang paling umum. Pada contoh 2, batas akhir perulangan tidak disebutkan dalam for, tapi diatur dengan if dan **break** (baris 11-13).



Gambar 3.7. Tampilan Program 3.7

Program 3.8

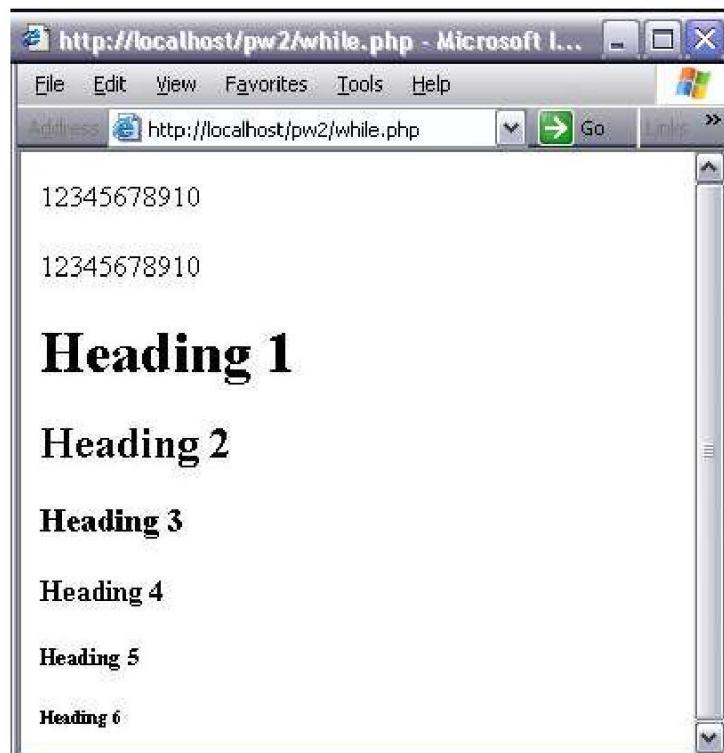
Nama File : while.php

Deskripsi : Program Struktur Perulangan While dan beberapa variasinya.

```
1 <?php
2 /* contoh 1 */
3 $i = 1;
4 while ($i <= 10) {
5     echo $i++;
6 }
7 echo "<br><br>";
8
9 /* contoh 2 */
10 $i = 1;
11 while ($i <= 10):
12     echo "$i";
13     $i++;
14 endwhile;
15 echo "<br><br>";
16
17 /* contoh 3 */
18 $i = 1;
19 while ($i <= 6) {
20     echo "<h{$i}>Heading $i</h{$i}>";
21     $i++;
22 }
23 ?>
```

Penjelasan Program 3.8

Program di atas merupakan beberapa variasi perulangan dengan while. Contoh yang pertama (baris 3-6) merupakan bentuk yang paling umum dipakai. Blok perintah (baris 5) akan dijalankan selama pemeriksaan kondisi pada baris ke-4 bernilai **TRUE**. Contoh kedua merupakan bentuk lain dari while, bentuk ini jarang digunakan. Pada contoh ketiga, penggunaan perulangan untuk menampilkan tulisan dengan format <H1> sampai <H6>



Gambar 3.8. Tampilan Program 3.8

Program 3.9

Nama File

: dowhile.php

Deskripsi

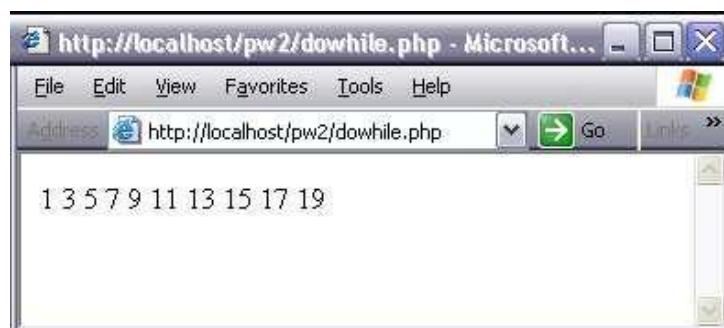
: Program Struktur Perulangan dengan Do...While.

```

1 <?php
2
3 $i = 1;
4 do {
5     echo "$i ";
6     $i+=2;
7 } while ($i <= 20);
8
9 ?>
```

Penjelasan Program 3.9

Program di atas akan menampilkan bilangan ganjil antara 1 sampai 20 menggunakan struktur perulangan **do..while**. Pada struktur perulangan do...while, pemeriksaan kondisi berada di bawah.



Gambar 3.9. Tampilan Program 3.9

Struktur Break dan Continue

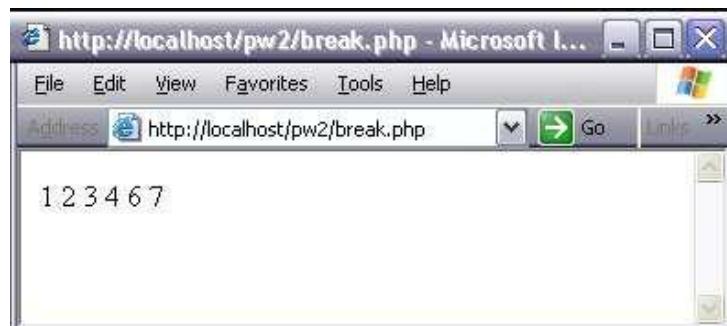
Program 3.10

Nama File : break.php
Deskripsi : Program Struktur Break dan Continue.

```
1 <?php
2
3 for ($i=1; $i<10; $i++) {
4     if ($i == 5)
5         continue;
6     if ($i == 8)
7         break;
8     echo "$i ";
9 }
10 ?>
```

Penjelasan Program 3.10

Dari program di atas, dapat disimpulkan bahwa perintah **continue** akan melanjutkan proses perulangan tanpa melewati (meng-eksekusi) baris perintah di bawahnya. Jadi perintah pada baris 6-8 akan dilewati (5 tidak akan tercetak) Sedangkan dengan perintah **break** akan menyebabkan program menghentikan perulangan (langsung keluar dari perulangan)



Gambar 3.10. Tampilan Program 3.10

Bab 4

Penanganan Form

- TM Berbagai Cara Penanganan Form
- TM Form Input Type TEXT dan PASSWORD
- TM Form Input Type RADIO
- TM Form Input Type CHECKBOX
- TM Form Input Type COMBO BOX
- TM Form Input Type TEXTAREA

Form inputan dibuat dengan tag-tag HTML. Halaman yang mengandung form murni (tidak ada *script php*) tidak harus disimpan dalam bentuk php, bisa dalam bentuk html.

Untuk merancang sebuah form inputan, setidaknya ada 3 (tiga) hal penting, yi :

1. METHOD

Method dari sebuah form menentukan bagaimana data inputan form dikirim. Method ini ada dua macam, yaitu GET dan POST. Method ini menentukan bagaimana data inputan dikirim dan diproses oleh PHP.

2. ACTION

Action dari sebuah form menentukan dimana data inputan dari form diproses. Jika action ini dikosongkan, maka dianggap proses form terjadi di halaman yang sama. Jadi halaman form dan halaman proses bisa saja dipisah atau dijadikan satu.

3. SUBMIT BUTTON

Submit button merupakan sebuah tombol (pada umumnya) yang berfungsi sebagai *trigger* pengiriman data dari form inputan. Jika tombol ini ditekan, maka data form akan dikirimkan (diproses) di halaman yang sudah ditentukan pada atribut action.

Berbagai Cara Penanganan Form

Cara 1 : Menyatukan antara Form dan Proses

Proses pengolahan form dilakukan di halaman yang sama dengan form inputannya. Jika proses penanganan form berada di halaman, maka *value* atribut **action** pada tag **form** tidak perlu (dikosongkan).

Program 4.1

Nama File : *input01.php*
Deskripsi : Program Contoh pengolahan form dimana antara form inputan dan proses pengolahan inputan berada dalam satu halaman.

```
1 <html>
2   <head><title>Pengolahan Form</title></head>
3   <body>
4     <FORM ACTION="" METHOD="POST" NAME="input">
5       Nama Anda : <input type="text" name="nama"><br>
6       <input type="submit" name="Input" value="Input">
7     </FORM>
8   </body>
9 </html>
10
11 <?php
12 if (isset($_POST['Input'])) {
13   $nama = $_POST['nama'];
14   echo "Nama Anda : <b>$nama</b>";
15 }
16 ?>
```

Penjelasan Program 4.1

Beberapa hal yang perlu diperhatikan dari program 4.1 di atas, di antaranya mengenai nama setiap komponen form karena nama ini akan menjadi index array dalam PHP. Pada program 4.1 di atas, *value* atribut **action** pada tag **form** tidak diisi (baris 4), ini berarti bahwa proses pengolahan form berada di halaman yang sama. Selanjutnya (baris 4) method yang digunakan dalam penanganan form adalah POST. Cara ini lebih disarankan dalam penanganan sebuah form inputan.

Mulai baris ke-11 hingga 16, terdapat script / program PHP yang akan menangani (mengolah) nilai yang diinputkan melalui form. Letak proses inputan ini boleh sebelum atau sesudah form, tergantung kebutuhan. Pada baris ke-12, terdapat pemeriksaan kondisi apakah tombol dengan nama "**Input**" (perhatikan index array **\$_POST** dan bandingkan dengan name tombol **submit**

pada form) benar-benar ditekan atau tidak oleh user. Selanjutnya pada baris ke-13, nilai inputan dari form akan diambil dengan cara mengakses array **\$_POST** sesuai dengan komponen form yang akan diambil. Pada baris 13 ini, isi komponen inputan dengan nama "**nama**" akan diambil dan dimasukkan ke variabel **\$nama**. Index array **\$_POST** pada baris 13 harus sama dengan *value* atribut **name** pada baris ke-5.



Gambar 4.1. Tampilan Program 4.1

Cara 2 : Memisahkan antara Form dan Proses

Proses pengolahan form dilakukan di halaman yang terpisah dengan form inputannya. Jika proses penanganan form berada dilakukan di halaman yang berbeda, maka *value* atribut **action** pada tag **form** harus diisi dengan alamat halaman tempat proses pengolahan form.

Program 4.2

Nama File : *input02.php*

Deskripsi : Program menampilkan form inputan dengan method POST

```
1 <html>
2   <head><title>Pengolahan Form</title></head>
3   <body>
4     <FORM ACTION="proses02.php" METHOD="POST" NAME="input">
5       Nama Anda : <input type="text" name="nama"><br>
6       <input type="submit" name="Input" value="Input">
7     </FORM>
8   </body>
9 </html>
```

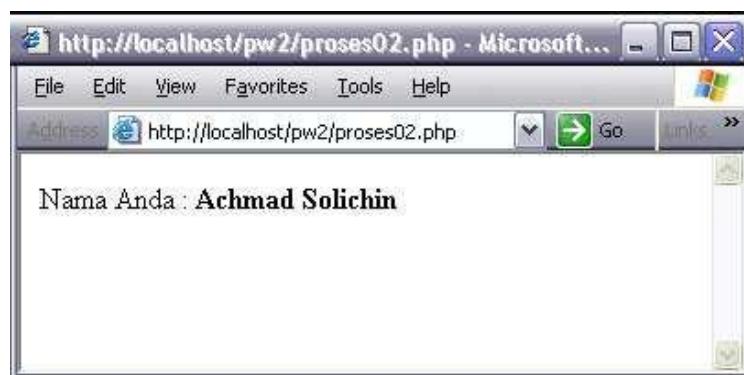
Program 4.3

Nama File : proses02.php
Deskripsi : Program penanganan form inputan untuk Program 4.2.

```
1 <?php
2 if (isset($_POST['Input'])) {
3     $nama = $_POST['nama'];
4     echo "Nama Anda : <b>$nama</b>";
5 }
6 ?>
```



Gambar 4.2. Tampilan Program 4.2



Gambar 4.3. Tampilan Program 4.3

Program 4.4

Nama File : input03.php
Deskripsi : Program menampilkan form inputan dengan method GET

```
1 <html>
2     <head><title>Pengolahan Form</title></head>
3     <body>
4         <FORM ACTION="proses03.php" METHOD="GET" NAME="input">
5             Nama Anda : <input type="text" name="nama"><br>
6             <input type="submit" name="Input" value="Input">
7         </FORM>
8     </body>
9 </html>
```

Program 4.5

Nama File : proses03.php
Deskripsi : Program penanganan form inputan untuk Program 4.4.

```
1 <?php
2 if (isset($_GET['Input'])) {
3     $nama = $_GET['nama'];
4     echo "Nama Anda : <b>$nama</b>";
5 }
6 ?>
```



Gambar 4.4. Tampilan Program 4.4



Gambar 4.5. Tampilan Program 4.5

Penjelasan Program 4.2, 4.3, 4.4 dan 4.5

Program 4.2 dan program 4.3 pada prinsipnya sama dengan program 4.1. Hasilnya pun sama. Akan tetapi, proses penanganan form (PHP) terpisah dengan tampilan form inputannya (HTML). Pada program 4.2 dan 4.3, penanganan form menggunakan method POST, sedang pada program 4.4 dan 4.5, penanganan form dilakukan dengan method GET. Perhatikan, method ini menentukan bagaimana sebuah form diproses dan bagaimana variabel inputan diambil. Bandingkan baris ke-2 dan ke-3 pada program 4.3 dan program 4.5 !

Form Input Type TEXT dan PASSWORD

Program 4.6

Nama File : input04.php
Deskripsi : Program menampilkan form inputan text dalam jumlah banyak.

```
1 <html>
2     <head><title>Pengolahan Form ~ Text</title></head>
```

```

3 <body>
4   <FORM ACTION="proses04.php" METHOD="POST" NAME="input">
5     Sahabat-sahabat Dekatku<br>
6     <input type="text" name="nama1"><br>
7     <input type="text" name="nama2"><br>
8     <input type="text" name="nama3"><br>
9     <input type="text" name="nama4"><br>
10    <input type="submit" name="Input" value="Input">
11  </FORM>
12 </body>
13 </html>

```

Program 4.7

Nama File : proses04.php

Deskripsi : Program penanganan form inputan untuk Program 4.6.

```

1 <?php
2 if (isset($_POST['Input'])) {
3   $nama1 = $_POST['nama1'];
4   $nama2 = $_POST['nama2'];
5   $nama3 = $_POST['nama3'];
6   $nama4 = $_POST['nama4'];
7   echo "<b>Nama Sahabat-sahabat Dekatku :</b> <br>";
8   echo $nama1. "<br>";
9   echo $nama2. "<br>";
10  echo $nama3. "<br>";
11  echo $nama4. "<br>";
12 }
13 ?>

```

Penjelasan Program 4.6 dan 4.7

Baris 3-6 program 4.7 merupakan proses pengambilan nilai dari masing-masing form inputan text yang terdapat pada program 4.6. Selanjutnya variabel ini ditampilkan di layar (baris 8-11, program 4.7).



Gambar 4.6. Tampilan Program 4.6



Gambar 4.7. Tampilan Program 4.7

Program 4.8

Nama File : *input05.php*

Deskripsi : Program menampilkan form login (inputan text dan password).

```

1 <html>
2   <head><title>Login Here</title></head>
3   <body>
4     <FORM ACTION="proses05.php" METHOD="POST" NAME="input">
5       <h2>Login Here...</h2>
6       Username : <input type="text" name="username"><br>
7       Password : <input type="password" name="password"><br>
8       <input type="submit" name="Login" value="Login">
9       <input type="reset" name="reset" value="Reset">
10      </FORM>
11    </body>
12  </html>

```

Program 4.9

Nama File : *proses05.php*

Deskripsi : Program sederhana untuk memeriksa inputan username dan password pada program 4.8

```

1 <?php
2 if (isset($_POST['Login'])) {
3   $user = $_POST['username'];
4   $pass = $_POST['password'];
5 if ($user == "achmatim" && $pass == "123") { 6 echo "<h2>Login
Berhasil</h2>";
7   } else {
8     echo "<h2>Login Gagal</h2>";
9   }
10  }
11 ?>

```

Penjelasan Program 4.8 dan 4.9

Program 4.8 akan menampilkan form login sederhana yang terdiri dari inputan **username** dan **password**. Selanjutnya nilai yang diinput akan diproses di program 4.9. Jika username dan password diinput dengan benar maka akan ditampilkan pesan berhasil (lihat gambar) dan jika login salah maka akan ditampilkan pesan kesalahan (lihat gambar).



Gambar 4.8. Tampilan Program 4.8



Gambar 4.9. Tampilan Program 4.9 Jika Login Berhasil



Gambar 4.10. Tampilan Program 4.9 Jika Login Gagal

Form Input Type RADIO

Pada inputan jenis radio button, user hanya bisa memilih satu pilihan di antara beberapa pilihan.

Program 4.10

Nama File : input06.php

Deskripsi : Program menampilkan form pilihan jurusan dengan radio button

```
1 <html>
2   <head><title>Pilih Jurusan</title></head>
3   <body>
4     <FORM ACTION="proses06.php" METHOD="POST" NAME="input">
5       <h2>Pilih Jurusan Anda :</h2>
```

```

6      <input type="radio" name="jurusan" value="TI" checked>
Teknik Informatika<br>
7      <input type="radio" name="jurusan" value="SI"> Sistem
Informasi<br>
8      <input type="radio" name="jurusan" value="SK"> Sistem
Komputer<br>
9      <input type="radio" name="jurusan" value="KA">
Komputerisasi Akuntansi<br>
10     <input type="submit" name="Pilih" value="Pilih">
11     </FORM>
12   </body>
13 </html>

```

Program 4.11

Nama File : proses06.php

Deskripsi : Program untuk mengambil dan menampilkan jurusan yang dipilih pada program 4.10

```

1 <?php
2 if (isset($_POST['Pilih'])) {
3     $jurusan = $_POST['jurusan'];
4     echo "Jurusan Anda adalah
5         <b><font color='red'>$jurusan</font></b>";
6 }
7 ?>

```

Penjelasan Program 4.10 dan 4.11

Program 4.10 akan menampilkan form pilihan inputan jurusan dengan radio button (lihat gambar). Pada form inputan jenis radio button, name dari masing-masing radio button harus sama, akan tetapi value-nya harus dibedakan. Perhatikan program 4.10 baris 6-9 ! Untuk mengambil nilai (value) dari form jenis radio, bisa langsung dengan mengakses program 4.11 baris ke-3 !



Gambar 4.11. Tampilan Program 4.10



Gambar 4.12. Tampilan Program 4.11

Form Input Type CHECK BOX

Pada form inputan jenis check box, user dimungkinkan memilih lebih dari satu pilihan.

Program 4.12

Nama File : *input07.php*
 Deskripsi : Program menampilkan form inputan nama band favorit dengan check box.

```

3 <html>
2   <head><title>Band Favorit ~ Inputan Checkbox</title></head>
3   <body>
4     <FORM ACTION="proses07.php" METHOD="POST" NAME="input">
5       <h2>Pilih Band Favorit Anda :</h2>
6       <input type="checkbox" name="band01" value="Padi"
checked> Padi<br>
7       <input type="checkbox" name="band02" value="Sheila On
7"> Sheila On 7<br>
8       <input type="checkbox" name="band03" value="Dewa 19">
Dewa 19<br>
9       <input type="checkbox" name="band04" value="Ungu">
Ungu<br>
10      <input type="submit" name="Pilih" value="Pilih">
11    </FORM>
12  </body>
13 </html>
```

Program 4.13

Nama File : *proses07.php*
 Deskripsi : Program untuk menampilkan nama band favorit sesuai dengan inputan pada program 4.12

```

1 <?php
2 if (isset($_POST['Pilih'])) {
3   echo "Band Favorit Anda adalah :<br>";
4   if (isset($_POST['band01'])) {
5     echo "+" . $_POST['band01'] . "<br>";
6   }
7   if (isset($_POST['band02'])) {
8     echo "+" . $_POST['band02'] . "<br>";
9   }
10  if (isset($_POST['band03'])) {
11    echo "+" . $_POST['band03'] . "<br>";
12  }
13  if (isset($_POST['band04'])) {
14    echo "+" . $_POST['band04'] . "<br>";
```

```
15 }  
16 }  
17 ?>
```

Penjelasan Program 4.12 dan 4.13

Program 4.12 akan menampilkan form pilihan inputan jenis check box, check box name dari masing-masing check box harus dibedakan. User dapat memilih lebih dari satu pilihan. Perhatikan program 4.12 baris 6-9 ! Untuk mengambil nilai (value) dari form jenis check box, sebaiknya diperiksa terlebih dahulu apakah check box dipilih atau tidak, dengan menggunakan fungsi **isset()**. Perhatikan program 4.13 baris ke-4. Jika check box di-cek (dipilih) maka ambil *value* dari check box tersebut

(baris ke-5)



Gambar 4.13. Tampilan Program 4.12



Gambar 4.14. Tampilan Program 4.13

Form Input Type COMBO BOX

Program 4.14

Nama File

: *input08.php*

Deskripsi

: Program menampilkan form inputan film kartun favorit dengan combo box.

```

1 <html>
2 <head><title>Film Kartun Favorit ~ Inputan Combo box</title></head>
3   <body>
4     <FORM ACTION="proses08.php" METHOD="POST" NAME="input">
5       <h2>Pilih Film Kartun Favorit Anda :</h2>
6       <select name="kartun">
7         <option value="Sponge Bob">Sponge Bob</option>
8         <option value="Sinchan">Sinchan</option>
9         <option value="Conan">Conan</option>
10        <option value="Doraemon">Doraemon</option>
11        <option value="Dragon Ball">Dragon Ball</option>
12        <option value="Naruto">Naruto</option>
13      </select>
14      <input type="submit" name="Pilih" value="Pilih">
15    </FORM>
16  </body>
17 </html>

```

Program 4.15

Nama File : proses08.php
 Deskripsi : Program untuk menampilkan nama film kartun favorit sesuai dengan inputan pada program 4.14

```

1 <?php
2 if (isset($_POST['Pilih'])) {
3   $film = $_POST['kartun'];
4   echo "Film Kartun Favorit Anda adalah :
5       <font color=blue><b>$film</b></font>";
6 }
7 ?>

```

Penjelasan Program 4.14 dan 4.15

Program 4.14 akan menampilkan form pilihan inputan jenis *combo box* film kartun favorit (lihat gambar). Untuk membuat inputan jenis *combo box*, bisa menggunakan tag *<select>* dan *<option>*. Pada form inputan jenis *check box*, name diletakkan pada tag *<select>*. User hanya dapat memilih satu pilihan dari sejumlah pilihan yang ditampilkan dalam bentuk *drop down list*. Lihat gambar. Untuk mengambil nilai (*value*) dari form jenis *combo box*, dapat langsung mengaksesnya sesuai dengan name-nya. Perhatikan program 4.15 baris ke-3 !



Gambar 4.15. Tampilan Program 4.14



Gambar 4.16. Tampilan Program 4.15

Form Input Type TEXTAREA

Program 4.16

Nama File : input09.php
 Deskripsi : Program menampilkan form inputan kritik dan saran dengan text area.

```

1 <html>
2   <head><title>Kritik dan Saran ~ Inputan
3 Textarea</title></head>
4   <body>
5     <FORM ACTION="proses09.php" METHOD="POST" NAME="input">
6       <h2>Input Kritik / Saran :</h2>
7       <textarea name="saran" cols="40"
8 rows="5"></textarea><br>
9       <input type="submit" name="Proses" value="Input
10 Saran">
11     </FORM>
12   </body>
13 </html>
```

Program 4.17

Nama File : proses09.php
 Deskripsi : Program untuk menampilkan isi kritik / saran sesuai dengan inputan text area pada program 4.16

```

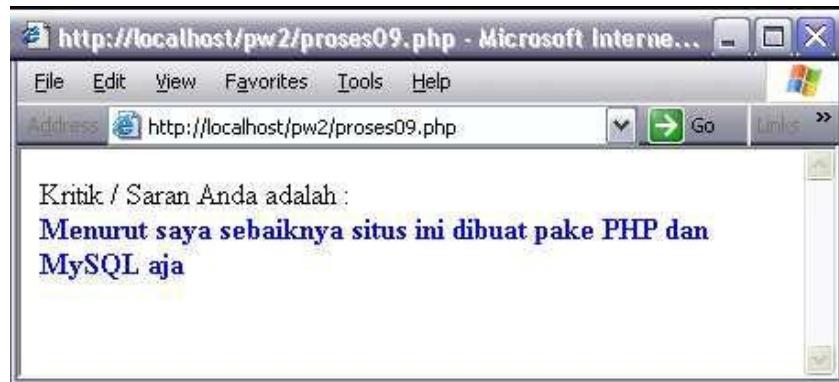
1 <?php
2 if (isset($_POST['Proses'])) {
3   $saran = nl2br($_POST['saran']);
4   echo "Kritik / Saran Anda adalah : <br>";
5   echo "<font color=blue><b>$saran</b></font>";
6 }
7 ?>
```

Penjelasan Program 4.16 dan 4.17

Program 4.16 akan menampilkan form kritik saran menggunakan text area (lihat gambar). Untuk membuat inputan jenis combo box, bisa menggunakan tag <textarea>. Berbeda dengan inputan type text yang hanya bisa diinput satu baris, pada text area, bisa diinput lebih dari satu baris. Lihat gambar. Untuk mengambil nilai (value) dari form jenis <textarea>, dapat langsung mengaksesnya sesuai dengan name-nya. Perhatikan program 4.17 baris ke-3 !



Gambar 4.17. Tampilan Program 4.16



Gambar 4.18. Tampilan Program 4.17

Bab 5

Array dan Fungsi

- TM Pengantar Array
- TM Mendeklarasikan dan Menampilkan Array
- TM Fungsi-fungsi Array dalam PHP
- TM Fungsi dalam PHP

Pengantar Array

Array merupakan tipe data terstruktur yang berguna untuk menyimpan sejumlah data yang bertipe sama. Bagian yang menyusun array disebut elemen array, yang masing-masing elemen dapat diakses tersendiri melalui index array. Index array dapat berupa bilangan integer atau string.

Mendeklarasikan dan Menampilkan Array

Program 5.1

Nama File : array01.php
Deskripsi : Program mendeklarasikan array dengan index numeric.

14

```
1 <?php
2
3 $arrBuah = array ("Mangga", "Apel", "Pisang", "Jeruk");
4 echo $arrBuah[0]; //Mangga
5 echo $arrBuah[3]; //Jeruk
6
7 $arrWarna = array();
8 $arrWarna[] = "Merah";
9 $arrWarna[] = "Biru";
10 $arrWarna[] = "Hijau";
11 $arrWarna[] = "Putih";
12 echo $arrWarna[0]; //Merah
13 echo $arrWarna[2]; //Hijau
14
15 ?>
```

Penjelasan Program 5.1

Untuk mendeklarasikan atau mendefinisikan sebuah array di PHP bisa menggunakan keyword **array()**. Pada program 5.1, terdapat pendeklarasian dan pendefinisian array pada baris ke-3. Jumlah elemen array tidak perlu disebutkan saat deklarasi. Sedangkan untuk menampilkan isi array pada elemen tertentu, cukup dengan menyebutkan nama array beserta index array-nya (lihat baris ke-4 dan ke-5). Pendeklarasian dan pengisian array juga bisa dilakukan seperti pada baris ke-7 sampai baris ke-11. Dengan pendefinisian array seperti pada dua cara di atas, index (key) array secara otomatis akan diisi dengan bilangan bulat integer yang dimulai dari 0.



Gambar 5.1. Tampilan Program 5.1

Program 5.2

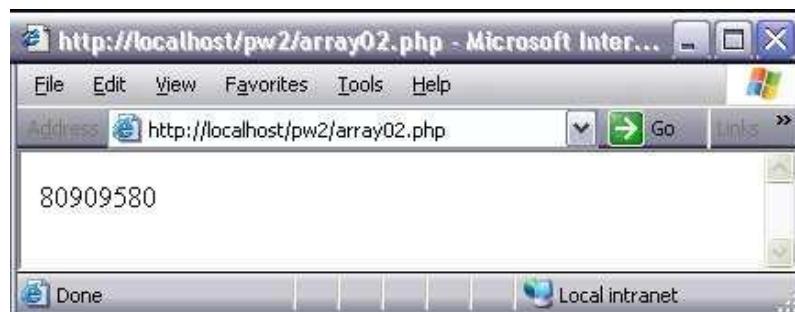
Nama File : array02.php
 Deskripsi : Program mendeklarasikan array dengan index string (array assosiatif).

```

1 <?php
2
3 $arrNilai = array ("Ani" => 80, "Otim" => 90, "Ana" => 75,
4 "Budi" => 85);
5 echo $arrNilai['Ani'];           //80
6 echo $arrNilai['Otim'];         //90
7
8 $arrNilai = array();
9 $arrNilai['Ami'] = 80;
10 $arrNilai['Asma'] = 95;
11 $arrNilai['Sri'] = 77;
12 echo $arrNilai['Asma'];        //95
13 echo $arrNilai['Ami'];         //80
14 ?>
```

Penjelasan Program 5.2

Untuk mendeklarasikan atau mendefinisikan sebuah array *assosiatif* (array yang menggunakan index selain integer), dapat dengan cara menyebutkan index-nya terlebih dahulu diikuti operator => dan diikuti value atau nilai elemennya. Lihat program baris ke-3 ! Sedangkan untuk menampilkan value atau nilai dari suatu array, bisa dengan menyebutkan nama array yang diikuti index arraynya (lihat baris ke-4 dan ke-5). Pendefinisian array *assosiatif* juga bisa dilakukan seperti pada baris ke-7 sampai baris ke-10



Gambar 5.2. Tampilan Program 5.2

Program 5.3

Nama File : array03.php
 Deskripsi : Program menampilkan seluruh isi array dengan index numerik.

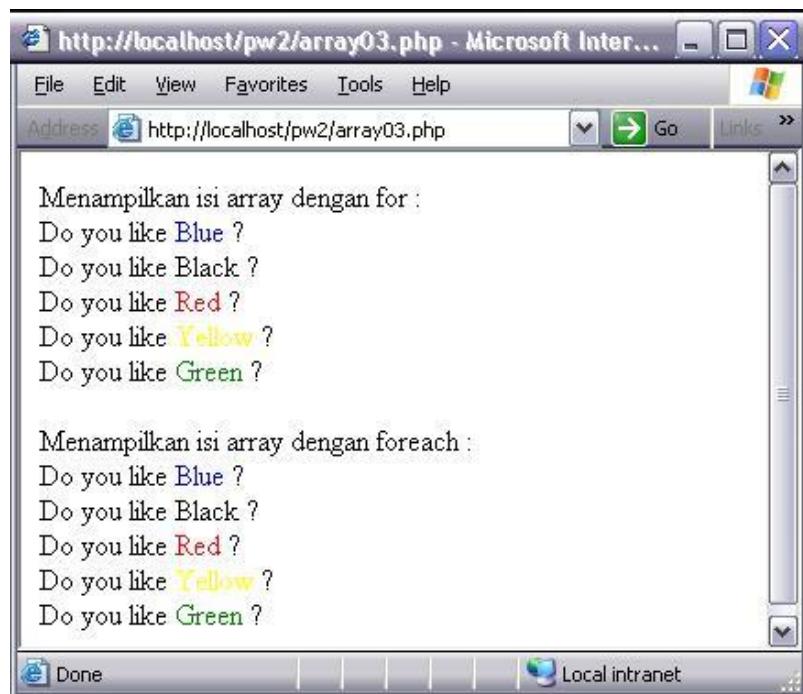
```

1 <?php
2 $arrWarna = array ("Blue","Black","Red","Yellow","Green");
3
4 echo "Menampilkan isi array dengan for : <br>";
5 for ($i=0; $i<count($arrWarna); $i++) {
6     echo "Do you like <font color=$arrWarna[$i]>". $arrWarna[$i]
7     . "</font> ?<br>"; 7 } 8
8
9 echo "<br>Menampilkan isi array dengan foreach : <br>";
10 foreach ($arrWarna as $warna) {
11     echo "Do you like <font color=$warna>". $warna . "</font> ?<br>";
12 }
13 ?>

```

Penjelasan Program 5.3

Untuk menampilkan keseluruhan isi array bisa dilakukan dengan perulangan for, while, do..while atau foreach. Baris 5 – 7 program di atas merupakan contoh menampilkan array dengan perulangan for. Fungsi **count()** pada baris ke-5 berfungsi untuk mencari jumlah elemen dari array \$assWarna. Untuk menampilkan array juga bisa menggunakan struktur **foreach()**. Struktur foreach akan mengambil elemen array mulai dari posisi pointer sampai akhir pointer. Pada contoh program di atas (baris ke-10) bisa dinyatakan dengan pernyataan bahwa “untuk setiap elemen array \$arrWarna, masukkan ke variabel \$warna”. Tampilan program 5.3 bisa dilihat pada gambar 5.3



Gambar 5.3. Tampilan Program 5.3

Program 5.4

*Nama File : array04.php
Deskripsi : Program menampilkan seluruh isi array assosiatif.*

```

1 <?php
2 $arrNilai = array ("Ani" => 80, "Otim" => 90, "Ana" => 75, "Budi" => 85);

```

```

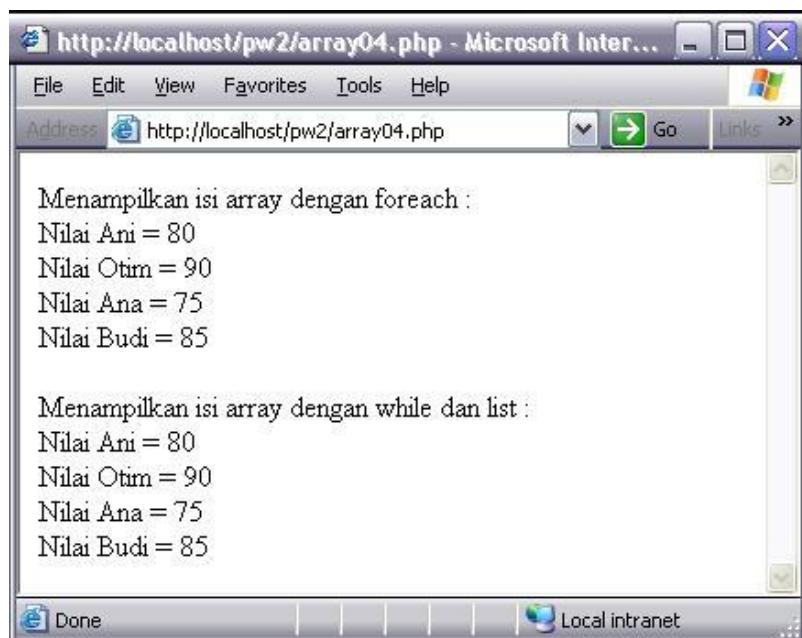
3 echo "Menampilkan isi array dengan foreach : <br>"; 5 foreach
4 ($arrNilai as $nama => $nilai) {
5     echo "Nilai $nama = $nilai<br>";
6 }
7 reset ($arrNilai);
8 echo "<br>Menampilkan isi array dengan while dan list : <br>";
9 while (list ($nama, $nilai) = each($arrNilai)) {
10    echo "Nilai $nama = $nilai<br>";
11 }
12
13
14 ?>

```

Penjelasan Program 5.4

Untuk menampilkan array assosiatif bisa menggunakan perulangan foreach dan while dengan fungsi list() dan each(). Baris ke-5 dari program 5.4. di atas bisa dinyatakan dengan pernyataan “untuk setiap elemen array \$arrNilai, masukkan key/index-nya ke variabel \$nama dan masukkan value-nya ke variabel

\$nilai”. Pada baris ke-8 terdapat fungsi **reset()** yang berfungsi mengembalikan pointer array ke elemen pertama. Pada baris ke-10, array assosiatif ditampilkan dengan struktur while yang menggunakan fungsi **list()** dan **each()**.



Gambar 5.4. Tampilan Program 5.4

Program 5.5

Nama File : array05.php

Deskripsi : Program mencetak struktur array.

```

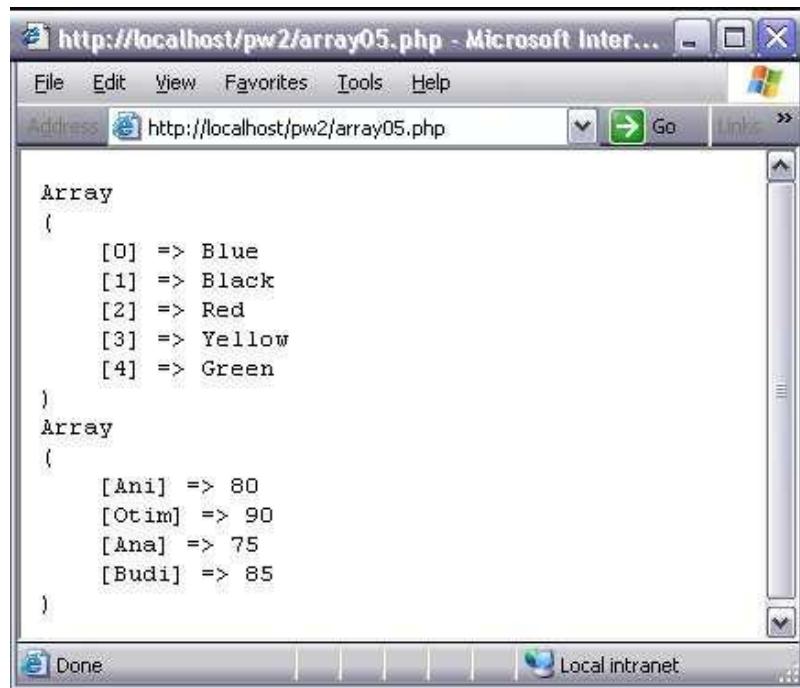
1 <?
2 $arrWarna = array ("Blue", "Black", "Red", "Yellow", "Green");
3 $arrNilai = array ("Ani" => 80, "Otim" => 90, "Ana" => 75,
4 "Budi" => 85);
5 echo "<pre>";
6 print_r ($arrWarna);
7 echo "<br>";
8 print_r ($arrNilai);
9 echo "</pre>";
?>

```

Penjelasan Program 5.5

Pada program 5.5 di atas terdapat yang berfungsi untuk menampilkan struktur 5.5 bisa dilihat pada gambar 5.5

fungsi **print_r()** (baris ke-5) dari array. Tampilan program



The screenshot shows a Microsoft Internet Explorer window with the URL <http://localhost/pw2/array05.php>. The page displays the output of the PHP `print_r()` function. It shows two arrays:

```
Array
(
    [0] => Blue
    [1] => Black
    [2] => Red
    [3] => Yellow
    [4] => Green
)
Array
(
    [Ani] => 80
    [Otim] => 90
    [Ana] => 75
    [Budi] => 85
)
```

Gambar 5.5. Tampilan Program 5.5

Fungsi-fungsi Array dalam PHP

PHP menyediakan lebih dari 70 fungsi untuk manipulasi array. Fungsi-fungsi array dalam PHP bisa dilihat di alamat <http://ca.php.net/manual/en/ref.array.php>.

Fungsi Pengurutan Array

- f **arsort()** – Pengurutan array berdasarkan **value** secara **descending**
- f **asort()** – Pengurutan array berdasarkan **value** secara **ascending**
- f **krsort()** - Pengurutan array berdasarkan **index/key** secara **descending**
- f **ksort()** - Pengurutan array berdasarkan **index/key** secara **ascending**
- f **rsort()** - Pengurutan array berdasarkan **value** secara **descending** dengan mengubah **index/key**
- f **sort()** - Pengurutan array berdasarkan **value** secara **ascending** dengan mengubah **index/key**
- f **shuffle()** – Random pengurutan array

Fungsi Pengaturan Pointer Array

- f **current()** – Mendapatkan elemen array yang ditunjuk oleh pointer
- f **end()** – Pointer menunjuk pada elemen array terakhir
- f **key()** – Mendapatkan key yang ditunjuk oleh pointer
- f **next()** – Pointer menunjuk pada elemen selanjutnya
- f **prev()** – Pointer menunjuk pada elemen sebelumnya
- f **reset()** – Memindahkan pointer array ke awal (elemen pertama)
- f **count()** – Menghitung jumlah elemen array

Fungsi Pencarian pada Array

- f **array_search()** – Mencari posisi (key) dari suatu value dalam array
- f **array_key_exists()** – Memeriksa suatu key ada dalam array atau tidak
- f **in_array()** – Memeriksa suatu elemen ada dalam array atau tidak

Program 5.6

Nama File : array06.php

Deskripsi : Program mengurutkan array dengan sort() dan rsort().

```
1 <?php
2 $arrNilai = array ("Ani" => 80, "Otim" => 90, "Sri" => 75,
3 "Budi" => 85);
4 echo "<b>Array sebelum pengurutan</b>";
5 print_r($arrNilai);
6 echo "</pre>";
7
8 sort($arrNilai);
9 reset($arrNilai);
10 echo "<b>Array setelah pengurutan dengan sort()</b>";
11 echo "<pre>";
12 print_r($arrNilai);
13 echo "</pre>";
14
15 rsort($arrNilai);
16 reset($arrNilai);
17 echo "<b>Array setelah pengurutan dengan rsort()</b>";
18 echo "<pre>";
19 print_r($arrNilai);
20 echo "</pre>";
21 ?>
```

Program 5.7

Nama File : array07.php

Deskripsi : Program mengurutkan array dengan asort() dan arsort().

```
1 <?php
2 $arrNilai = array ("Ani" => 80, "Otim" => 90, "Sri" => 75, "Budi" => 85);
3 echo "<b>Array sebelum pengurutan</b>";
4 echo "<pre>";
5 print_r($arrNilai);
6 echo "</pre>";
7
8 asort($arrNilai);
9 reset($arrNilai);
10 echo "<b>Array setelah pengurutan dengan asort()</b>";
11 echo "<pre>";
12 print_r($arrNilai);
13 echo "</pre>";
14
15 arsort($arrNilai);
16 reset($arrNilai);
17 echo "<b>Array setelah pengurutan dengan arsort()</b>";
18 echo "<pre>";
19 print_r($arrNilai);
20 echo "</pre>";
21 ?>
```

Program 5.8

Nama File : array08.php

Deskripsi : Program mengurutkan array dengan ksort() dan krsort().

```

1 <?php
2 $arrNilai = array ("Ani"=>80, "Otim"=>90, "Sri"=>75, "Budi"=>85);
3 echo "<b>Array sebelum pengurutan</b>";
4 echo "<pre>";
5 print_r($arrNilai);
6 echo "</pre>";
7
8 ksort($arrNilai);
9 reset($arrNilai);
10 echo "<b>Array setelah pengurutan dengan ksort()</b>";
11 echo "<pre>";
12 print_r($arrNilai);
13 echo "</pre>";
14
15 krsort($arrNilai);
16 reset($arrNilai);
17 echo "<b>Array setelah pengurutan dengan krsort()</b>";
18 echo "<pre>";
19 print_r($arrNilai);
20 echo "</pre>";
21 ?>

```

Program 5.9

Nama File : array09.php

Deskripsi : Program mengatur posisi pointer dalam array.

```

1 <?php
2 $transport = array('foot', 'bike', 'car', 'plane');
3 echo "<pre>";
4 print_r ($transport);
5 echo "</pre>";
6
7 $mode = current($transport);
8 echo $mode."<br>"; // $mode = 'foot';
9 $mode = next($transport);
10 echo $mode."<br>"; // $mode = 'bike';
11 $mode = current($transport);
12 echo $mode."<br>"; // $mode = 'bike';
13 $mode = prev($transport);
14 echo $mode."<br>"; // $mode = 'foot';
15 $mode = end($transport);
16 echo $mode."<br>"; // $mode = 'plane';
17 $mode = current($transport);
18 echo $mode."<br>"; // $mode = 'plane';
19 ?>

```

Program 5.10

Nama File : array10.php

Deskripsi : Program mencari elemen array.

```

1 <?php
2 $arrBuah = array ("Mangga", "Apel", "Pisang", "Kedondong",
3 "Jeruk");
4 if (in_array ("Kedondong", $arrBuah)) {
5     echo "Ada buah Kedondong di sini";
6 } else {
7     echo "Tidak ada buah Kedondong di sini";
8 }
9 ?>

```

Fungsi dalam PHP

Bentuk umum pendefinisan fungsi dalam PHP

```
function nama_fungsi(parameter1,...,n) { statement2;  
}
```

Program 5.11

Nama File : fungsi01.php

Deskripsi : Program sederhana mendefinisikan dan memanggil fungsi tanpa parameter dan tanpa return value.

```
1 <?php  
2 function cetak_ganjil () {  
3     for ($i=0; $i<100; $i++) {  
4         if ($i%2 == 1) {  
5             echo "$i ";  
6         }  
7     }  
8 } //pemanggilan fungsi  
9 cetak_ganjil();  
10 ?>
```

Program 5.12

Nama File : array02.php

Deskripsi : Program fungsi dengan parameter.

```
1 <?php  
2 function cetak_ganjil ($awal, $akhir) { 3 for ($i=$awal;  
$i<$akhir; $i++) {  
4         if ($i%2 == 1) {  
5             echo "$i ";  
6         }  
7     }  
8 } //pemanggilan fungsi  
9 $a = 10;  
10 $b = 50;  
11 echo "<b>Bilangan ganjil dari $a sampai $b : </b><br>";  
12 cetak_ganjil($a, $b);  
13 ?>
```

Program 5.13

Nama File : fungsi03.php

Deskripsi : Program fungsi yang mengembalikan nilai.

```
1 <?php  
2 function luas_lingkaran ($jari) {  
3     return 3.14 * $jari * $jari;  
4 }  
5 //pemanggilan fungsi  
6 $r = 10;  
7 echo "Luas lingkaran dengan jari-jari $r = "; 8 echo  
luas_lingkaran($r);  
9 ?>
```

Program 5.14

Nama File : fungsi04.php

Deskripsi : Program passing by value dalam fungsi.

```

1 <?php
2 function tambah_string ($str) {
3     $str = $str . ", Jakarta";
4     return $str;
5 }
6 //
7 $str = "Universitas Budi Luhur";
8 echo "\$str = $str<br>";
9 echo tambah_string ($str). "<br>";
10 echo "\$str = $str<br>";
11 ?>
```

Program 5.15

Nama File : fungsi05.php

Deskripsi : Program passing by reference dalam fungsi.

```

1 <?php
2 function tambah_string (&$str) {
3     $str = $str . ", Jakarta";
4     return $str;
5 }
6 //
7 $str = "Universitas Budi Luhur";
8 echo "\$str = $str<br>";
9 echo tambah_string ($str). "<br>";
10 echo "\$str = $str<br>";
11 ?>
```

Program 5.16

Nama File : fungsi06.php

Deskripsi : Program menampilkan defined-function yang didukung oleh PHP.

```

1 <?php
2 function luas_lingkaran ($jari) {
3     return 3.14 * $jari * $jari;
4 }
5
6 $arr = get_defined_functions();
7 echo "<pre>";
8 print_r($arr);
9 echo "</pre>";
10 ?>
```

Program 5.17

Nama File : fungsi07.php

Deskripsi : Program memeriksa suatu fungsi tersedia di PHP atau tidak.

```

1 <?php
2 if (function_exists('exif_read_data')) {
3     echo "Fungsi exif_read_data() ada di PHP.<br />\n";
4 } else {
5 echo "Fungsi exif_read_data() tidak ada di PHP.<br />\n"; 6 }
7 ?>
```

Bab 6

Penanganan String dan Tanggal

- | | |
|----|---|
| TM | Pengantar String |
| TM | Fungsi-fungsi String dalam PHP |
| TM | Fungsi-fungsi Operasi Tanggal dalam PHP |

Pengantar String

String merupakan kumpulan dari karakter. Dalam PHP, karakter tidak sama dengan byte, dimana terdapat 256 karakter. PHP tidak mendukung native-unicode.

Untuk menuliskan sebuah string dalam cara, yaitu dengan :

1. single quoted – kutip tunggal (')
2. double quoted – kutip ganda (")
3. heredoc sintaks

Program 6.1

Nama File : string01.php
Deskripsi : Program string dengan single quoted (').

```
1 <?php
2 echo 'this is a simple string';
3
4 echo 'You can also have embedded newlines in
5 strings this way as it is
6 okay to do';
7
8 // Outputs: Arnold once said: "I'll be back"
9 echo 'Arnold once said: "I'll be back"';
10
11 // Outputs: You deleted C:\*.*?
12 echo 'You deleted C:\*.*?';
13
14 // Outputs: You deleted C:\*.*?
15 echo 'You deleted C:\*.*?';
16
17 // Outputs: This will not expand: \n a newline
18 echo 'This will not expand: \n a newline';
19
20 // Outputs: Variables do not $expand $either
21 echo 'Variables do not $expand $either';
22 ?>
```

Program 6.2

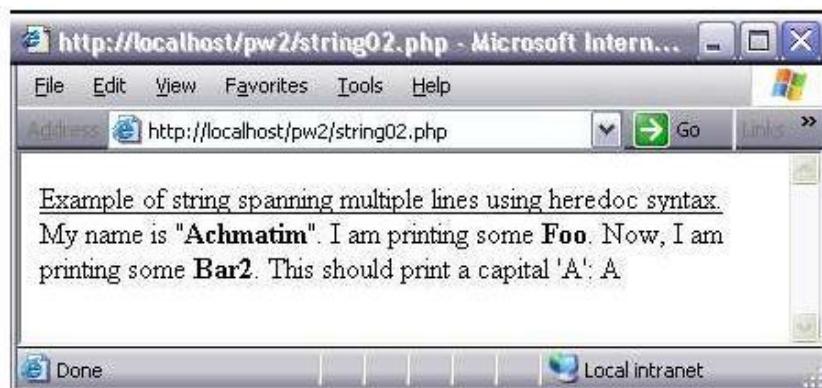
Nama File : string02.php
Deskripsi : Program string dengan heredoc sintaks.

```
1 <?php
2 $str = <<<EOD
3 Example of string
4 spanning multiple lines
5 using heredoc syntax.
6 EOD;
7
8 /* More complex example, with variables. */
9 class foo
10 {
11     var $foo;
```

```

12     var $bar;
13
14     function foo()
15     {
16         $this->foo = 'Foo';
17         $this->bar = array('Bar1', 'Bar2', 'Bar3');
18     }
19 }
20
21 $foo = new foo();
22 $name = 'Achmatim';
23
24 echo <<<EOT
25 <u>$str</u><br>
26 My name is <b>$name</b>. I am printing some <b>$foo->foo</b>.
27 Now, I am printing some <b>{$foo->bar[1]}</b>.
28 This should print a capital 'A': \x41
29 EOT;
30 ?>

```



Gambar 6.1. Tampilan Program 6.2

Fungsi-fungsi String dalam PHP

PHP menyediakan lebih dari 90 fungsi untuk manipulasi string. Fungsi-fungsi string dalam PHP bisa dilihat di <http://ca.php.net/manual/en/ref.strings.php>.

Beberapa fungsi manipulasi string yang sering digunakan, antara lain :

- f **addslashes()** – menambahkan backslashes (\) di setiap tanda kutip (quote) dalam string.
- f **chr()** – menghasilkan karakter dari bilangan ASCII
- f **crypt()** – menghasilkan string terenkripsi searah.
- f **echo()** – menampilkan satu atau beberapa string.
- f **explode()** – memecah string berdasarkan suatu delimiter (separator).
- f **htmlentities ()** – mengubah semua tag html menjadi HTML entities.
- f **htmlspecialchars()** – mengubah semua karakter khusus menjadi HTML entities
- f **implode()** – menggabungkan elemen array menjadi string dengan suatu delimiter (separator).
- f **join()** – sama dengan implode()
- f **ltrim()** – menghapus karakter tertentu (" ", \t, \n, \r, \0, \x0B) di awal string.
- f **md5()** – menghasilkan string terenkripsi searah.
- f **nl2br()** – menambahkan HTML line break (
) sebelum semua baris baru dalam string.
- f **number_format()** – mengatur format angka.
- f **ord()** – menghasilkan kode ASCII dari suatu karakter.

-
- f **`trim()`** - menghapus karakter tertentu (" ", \t, \n, \r, \0, \x0B) di akhir string.
- f **`split()`** – sama dengan explode
- f **`str_repeat()`** – mengulang string
- f **`str_replace()`** – mengganti semua string dalam pola menjadi suatu string.
- f **`strip_tags()`** – mengabaikan HTML dan PHP tag dalam string.
- f **`stripslashes()`** – menghilangkan backslashes (\) dalam string.
- f **`strlen()`** – menghitung panjang string.
- f **`strpos()`** – mencari posisi pertama sebuah string dalam string.
- f **`strrchr()`** – mencari posisi terakhir sebuah karakter dalam string.
- f **`strrpos()`** – mencari posisi terakhir sebuah string dalam string.
- f **`strrev()`** – membalik string.
- f **`strstr()`** – mencari posisi pertama sebuah karakter dalam string.
- f **`strtolower()`** – mengubah string menjadi huruf kecil (lower-case).
- f **`strtoupper()`** – mengubah string menjadi huruf kapital (upper-case)
- f **`substr()`** – memotong string
- f **`trim()`** - menghapus karakter tertentu (" ", \t, \n, \r, \0, \x0B) di akhir dan akhir string.
- f **`ucfirst()`** – mengubah huruf pertama dari semua string menjadi Upper-case.
- f **`ucwords()`** – mengubah huruf pertama tiap kata dalam string menjadi upper-case.
- f **`wordwrap()`** – memotong sejumlah karakter dalam string dengan *string-break-character*.

Program 6.3

Nama File : *string03.php*
 Deskripsi : Program penggunaan fungsi *strtolower*, *strtoupper*, *ucfirst*, *ucwords*, *strrev*, dan *strlen* dalam string.

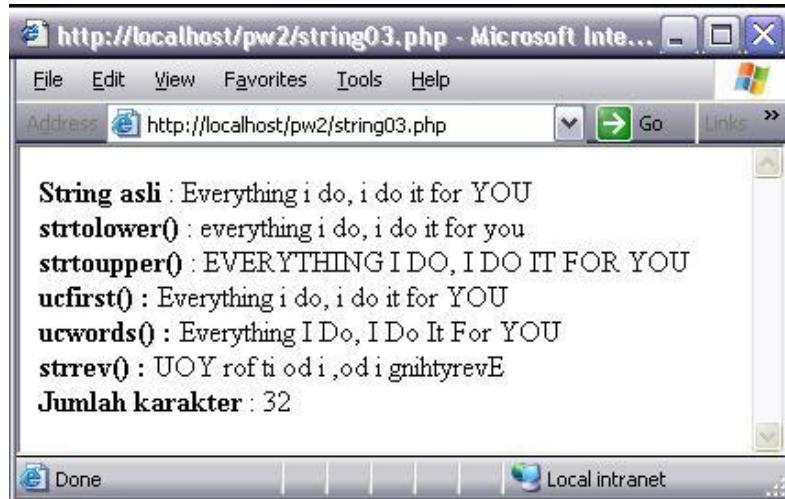
```

1 <?php
2 $str = "Everything i do, i do it for YOU"; 3
4 echo "<b>String asli</b> : $str";
5 echo "<br><b>strtolower()</b> : ". strtolower($str);
6 echo "<br><b>strtoupper()</b> : ". strtoupper($str);
7 echo "<br><b>ucfirst()</b> : ". ucfirst($str);
8 echo "<br><b>ucwords()</b> : ". ucwords($str);
9 echo "<br><b>strrev()</b> : ". strrev($str);
23 echo "<br><b>Jumlah karakter</b> : ". strlen($str);
24 ?>
  
```

Penjelasan Program 6.3

Beberapa fungsi yang digunakan dalam program 6.3 di atas antara lain :

- *strtolower*, untuk mengubah string menjadi lower-case.
- *strtoupper*, untuk mengubah string menjadi upper-case.
- *ucfirst*, untuk mengubah huruf pertama string menjadi upper-case.
- *ucwords*, untuk mengubah huruf pertama tiap kata menjadi upper-case.
- *strrev*, untuk membalik string.
- *strlen*, untuk menghitung panjang string.



Gambar 6.2 Tampilan Program 6.3

Program 6.4

*Nama File : string04.php
Deskripsi : Program penggunaan fungsi addslashes dan stripslashes pada string.*

```

1 <?php
2 $str = "Is your name O'Reilly ?";
3 $str2 = addslashes ($str);
4 $str3 = stripslashes ($str2);
5
6 echo "<b>String asli</b> : $str";
7 echo "<br><b>addslashes()</b> : $str2";
8 echo "<br><b>stripslashes()</b> : $str3";
9 ?>
  
```

Penjelasan Program 6.4

Pada program 6.4 di atas terdapat fungsi addslashes() yang akan menambahkan *backslashes* (\) pada setiap *quote* dalam string, dan sebaliknya fungsi stripslashes akan menghilangkan *backslashes* pada string.



Gambar 6.3. Tampilan Program 6.4

Program 6.5

*Nama File : string05.php
Deskripsi : Program penggunaan fungsi chr untuk menampilkan 256 karakter ASCII.*

```

1 <?php
2 echo "Menampilkan bilangan ASCII"; 3 for ($i=1;
3 $i<=256; $i++) {
4     echo "<br>$i.\t". chr($i);
5 }
6 ?>

```

Penjelasan Program 6.5

Program 6.5 di atas akan menampilkan 256 karakter ASCII. Pada baris 4, terdapat fungsi `chr()` yang akan mengubah nilai `$i` (integer) menjadi karakter ASCII.

Program 6.6

*Nama File : string06.php
Deskripsi : Program penggunaan fungsi strip_tags, htmlspecialchars, dan htmlentities untuk mengatur tag HTML.*

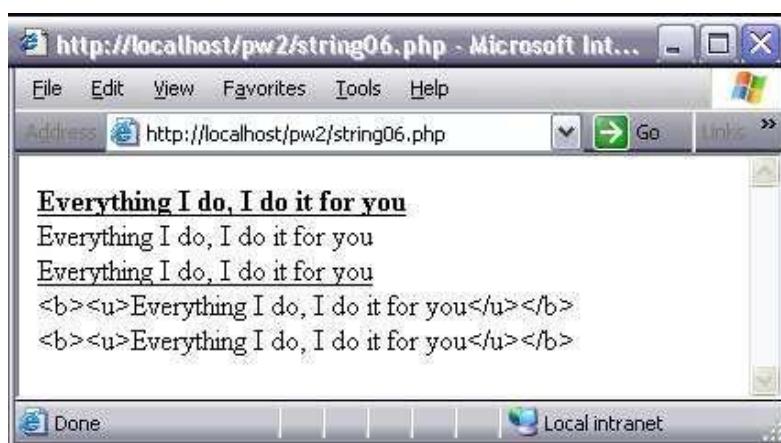
```

1 <?php
2 $str = "<b><u>Everything I do, I do it for you</u></b>";
3
4 echo $str."<br>";
5 //no tag html
6 echo strip_tags ($str). "<br>";
7 //allow tag <u>
8 echo strip_tags ($str,"<u>")."<br>";
9 //tampil apa adanya
10 echo htmlspecialchars ($str)."<br>";
11 //tampil apa adanya
12 echo htmlentities ($str);
13 ?>

```

Penjelasan Program 6.6

Program 6.6 di atas merupakan contoh penggunaan fungsi `strip_tags`, `htmlspecialchars` dan `htmlentities` untuk menangani tag-tag HTML yang ada dalam string. Pada baris ke-6, fungsi `strip_tags` akan menghilangkan semua tag HTML. Pada baris ke-8, fungsi `strip_tags` dengan tambahan parameter "`<u>`" akan menghilangkan semua tag HTML kecuali tag `<u>`. Pada baris ke-10 dan ke-12, fungsi `htmlspecialchars` dan `htmlentities` akan mengubah semua tag HTML menjadi HTML special character, sehingga tag HTML akan ditampilkan apa adanya



Gambar 6.4. Tampilan Program 6.6

Program 6.7

*Nama File : string07.php
Deskripsi : Program penggunaan fungsi number_format untuk mengatur format tampilan angka.*

```

1 <?php
2 $number = 1234.56;
3
4 // english notation (default)
5 $english_format_number = number_format($number); 6 echo
6 "<br>".$english_format_number; // 1,235
7
8 // French notation
9 $nombre_format_francais = number_format($number, 2, ',', ' '); 10 echo
10 "<br>".$nombre_format_francais; // 1 234,56
11
12 //indonesian notation
13 $format_indonesia = number_format ($number, 2, ',', '.');
14 echo "<br>".$format_indonesia; //1.234,56
15
16 $number = 1234.5678;
17 // english notation without thousands separator
18 $english_format_number = number_format($number, 2, ',', '');
19 echo "<br>". $english_format_number; // 1234.57
20
21 ?>

```

Penjelasan Program 6.7

Program 6.7 di atas merupakan contoh penggunaan fungsi `number_format` untuk mengatur tampilan angka. Secara default, fungsi `number_format` akan memformat angka dengan model Inggris yaitu setiap tiga digit dipisahkan dengan koma (,) dan tanpa pecahan. Pada baris ke-9, fungsi `number_format` akan mengubah format angka dengan 2 digit pecahan (parameter kedua), pecahan dipisahkan dengan koma (parameter ketiga) dan setiap tiga digit dipisahkan dengan spasi (parameter keempat).

Program 6.8

*Nama File : string08.php
Deskripsi : Program penggunaan fungsi `explode` untuk memecah string menjadi array.*

```

1 <?php
2 // Contoh 1
3 $buah = "mangga pisang rambutan apel jeruk kedondong";
4 $buahan = explode(" ", $buah);
5 echo $buahan[0]."<br>"; // mangga
6 echo $buahan[3]."<br>"; // apel
7
8 // Contoh 2
9 $data = "foo:*:1023:1000::/home/foo:/bin/sh";
10 list($user, $pass, $uid, $gid, $gecos, $home, $shell) = explode(":", $data);
11 echo $user."<br>"; // foo
12 echo $pass; // *
13
14 ?>

```

Penjelasan Program 6.8

Program 6.8 di atas merupakan contoh penggunaan fungsi `explode` untuk memecah string berdasarkan pola (aturan) tertentu. Pada baris ke-4, dengan fungsi `explode()` akan memecah string `$buah` berdasarkan spasi dan masing-masing akan menjadi elemen array `$buahan`.

Program 6.9

*Nama File : string09.php
Deskripsi : Program penggunaan fungsi `implode` untuk menggabungkan array menjadi string.*

```

1 <?php
2 $buahan = array('mangga','jeruk','rambutan','apel','nanas');
3 $buah = implode(", ", $buahan);
4
5 echo "Saya suka buah ". $buah;
// mangga,jeruk,rambutan,apel,nanas
6 ?>

```

Penjelasan Program 6.9

Program 6.9 di atas merupakan contoh penggunaan fungsi implode() untuk menggabungkan semua isi array menjadi satu string. Fungsi implode merupakan kebalikan dari fungsi explode. Pada baris ke-3, dengan fungsi implode() akan menggabungkan setiap elemen array \$buahan menjadi string \$buah dengan pemisah koma di antara tiap elemen.

Program 6.10

*Nama File : string10.php
Deskripsi : Program penggunaan fungsi strstr, strchr dan strrchr untuk mengambil sebagian string berdasarkan pola (karakter) tertentu.*

```

1 <?php
2 $file = "test.this.txt";
3 $ext1 = strstr($file, ".");
4 $ext2 = strchr($file, ".");
5 $ext3 = strrchr($file, ".");
6
7 echo $ext1. "<br>";           // .this.txt
8 echo $ext2. "<br>";           // .this.txt
9 echo $ext3;                  // .txt
10 ?>

```

Penjelasan Program 6.10

Program 6.10 di atas merupakan contoh pengambilan sebagian string berdasarkan pola (karakter) tertentu. Fungsi strstr dan strchr pada baris ke-3 dan ke-4 akan mengambil string setelah karakter titik (.) dimana pencarian dilakukan dari awal string. Sedangkan fungsi strrchr, akan mengambil string setelah karakter titik (.) yang mana pencarian dilakukan dari akhir string.

Program 6.11

*Nama File : string11.php
Deskripsi : Program penggunaan fungsi wordwrap untuk memotong sejumlah karakter dari string.*

```

1 <?php
2 $text = "The quick brown fox jumped over the lazy dog.";
3 $newtext = wordwrap($text, 15, "<br>");
4
5 echo $text."<br><br>";
6 echo $newtext;
7 ?>

```

Penjelasan Program 6.11

Program 6.11 di atas merupakan contoh penggunaan fungsi wordwrap untuk memotong sejumlah karakter dari string. Pada baris ke-3, fungsi wordwrap akan memotong string \$text per 15 karakter dan dipisahkan (ditambahkan) dengan string "
"



Gambar 6.5. Tampilan Program 6.11

Program 6.12

Nama File : string12.php

Deskripsi : Program penggunaan fungsi nl2br dalam string.

```

1 <form action="" method="post">
2   Input teks di sini :<br>
3   <textarea name="input" cols="40" rows="4"></textarea><br>
4   <input type="submit" name="Submit" value="Proses">
5 </form>
6
7 <?php
8 if (isset($_POST['Submit'])) {
9   $txt = $_POST['input'];
10  echo "<u>Tanpa nl2br() :</u> <br>$txt<br><br>";
11  echo "<u>Dengan nl2br :</u> <br>". nl2br ($txt);
12 }
14 ?>
```

Penjelasan Program 6.12

Program 6.12 di atas merupakan contoh penggunaan fungsi nl2br. Fungsi nl2br() akan menambahkan karakter pindah baris (
) di setiap awal baris string inputan. Untuk mencoba program di atas, inputkan string sebagai berikut :

bila yang tertulis untukku
 adalah yang terbaik untukku kan
 kujadikan kau kenangan yang terindah
 dalam hidupku namun tak kan mudah
 bagiku meninggalkan jejak hidupku yang
 tlah terukir abadi sebagai kenangan
 yang terindah

Program 6.13

Nama File : string13.php

Deskripsi : Program penggunaan fungsi substr untuk memotong string.

```

1 <form action="" method="post">
2   NIM Anda :
3     <input type="text" name="txtnim"><br>
4     <input type="submit" name="Submit" value="Proses">
5 </form>
6
7 <?php
8 if (isset($_POST['Submit'])) {
9   $nim = $_POST['txtnim'];
```

```

10 $a = substr ($nim, 2, 2);
11 switch($a) {
12     case '11' : $jurusan = "Teknik Informatika";
13         break;
14     case '22' : $jurusan      = "Sistem Informasi";
15         break;
16     case '33' : $jurusan      = "Sistem Komputer";
17         break;
18     case '44' : $jurusan      = "Komputerisasi Akuntansi";
19         break;
20     default   : $jurusan = "Salah jurusan";
21 }
22 echo "Nim Anda : $nim<br>";
23 echo "Jurusan Anda : $jurusan";
24 }
25 ?>

```

Penjelasan Program 6.13

Program 6.13 di atas merupakan contoh penggunaan fungsi substr (baris ke-10) untuk memotong string. Pada baris ke-10 tersebut, string \$nim akan dipotong mulai karakter index ke-2 (index dimulai dari 0) sepanjang 2 karakter.

Fungsi-fungsi Operasi Tanggal dalam PHP

Fungsi dalam Operasi tanggal dalam PHP yang utama dan sering digunakan adalah fungsi date(). Fungsi ini akan menghasilkan tanggal dan waktu server sekarang. Beberapa pilihan parameter dari fungsi date() ini dapat dilihat pada tabel berikut ini :

parameter	Keterangan	Contoh nilai
Hari		
d	Tanggal, 2 digit dengan nol	01 s/d 31
D	Tiga digit nama hari dalam seminggu	
j	Tanggal tanpa nol	Mon s/d Sun 1 s/d 31
I (lowercase L)	Nama hari lengkap dalam seminggu	Sunday through Saturday
N	Urutan hari dalam seminggu	1 (untuk Monday) s/d 7 (untuk Sunday)
S	Akhiran angka English untuk tanggal, 2 karakter	st, nd, rd atau th.
w	Urutan hari dalam seminggu	0 (untuk Sunday) s/d 6 (untuk Saturday)
z	Urutan hari dalam setahun	0 s/d 365
Minggu		
W	Urutan minggu dalam setahun	Contoh: 42 (minggu ke-42 dalam tahun ini)
Bulan		
F	Nama bulan lengkap	January s/d December
m	Urutan bulan dalam setahun dengan nol	01 s/d 12
M	Tiga digit nama bulan dalam	Jan s/d Dec

n	setahun Urutan bulan dalam setahun,	1 s/d 12

parameter	Keterangan	Contoh nilai
t	tanpa nol Jumlah hari dalam tiap bulan	28 s/d 31
Tahun		
Y	4 (empat) digit tahun	Contoh: 1999 atau 2006
y	2 (dua) digit tahun	Contoh: 99 or 06
Waktu		
a	Lowercase Ante meridiem dan Post meridiem	am atau pm
A	Uppercase Ante meridiem dan Post meridiem	AM atau PM
g	Jam format 12 tanpa nol	1 s/d 12
G	Jam format 24 tanpa nol	0 s/d 23
h	Jam format 12 dengan nol	01 s/d 12
H	Jam format 24 dengan nol	00 s/d 23
i	Menit dengan nol	00 s/d 59
s	Detik dengan nol	00 s/d 59

Program 6.14

Nama File : date01.php

Deskripsi : Program menampilkan date/time dengan berbagai parameter.

```

1 <?php
2 // Sekarang tanggal 13 April 2006 09:43:49
3
4 echo "<br>". date("d/m/Y H:i:s");           // 13/04/2006 09:43:03
5 echo "<br>". date("F j, Y, g:i a");        // April 13, 2006, 9:43 am
6 echo "<br>". date("d.m.y");                  // 13.04.06
7 echo "<br>". date("Ymd");                   // 20060413
8 // 13-04-06, 4330 4349 4 Thuam06 102
9 echo "<br>". date('j-m-y, it is w Day z ');
10 // it is the 13th day.
11 echo "<br>". date('i\lt \lS \t\h\le jS \d\aly.');
12 // Thu Apr 13 9:43:49 KRAST 2006
13 echo "<br>". date("D M j G:i:s T Y");
14 echo "<br>". date("H:i:s");                  // 17:16:17
15 echo "<br>". date("H:i:s");
16 ?>

```

Program 6.15

Nama File : date02.php

Deskripsi : Program menampilkan nama hari dalam bahasa Indonesia.

```

1 <?php
2 // Sekarang tanggal 13 April 2006 09:43:49
3 $arrDay = array("Minggu", "Senin", "Selasa", "Rabu", "Kamis",
"Jum'at", "Sabtu");
4 $day = date ("w"); //0 - 6 of day 5
5
6 echo "Hari ini hari : <b>". $arrDay[$day]. "</b>";
7 ?>

```

Bab 7

File dan Direktori

- TM Membuat, Menulis dan Membaca Isi File
 - + Membuka dan Membuat File
 - + Menulis ke File
 - + Membaca Isi File
 - + Menutup File
- TM Membuat dan Menghapus Direktori
- TM Manipulasi File
 - + Mendapatkan Informasi File
 - + Copy, Rename dan Delete File
- TM Upload File

Penanganan File

Dalam management file dan direktori, PHP menyediakan lebih dari 70 fungsi. Beberapa fungsi utama yang berhubungan dengan management file (create, write, append, dan delete), antara lain :

Membuka dan Membuat File

```
fopen ($namafile, $mode);
```

Keterangan :

\$namafile merupakan nama file yang akan dibuat, sedangkan \$mode merupakan mode akses file. Mode akses file yang bisa digunakan yaitu :

Mode	Keterangan
r	Hanya untuk baca file, pointer berada di awal file
r+	Untuk baca dan tulis file, pointer berada di awal file
w	Hanya untuk tulis file, isi file lama dihapus, jika file belum ada maka akan di-create
w+	Untuk baca dan tulis file, isi file lama dihapus, jika file belum ada maka akan di-create
a	Hanya untuk menambahkan isi file, pointer berada di akhir file, jika file belum ada maka di-create
a+	Untuk membaca dan menambahkan isi file, pointer berada di akhir file, jika file belum ada maka di-create

Program 7.1

Nama File : file01.php

Deskripsi : Program mengakses (membuka) file dengan mode r.

```
1 <?php
2 $namafile = "data.txt";
3 $handle = fopen ($namafile, "r");
4 if (!$handle) {
5     echo "<b>File tidak dapat dibuka atau belum ada</b>";
6 } else {
7     echo "<b>File berhasil dibuka</b>";
8 }
9 fclose($handle);
10 ?>
```



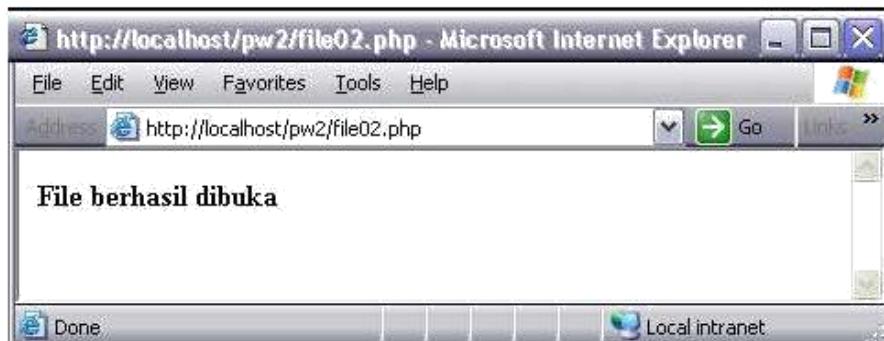
Gambar 7.1 Tampilan Program 7.1

Program 7.2

Nama File : file02.php
 Deskripsi : Program mengakses (membuka) file dengan mode w.

```

1 <?php
2 $namafile = "data.txt";
3 $handle = fopen ($namafile, "w");
4 if (!$handle) {
5     echo "<b>File tidak dapat dibuka atau belum ada</b>";
6 } else {
7     echo "<b>File berhasil dibuka</b>";
8 }
9 fclose($handle);
10 ?>
  
```



Gambar 7.2 Tampilan Program 7.2

Menulis ke File

```

fwrite ($handle, $isi); fputs
($handle, $isi);
  
```

Program 7.3

Nama File : file03.php
 Deskripsi : Program menulis ke file.

```

1 <?php
2 $namafile = "data.txt";
3 $handle = fopen ($namafile, "w");
  
```

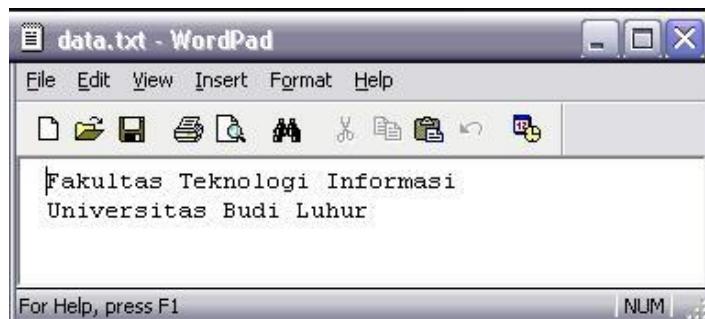
```

4 if (!$handle) {
5     echo "<b>File tidak dapat dibuka atau belum ada</b>";
6 } else {
7     fwrite ($handle, "Fakultas Teknologi Informasi\n");
8 fputs ($handle, "Universitas Budi Luhur\n"); 9 //file_put_contents
($namafile, "Jakarta");
10    echo "<b>File berhasil ditulis</b>";
11 }
12 fclose($handle);
13 ?>

```

Penjelasan Program 7.3

Untuk menulis ke file, dapat menggunakan fungsi **fwrite()** dan **fputs()**. Variabel \$handle merupakan mode pengaksesan file, dimana mode yang digunakan harus bisa menulis ke file. Contohnya mode w dan a. Selain itu, kita juga bisa menggunakan fungsi **file_puts_contents()** dengan parameter berupa nama file dan isi yang akan ditambahkan. Jika program di atas dijalankan, maka isi file data.txt akan berisi seperti pada gambar 7.3 (buka dengan WordPad).



Gambar 7.3. File data.txt dibuka dengan WordPad

Membaca Isi File

```

fread ($handle [, $panjang]); fgets
($handle [, $panjang]);

```

Program 7.4

Nama File : file04.php

Deskripsi : Program membaca dan menampilkan isi file.

```

1 <?php
2 $namafile = "data.txt";
3 $handle = fopen ($namafile, "r");
4 if (!$handle) {
5     echo "<b>File tidak dapat dibuka atau belum ada</b>";
6 } else {
7     $isi = fgets ($handle, 2048);
8     // $isi2 = fread ($handle, 20);
9     echo "Isi 1 : $isi<br>";
10    //echo "Isi 2 : $isi2<br>";
11 }
12 fclose($handle);
13 ?>

```

Penjelasan Program 7.4

Untuk membaca isi file, bisa menggunakan fungsi **fgets()** dan **fread()**. Fungsi **fgets()** akan membaca isi file tiap baris. Sedangkan fungsi **fread()** akan membaca isi file setiap jumlah byte tertentu. Pada program di atas, baris ke-8 akan membaca file setiap 20 byte atau 20 karakter.

Program 7.5

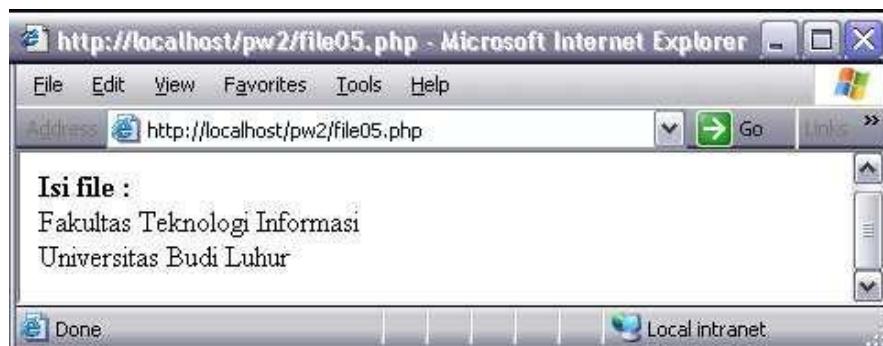
Nama File : file05.php

Deskripsi : Program membaca isi file baris tiap baris.

```
1 <?php
2 $namafile = "data.txt";
3 $handle = fopen ($namafile, "r");
4 if (!$handle) {
5     echo "<b>File tidak dapat dibuka atau belum ada</b>";
6 } else {
7     echo "<b>Isi file : </b><br>";
8     while ($isi = fgets ($handle, 2048)) {
9         echo "$isi<br>";
10    }
11 }
12 fclose($handle);
13 ?>
```

Penjelasan Program 7.5

Program 7.5 di atas akan menampilkan keseluruhan isi file, dimana isi file akan dibaca tiap baris. Pembacaan dilakukan dari baris pertama sampai baris terakhir.



Gambar 7.4. Tampilan Program 7.5

Program 7.6

Nama File : file06.php

Deskripsi : Program membaca isi file baris tiap baris.

```
1 <?php
2 $namafile = "data.txt";
3 $handle = @fopen($namafile, "r");
4 if ($handle) {
5     while (!feof($handle)) {
6         $buffer = fgets($handle, 4096);
7         echo $buffer."<br>";
8     }
9     fclose($handle);
10 }
11 ?>
```

Penjelasan Program 7.6

Program 7.6 di atas akan menampilkan keseluruhan isi file, dimana isi file akan dibaca tiap baris. Pembacaan dilakukan selama pointer file belum sampai di end-of-file. Fungsi feof() pada baris ke-5 berfungsi untuk memeriksa pointer file sudah berada di akhir file atau belum.

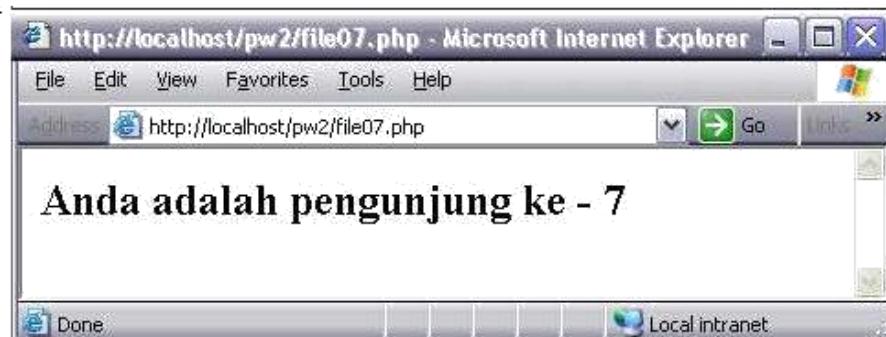
Menutup File

```
fclose ($handle);
```

Program 7.7

Nama File : file07.php
Deskripsi : Program aplikasi user counter dengan file.

```
1 <?php
2 $counter_file="counter.txt";
3 if (!file_exists ($counter_file)) { 4 fopen
($counter_file, "w");
5 }
6 $file = fopen($counter_file,"r");
7
8 $counter = fread($file,10);
9 fclose($file);
10
11 $counter++;
12
13 echo "<h2>Anda adalah pengunjung ke - $counter</h2>";
14 $file = fopen($counter_file, "w");
15 fwrite($file,$counter);
16 fclose($file);
17 ?>
```



Gambar 7.5. Tampilan Program 7.7

Membuat dan Menghapus Direktori

Membuat Direktori Baru

Program 7.8

Nama File : file08.php
Deskripsi : Program membuat direktori baru.

```
1 <?php
2 $dir    = "images"; //nama direktori
3 $cek   = mkdir ($dir);
4 if ($cek) {
5     echo "Direktori <b>$dir</b> berhasil dibuat";
6 } else {
7     echo "Direktori <b>$dir</b> gagal dibuat";
8 }
9 ?>
```

Penjelasan Program 7.8

Program 7.8 di atas akan membuat direktori bernama "images" di direktori tempat program 7.8 disimpan. Fungsi untuk membuat direktori baru dalam PHP adalah **mkdir()**. Jika direktori yang akan dibuat berada di luar tempat program disimpan, maka sertakan *path* direktori secara lengkap.

Menghapus Direktori

Program 7.9

Nama File : file09.php
Deskripsi : Program menghapus direktori.

```
1 <?php
2 //buat direktori
3 $dir = "include";           //nama direktori
4 $cek = mkdir ($dir);
5 if ($cek) {
6     echo "Direktori <b>$dir</b> berhasil dibuat";
7 } else {
8     echo "Direktori <b>$dir</b> gagal dibuat";
9 }
17 //hapus direktori
18 $del = rmdir ($dir);
19 if ($del) {
20     echo "<br>Direktori <b>$dir</b> berhasil dihapus";
21 } else {
22     echo "<br>Direktori <b>$dir</b> gagal dihapus";
23 }
24 ?>
```

Penjelasan Program 7.9

Program 7.9 di atas akan membuat direktori baru bernama "include" (lihat baris ke-4). Setelah itu, direktori akan dihapus dengan fungsi **rmdir()** pada baris ke-10. Fungsi rmdir() hanya bisa menghapus direktori kosong. Jika direktori ada isinya maka, gunakan fungsi rekursif untuk menghapus isi direktori. Lihat contoh di PHP Manual.

Menampilkan Isi Direktori

Program 7.10

Nama File : file10.php
Deskripsi : Program menampilkan file dan direktori dalam suatu direktori.

```
1 <?php
2 $dir = "images";
3 if ($handle = opendir($dir)) {
4     while (false !== ($file = readdir($handle))) {
5         if ($file != "." && $file != "..") {
6             echo "$file<br>";
7         }
8     }
9     closedir($handle);
10}
11?>
```

Penjelasan Program 7.10

Program 7.10 di atas akan menampilkan di layar, semua file dan direktori yang berada dalam direktori "images". Fungsi **opendir()** pada baris ke-3 adalah untuk membuka direktori. Sedangkan fungsi **readdir()** akan membaca file dalam direktori satu per satu secara urut. Fungsi **closedir()** pada baris ke-9 adalah untuk menutup pembukaan direktori.

Manipulasi File

Mendapatkan Informasi File

Beberapa Fungsi yang berhubungan dengan informasi dan manipulasi file, terdapat dalam tabel sebagai berikut :

Fungsi	Keterangan
file_exists(\$file)	Memeriksa apakah \$file ada atau tidak
is_file(\$file)	Memeriksa apakah \$file adalah file atau bukan.
is_dir(\$file)	Memeriksa apakah \$file adalah direktori atau bukan.
is_executable(\$file)	Memeriksa apakah \$file termasuk file yang bisa dijalankan secara langsung atau bukan.
is_writable(\$file)	Memeriksa apakah \$file termasuk file yang bisa ditulis atau diedit.
is_readable(\$file)	Memeriksa apakah \$file termasuk file yang bisa dibaca atau tidak.
fileatime(\$file)	Menghasilkan waktu akses terakhir file (unix timestamp).
filectime(\$file)	Menghasilkan waktu pembuatan file (unix timestamp).
filemtime(\$file)	Menghasilkan waktu modifikasi terakhir file (dalam unix timestamp).
filesize(\$file)	Menghasilkan ukuran besar file (dalam byte).
filetype(\$file)	Menghasilkan jenis file.

Program 7.11

Nama File : file11.php

Deskripsi : Program memeriksa keberadaan suatu file.

```
1 <?php
2 $filename = "data.txt";
3
4 if (file_exists($filename)) {
5     echo "File $filename ADA";
6 } else {
7     echo "File $filename TIDAK ADA";
8 }
9 ?>
```

Program 7.12

Nama File : file12.php

Deskripsi : Program mendapatkan beberapa informasi file.

```
1 <?php
2 $file = "data.txt";
3
4 if (is_file($file)) {
5     echo "File <b>$file</b> adalah FILE<br>";
6     //
7     if (is_executable ($file)) {
8         echo "File <b>$file</b> bisa dijalankan secara
langsung (executable)<br>";
9     } else {
10        echo "File <b>$file</b> TIDAK executable<br>";
11    }
12    //
13    if (is_writable ($file)) {
14        echo "File <b>$file</b> bisa ditulis / diedit<br>";
15    } else {
16        echo "File <b>$file</b> TIDAK bisa ditulis /
diedit<br>";
17    }
}
```

```

18 // 
19 if (is_readable ($file)) {
20     echo "File <b>$file</b> bisa dibaca<br>";
21 } else {
22     echo "File <b>$file</b> TIDAK bisa dibaca<br>";
23 }
24 //
25 echo "Akses terakhir file <b>$file</b> = ". date("d-m-Y H:i:s.", fileatime($file)). "<br>";
26 echo "File <b>$file</b> dibuat = ". date("d-m-Y H:i:s.", filectime($file)). "<br>";
27 echo "Modifikasi terakhir file <b>$file</b> = ". date("d-m-Y H:i:s.", filemtime($file)).
28 "<br>";
29 echo "Ukuran file <b>$file</b> = ". filesize($file). " bytes<br>";
30 echo "Jenis file <b>$file</b> = ". filetype($file). "<br>";
31 } else if (is_dir($file)){
32     echo "File <b>$file</b> adalah DIREKTORI<br>";
33 } else {
34     echo "File <b>$file</b> tidak dikenal<br>";
35 }
36 ?>

```

Copy, Rename dan Delete File

Program 7.13

Nama File : file13.php

Deskripsi : Program meng-copy, rename dan delete file.

```

1 <?php
2 $file = "data.txt";
3 $filebaru = "databaru.txt";
4 $filebaru2 = "databaru2.txt";
5 if (copy ($file, $filebaru)) {
6 echo "File <b>$file</b> berhasil dicopy menjadi <b>$filebaru</b>. <br>";
7 }
8 if (rename ($filebaru, $filebaru2)) {
9 echo "File <b>$filebaru</b> berhasil direname menjadi <b>$filebaru2</b>. <br>";
10 }
11 if (unlink ($filebaru2)) {
12     echo "File <b>$filebaru2</b> berhasil dihapus. <br>";
13 }
14 ?>

```

Penjelasan Program 7.13

Program 7.13 di atas terdapat beberapa fungsi untuk meng-copy, rename dan delete file. Fungsi **copy()** pada baris ke-5 akan menduplikasi file \$file menjadi file \$filebaru. Selanjutnya fungsi **rename()** akan mengganti nama file \$filebaru menjadi \$filebaru2. Fungsi **unlink()** pada baris ke-11 akan menghapus file \$filebaru2.

Upload File

PHP, sebagai server-side-scripting, sangat memungkinkan untuk menangani upload file ke server. Ada beberapa hal yang perlu diperhatikan dalam upload file ini, yaitu :

1. Pada Form HTML harus ditambahkan atribut :
ENCTYPE="multipart/form-data"

-
2. Form inputan upload file dapat menggunakan tag <input> dengan value atribut TYPE="FILE".
 3. Untuk menangani inputan, PHP menyediakan sebuah variabel array global yaitu `$_FILES`. Index dari variabel ini antara lain :
 - `$_FILES['file']['name']` : Nama asli dari file yang diupload
 - `$_FILES['file']['tmp_name']` : Nama temporary file yang diupload
 - `$_FILES['file']['size']` : Ukuran file asli (dalam byte)
 - `$_FILES['file']['type']` : MIME type file yang diupload
 4. *Destination folder* file upload harus bisa writable (bisa diakses), biasanya dengan permission 777 atau 775.

Program 7.14

Nama File : `form_upload.php`
Deskripsi : Program menampilkan form upload file.

```

1 <html>
2   <head><title>Upload File</title></head>
3   <body>
4     <FORM ACTION="upload.php" METHOD="POST"
ENCTYPE="multipart/form-data">
5       Upload File : <input type="file" name="file"><br>
6       <input type="submit" name="Upload" value="Upload">
7     </FORM>
8   </body>
9 </html>
```

Program 7.15

Nama File : `upload.php`
Deskripsi : Program proses upload file.

```

1 <?php
2 if (isset($_POST['Upload'])) {
3   $dir_upload = "images/";
4   $nama_file = $_FILES['file']['name'];
5   //
6   if (is_uploaded_file($_FILES['file']['tmp_name'])) {
7     $cek = move_uploaded_file($_FILES['file']['tmp_name'], $dir_upload.$nama_file);
8
9     if ($cek) {
10       die ("File berhasil diupload");
11     } else {
12       die ("File gagal diupload");
13     }
14   }
15 ?>
```

Penjelasan Program 7.15

Program 7.15 di atas merupakan program sederhana penanganan upload file. Fungsi `is_uploaded_file()` pada baris ke-6 akan mengupload file yang telah dipilih melalui form pada program 7.14 ke folder temporary (sementara). Selanjutnya pada baris ke-7, file yang sudah diupload ke folder temporary akan dipindahkan (move) ke direktori yang diinginkan menggunakan fungsi `move_uploaded_file()`. Lihat baris ke-7!



Gambar 7.6. Tampilan Program 7.14

Bab 9

Session dan Cookies

TM	Session
TM	Cookies

Session

Session merupakan hal yang cukup penting dalam aplikasi berbasis web. Dengan session memungkinkan programmer menyimpan informasi user secara

semi-permanen, artinya selama masa tertentu informasi akan tersimpan. Penyimpanan isi variabel session berada di server, jadi tidak bisa diakses secara langsung oleh client.

Dalam aplikasi berbasis web, session banyak digunakan sebagai autentifikasi login. Dengan session memungkinkan programmer mengatur siapa saja yang bisa mengakses suatu halaman. Misalnya saja, untuk melihat halaman kotak surat pada email, kita harus login terlebih dahulu. Dalam proses login antara lain akan terjadi pembuatan suatu session yang akan dibawa oleh user di setiap halaman. Di halaman kotak surat, session tersebut akan diperiksa. Jika session benar maka user dipersilahkan membuka halaman kotak surat, namun jika salah maka user tidak bisa membuka halaman kotak surat dan biasanya akan diminta untuk login terlebih dahulu. Itulah sebabnya, user tidak bisa mengakses halaman kotak surat secara langsung tanpa melakukan login.

Dalam penanganan session terdapat beberapa proses yang perlu diperhatikan :

- f Proses pembuatan session
- f Proses pemeriksaan session
- f Proses penghapusan session

Selanjutnya bagaimana session itu sendiri dijalankan? Agar proses penyimpanan dalam session berjalan, PHP melakukan beberapa hal berikut ini :

1. PHP meng-generate (membentuk) sebuah ID session.

ID session ini merupakan sejumlah deret angka random yang unik untuk setiap user dan hampir tidak mungkin bisa ditebak. ID session disimpan oleh PHP di dalam variabel sistem PHP dengan nama PHPSESSID

2. PHP menyimpan nilai yang akan Anda simpan dalam session di dalam file yang berada di server. Nama file tempat penyimpanan session tersebut sesuai (sama) dengan ID session. File disimpan dalam suatu direktori yang ditunjukkan oleh session.save_path dalam file php.ini.

3. PHP melempar ID session ke setiap halaman.

4. PHP mengambil nilai session dari file session untuk setiap halaman session.

Program 9.1

Nama File : session01.php

Deskripsi : Program menciptakan session.

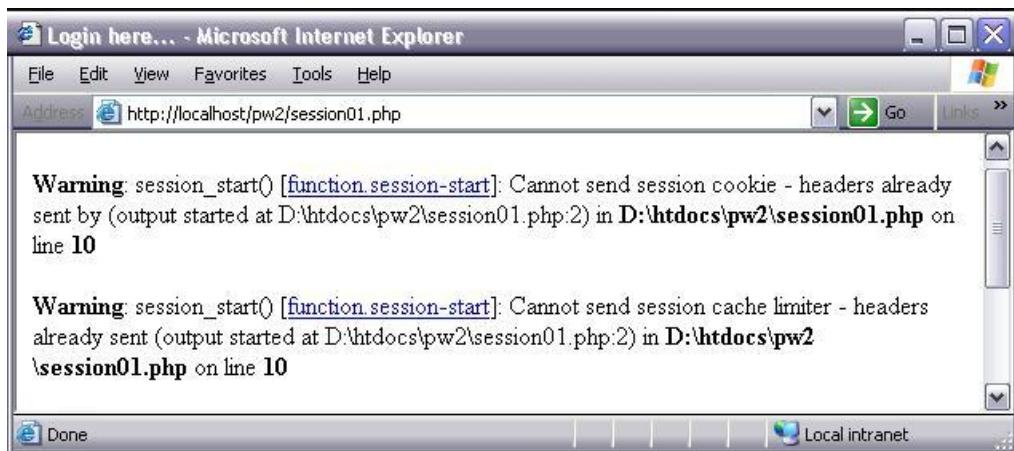
```
1 <?php
2 ****
3 Nama file : session01.php
4 Halaman ini merupakan halaman contoh penciptaan session.
5 Perintah session_start() harus ditaruh di perintah pertama
6 tanpa spasi di depannya. Perintah session_start() harus ada
7 pada setiap halaman yang berhubungan dengan session
8 ****
9 session_start();
10 if (isset ($_POST['Login'])) {
11     $user = $_POST['user'];
```

```

12     $pass = $_POST['pass'];
13 //periksa login
14 if ($user == "achmatim" && $pass = "123") {
15     //menciptakan session
16     $_SESSION['login'] = $user;
17     //menuju ke halaman pemeriksaan session
18     echo "<h1>Anda berhasil LOGIN</h1>";
19 echo "<h2>Klik <a href='session02.php'>di sini (session02.php)</a>
20     untuk menuju ke halaman pemeriksaan session";
21 }
22 } else {
23 ?>
24 <html>
25     <head>
26         <title>Login here...</title>
27     </head>
28     <body>
29         <form action="" method="post">
30             <h2>Login Here...</h2>
31             Username : <input type="text" name="user"><br>
32             Password : <input type="password" name="pass"><br>
33             <input type="submit" name="Login" value="Log In">
34         </form>
35     </body>
36 </html>
37 <? } ?>
```

Penjelasan Program 9.1.

Pada program di atas terdapat fungsi **session_start()** yang berada pada baris ke-9. Fungsi **session_start()** berfungsi untuk memulai sebuah session. Fungsi harus dipanggil pertama kali dalam suatu halaman PHP, artinya fungsi **session_start()** harus dipanggil sebelum sesuatu tampil di layar, walaupun hal itu hanya berupa spasi. Jika sebelum **session_start()** terdapat output yang ditampilkan maka akan terjadi error seperti pada gambar 9.1 berikut ini :



Gambar 9.1. Error yang terjadi karena sebelum **session_start()** sudah ada output

Selanjutnya, program 9.1 akan menampilkan sebuah form login sederhana di layar. Form terdiri dari form inputan username, password dan sebuah tombol login. Jika tombol login ditekan maka pemeriksaan kondisi pada baris 10 akan bernilai TRUE dan perintah yang ada di blok **if** akan dieksekusi. Baris 11 dan 12 merupakan perintah untuk mengambil nilai pada form inputan username dan password. Nilai username dan password tersebut diperiksa pada baris 14. Pemeriksaan username dan password ini, untuk aplikasi web yang sudah

menggunakan database, bisa diganti dengan pemeriksaan username dan password ke tabel database.

Pada baris 16, terdapat perintah sebagai berikut :

```
$SESSION['login'] = $user;
```

Baris perintah tersebut merupakan perintah untuk membuat session baru (*create session*) dimana nama session adalah "login" dan isi dari session adalah \$user. `$_SESSION` merupakan sebuah variabel array global yang didefinisikan oleh PHP, jadi variabel ini harus ditulis dengan huruf kapital (*upper-case*).

Selanjutnya, pada baris 18-20 akan ditampilkan pesan bahwa login berhasil dan juga sebuah link menuju ke halaman pemeriksaan session (session02.php, program 9.2).

Program 9.2

Nama File : session02.php
Deskripsi : Program pemeriksaan session.

```
1 <?php
2 ****
3 Halaman ini merupakan contoh halaman pemeriksaan session. Pemeriksaan session biasanya dilakukan jika suatu halaman memiliki akses terbatas. Misalnya harus login terlebih dahulu.
4 ****
5 session_start();
6
7 //pemeriksaan session
8 if (isset($_SESSION['login'])) { //jika sudah login
9     //menampilkan isi session
10    echo "<h1>Selamat Datang ". $_SESSION['login'] . "</h1>";
11    echo "<h2>Halaman ini hanya bisa diakses jika Anda sudah login</h2>";
12
13 echo "<h2>Klik <a href='session03.php'>di sini (session03.php)</a>
untuk LOGOUT</h2>";
14 } else {
15     //session belum ada artinya belum login
16     die ("Anda belum login! Anda tidak berhak masuk ke halaman ini.Silahkan login <a
17 href='session01.php'>di sini</a>");
18 }
19 ?>
```

Penjelasan Program 9.2

Program 9.2 di atas diawali dengan pemanggilan fungsi `session_start()`. Hal ini berarti dalam halaman program ini terdapat operasi session. Pada baris 10 program di atas, terdapat pemeriksaan keberadaan variabel `$_SESSION['login']` dengan menggunakan fungsi `isset()`. Pemeriksaan kondisi ini akan bernilai TRUE jika `$_SESSION['login']` sudah terbentuk. Jika bernilai TRUE maka blok program baris 11-14 yang akan dieksekusi, dan sebaliknya, jika FALSE maka blok program baris 16-17 yang akan dijalankan. Blok program baris 11-14 hanya bisa diakses

saat user sudah login dengan benar. Dengan kata lain, bagian tersebut merupakan bagian yang bersifat private (terbatas).

Untuk menampilkan isi variabel session dapat dilakukan seperti pada baris 12 program di atas. Cukup dengan meng-echo variabel `$_SESSION['login']`.

Jika user belum login (tidak melewati halaman login) atau langsung mengakses halaman ini, maka akan ditampilkan pesan bahwa user tidak berhak masuk ke halaman ini. Fungsi `die()` pada baris 17 digunakan untuk menampilkan suatu pesan sekaligus men-*terminate* program, artinya perintah selain `die()` akan diabaikan.

Program 9.3

Nama File : session03.php
 Deskripsi : Program menghapus session.

```

6. <?php
7. ****
8. Halaman ini merupakan halaman logout, dimana kita menghapus
9. session yang ada.
10. ****
11. session_start();
12. if (isset($_SESSION['login'])) {
13.     unset ($_SESSION);
14.     session_destroy();
15. }
16. ?>

```

Penjelasan Program 9.3.

Program 9.3 merupakan contoh program penghapusan session. Penghapusan session biasanya digunakan ketika user melakukan logout. Proses penghapusan session dilakukan dengan fungsi **unset()** dan fungsi **session_destroy()**. Fungsi **unset()** digunakan untuk menghapus atau menghilangkan suatu variabel. Sedangkan **session_destroy()** digunakan untuk menghapus semua data yang berhubungan dengan session tersebut.

Cookies

Seperti halnya session, cookies juga merupakan sebuah konsep penyimpanan informasi user. Hanya saja, jika session tempat penyimpanan berada di server, cookies berada di client. Oleh karena itu, konsep cookies sebaiknya jangan digunakan untuk menyimpan informasi login user seperti username, password dsb. Selain user bisa melihat informasi yang disimpan, user juga bisa men-disable cookies itu sendiri. Jika cookies di-disable, maka program yang memanfaatkan cookies tentunya tidak akan berjalan dengan baik.

Cookies sendiri biasanya dipakai dalam aplikasi shopping cart. Biasa digunakan untuk menyimpan sementara, produk-produk yang dipilih oleh pengunjung pada saat berbelanja.

Dalam penanganan cookies juga terdapat beberapa proses yang perlu diperhatikan :

- f Proses pembuatan cookies
- f Proses pemeriksaan cookies
- f Proses penghapusan cookies

Program 9.4

Nama File : cookie01.php
 Deskripsi : Program membuat cookie.

```

1. <?php
2. $value = 'achmatim';
3. $value2 = 'Achmad Solichin';
4.
5. setcookie("username", $value);
6. setcookie("namalengkap", $value2, time() + 3600); /* expire in 1
hour */
7.
8. echo "<h1>Ini halaman pengesetan cookie</h1>";
9.

```

```
10 echo "<h2>Klik <a href='cookie02.php'>di sini</a> untuk pemeriksaan  
11 cookies</h2>";  
12 ?>
```

Program 9.5

Nama File : cookie02.php
Deskripsi : Program pemeriksaan cookie.

```
1 <?php  
2  
3 if(isset($_COOKIE['username'])) {  
4     echo "<h1>Cookie 'username' ada. Isinya : " .  
5 $_COOKIE['username'];  
6 } else {  
7     echo "<h1>Cookie 'username' TIDAK ada.</h1>";  
8 }  
9 if(isset($_COOKIE['namalengkap'])) {  
10    echo "<h1>Cookie 'namalengkap' ada. Isinya : " .  
11 $_COOKIE['namalengkap'];  
12 } else {  
13    echo "<h1>Cookie 'namalengkap' TIDAK ada.</h1>";  
14 }  
15 echo "<h2>Klik <a href='cookie01.php'>di sini</a> untuk penciptaan  
cookies</h2>";  
16 echo "<h2>Klik <a href='cookie03.php'>di sini</a> untuk penghapusan  
cookies</h2>";  
17 ?>
```

Program 9.6

Nama File : cookie03.php
Deskripsi : Program penghapusan cookie.

```
1 <?php  
2 // set the expiration date to one hour ago  
3 setcookie ("username", "", time() - 3600);  
4 setcookie ("namalengkap", "", time() - 3600);  
5 echo "<h1>Cookie Berhasil dihapus.</h1>";  
6 echo "<h2>Klik <a href='cookie01.php'>di sini</a> untuk  
7 penciptaan cookies</h2>";  
8 echo "<h2>Klik <a href='cookie02.php'>di sini</a> untuk  
9 pemeriksaan cookies</h2>";  
10 ?>
```

Bab 10

Pemrograman Berorientasi Objek dalam PHP

- | | |
|----|--|
| TM | Pengantar Pemrograman Berorientasi Objek |
| TM | Object dan Class |
| TM | Properties dan Method |
| TM | Mendefinisikan Class |

Pengantar Pemrograman Berorientasi Objek

PHP pada awalnya hanyalah kumpulan script sederhana. Dalam perkembangannya, selanjutnya ditambahkan berbagai fitur pemrograman berorientasi objek. Hal ini dimulai sejak PHP 4. Dengan lahirnya PHP 5, fitur-fitur pemrograman berorientasi objek semakin mantap dan semakin cepat. Dengan PHP 5, script yang menggunakan konsep *object-oriented* akan lebih cepat dan lebih efisien.

Pemrograman berorientasi objek atau *object-oriented programming* (OOP) merupakan suatu pendekatan pemrograman yang menggunakan object dan class. Saat ini konsep OOP sudah semakin berkembang. Hampir setiap perguruan tinggi di dunia mengajarkan konsep OOP ini pada mahasiswanya. Pemrograman yang banyak dipakai dalam penerapan konsep OOP adalah Java dan C++.

OOP bukanlah sekedar cara penulisan sintaks program yang berbeda, namun lebih dari itu, OOP merupakan cara pandang dalam menganalisa sistem dan permasalahan pemrograman. Dalam OOP, setiap bagian dari program adalah *object*. Sebuah *object* mewakili suatu bagian program yang akan diselesaikan.

Beberapa konsep OOP dasar, antara lain :

1. *Encapsulation* (Class dan Object)
2. *Inheritance* (Penurunan sifat), dan
3. *Polymorphism*

PHP khususnya PHP 5 sudah mendukung beberapa konsep OOP. Akan tetapi PHP 5 tidak mendukung konsep *Multiple-inheritance* dan *polymorphism*.

Object dan Class

Bagian dasar dari sebuah program yang berorientasi objek adalah **objects**.

Secara mudah kita dapat memahami mengenai *object* ini. Sebagai contoh, sebuah **mobil** adalah objek. Sebuah mobil mempunyai *properties* atau bagian-bagian di dalamnya, seperti warna, mesin, roda, pintu dsb. Sebuah mobil juga dapat melakukan sesuatu (ada sesuatu yang bisa dilakukan dengan mobil), seperti mengisi bensin, menyalaikan mesin, berjalan, mengerem dsb.

Biasanya *object* adalah sebuah kata benda. **Orang** adalah *object*. Demikian juga mobil, pohon, bunga, komputer, TV, buku dsb. Namun, *object* tidak selamanya sebuah objek fisik. Bisa saja sebuah benda abstrak, seperti *account* bank, sebuah file di komputer, database, pesan email, acara TV, dsb.

Class merupakan penjelasan atau deskripsi dari *object*. Di dalam class, terdapat penjelasan tentang suatu *object* termasuk *properties* yang dimilikinya serta kelakuan atau *method* yang bisa dilakukan oleh *object*. Sebagai contoh, **class Orang**. Class Orang tentu setidaknya memiliki beberapa bagian seperti tangan, kaki, mata, telinga dsb. Class Orang juga setidaknya harus bisa jalan, bisa loncat, bisa lari, bisa melihat, bisa bicara dsb.

Salah satu keuntungan program didefinisikan dengan konsep OOP adalah adanya pengkapsulan (*encapsulation*) program dalam *class* dan *object*, dimana programmer yang menggunakan *class* tidak perlu mengetahui isi dan jalannya *class* secara detail, hanya perlu tahu bagaimana cara menggunakannya. Sama halnya dengan sebuah **mobil** misalnya. Seorang pemilik mobil tentunya tidak

perlu mengetahui bagian-bagian mobil secara menyeluruh. Dia tidak perlu mengetahui bagaimana mesin mobil melakukan pembakaran dan bagaimana mesin mobil bisa menggerakkan roda, dsb. Dia hanya perlu tahu bagaimana cara menjalankan mobil, bagaimana menghentikan mobil, dan fungsi mobil lainnya.

Properties dan Method

Setiap class memiliki *properties* yang kadang disebut juga *attributes*. *Properties* dari sebuah mobil misalnya warna, ukuran, harga dsb. Di dalam class, *properties* dinyatakan dengan sebuah variabel. Misalnya \$warna, \$harga, dsb.

Method merupakan sesuatu yang bisa dilakukan oleh *object*. Method dalam PHP sama artinya dengan sebuah fungsi. Method yang mungkin dippunyai dari sebuah mobil misalnya, method untuk menghidupkan mobil, menjalankan mobil, menghentikan mobil, dsb.

Penamaan *properties* dan *method* memiliki aturan yang sama dengan penamaan sebuah variabel atau fungsi. Akan tetapi berdasarkan kesepakatan

(*convention*), penamaan *properties* dan *method* harus menggunakan *camel Caps*, dimana tiap kata diawali dengan huruf besar kecuali kata pertama, setiap kata digabung tanpa spasi atau *under-score* (_).

Mendefinisikan Class

Bentuk umum mendefinisikan sebuah *class* adalah sbb :

```
class namaClass
{
    Deklarasikan dan definisikan properties di sini Definisikan
    semua method di sini
}
```

Penamaan **namaClass** pada dasarnya sama dengan penamaan variabel. Penamaan bebas, boleh apa saja, kecuali **stdClass**. PHP sudah menggunakan nama **stdClass** sebagai nama *class built-in*. Isi tubuh class terletak di antara tanda kurung kurawal buka ({) dan kurawal tutup (}). Di tubuh class terdapat pendefinisian properties (variabel) dan method-method class.

Menambahkan Properties (Variabel)

Perhatikan contoh pendefinisian class beserta propertiesnya, sbb :

```
1 class Mobil
2 {
3     var $warna;
4     var $merk;
5     var $harga;
6
7     //Tambahkan pendefinisian method di sini
8 }
```

Pendeklarasian variabel dalam class seperti pada contoh di atas, bukanlah suatu keharusan, karena dalam PHP variabel tidak perlu dideklarasikan, cukup digunakan saja. Namun demikian, pendeklarasian variabel yang digunakan sangat dianjurkan untuk kemudahan pembacaan dan pemahaman program.

Variabel class juga dapat langsung diinisialisasi dengan sebuah nilai. Namun inisialisasi variabel tidak boleh mengandung operasi aritmatika maupun operasi yang lainnya. Perhatikan contoh sebagai berikut :

```
1 class Mobil
2 {
3     var $warna = "Biru";
4     var $merk = "BMW";
5     var $harga = "10000000";
6 }
```

7 //Tambahkan pendefinisian method di sini }8

Menambahkan Method

Untuk menambahkan method, tinggal mendefinisikan method seperti halnya fungsi biasa. Misalnya kita akan menambahkan fungsi atau method untuk mengganti warna mobil dan untuk menampilkan warna mobil. Nama method pada dasarnya terserah (mengikuti aturan nama variabel). Namun demikian, jangan menggunakan nama method yang diawali dengan dua buah *under-score*.

(_) yaitu **__construct()**, **__destruct()** dan **__clone()** karena ketiga fungsi tersebut mempunyai arti tersendiri di PHP.

Lihat contoh menambahkan method sbb :

```
1 class Mobil
2 {
3     var $warna = "Biru";
4     var $merk = "BMW";
5     var $harga = "10000000";
6
7     function gantiWarna ($warnaBaru)
8     {
9         $this->warna = $warnaBaru;
10    }
11
12    function tampilWarna ()
13    {
14        echo "Warna mobilnya : " . $this->warna;
15    }
16 }
```

Pada listing program di atas sudah ditambahkan fungsi atau *properties* method **gantiWarna()** dan **tampilWarna()**. Untuk mengakses *me-refer* (variabel) bisa menggunakan keyword **\$this**. Keyword ini pada *satu* *class* tempatnya berada. Method **gantiWarna()** mempunyai parameter, yaitu **\$warnaBaru**.

Menambahkan Konstruktur

Konstruktur merupakan sebuah *method* khusus yang akan secara otomatis dijalankan saat *object* terbentuk. Konstruktur tidak harus ada, namun dalam satu *class* hanya boleh ada satu konstruktur. Method konstruktur biasanya berisi pemberian nilai *default* dari masing-masing *properties* (variabel).

Untuk membuat konstruktur, cukup dengan mendefinisikan suatu fungsi dengan nama **__construct()**. Perhatikan contoh sebagai berikut :

```
1 class Mobil
2 {
3     var $warna;
4     var $merk;
5     var $harga;
6
7     function __construct()
8     {
9         $this->warna = "Biru";
10        $this->merk = "BMW";
11        $this->harga = "10000000";
12    }
13
14    function gantiWarna ($warnaBaru)
15    {
16        $this->warna = $warnaBaru;
17    }
18
19    function tampilWarna ()
20    {
21        echo "Warna mobilnya : " . $this->warna;
```

```
22     }
23 }
```

Membentuk Objek Class

Untuk menggunakan sebuah objek, harus dibentuk objek dari classnya. Dari sebuah class bisa dibentuk beberapa objek sekaligus. Bentuk umum pembentukan objek adalah sbb :

```
$namaObjek = new namaClass();
```

Sedangkan untuk memanggil anggota (member) dari class dapat dengan format sbb :

```
$namaObjek->variabel;
$namaObjek->namaMethod();
```

Program 10.1

Nama File : pbo01.php

Deskripsi : Program sederhana pendefinisian class dan pemanggilan class.

```
1 <?
2 class Mobil
3 {
4     var $warna;
5     var $merk;
6     var $harga;
7
8     function __construct()
9     {
10        $this->warna = "Biru";
11        $this->merk = "BMW";
12        $this->harga = "10000000";
13    }
14    function gantiWarna ($warnaBaru)
15    {
16        $this->warna = $warnaBaru;
17    }
18
19    function tampilWarna ()
20    {
21        echo "Warna mobilnya : " . $this->warna;
22    }
23 }
24
25 $a = new Mobil();
26 $b = new Mobil();
27 echo "<b>Mobil pertama</b><br>";
28 $a->tampilWarna();
29 echo "<br>Mobil pertama ganti warna<br>";
30 $a->gantiWarna("Merah");
31 $a->tampilWarna();
32 //
33 echo "<br><b>Mobil kedua</b><br>";
34 $b->gantiWarna("Hijau");
35 $b->tampilWarna();
36 ?>
```

Program 10.2

Nama File : pbo02.inc.php

Deskripsi : Program class untuk membuat sebuah form inputan sederhana.

```
1 <?php
2 /*
3 Nama Class : Form
```

```

4 Deskripsi : CLass untuk membuat form inputan text sederhana
5 */
6
7 class Form
8 {
9     var $fields = array();
10    var $action;
11    var $submit = "Submit Form";
12    var $jumField = 0;
13
14    function __construct($action, $submit)
15    {
16        $this->action = $action;
17        $this->submit = $submit;
18    }
19    function displayForm()
20    {
21        echo "<form action=\".$this->action.\" method='POST'>";
22        echo "<table width='100%'>";
23        for ($j=0; $j<count($this->fields); $j++) {
24            echo "<tr><td align='right'>".$this->
25 fields[$j]['label']."</td>";
26            echo "<td><input type='text' name=\"".$this-> fields[$j]['name']."'></td></tr>";
27        }
28        echo "<tr><td colspan='2'>";
29        echo "<input type='submit' value=\"".$this->submit. "></td></tr>";
30
31        echo "</table>";
32    }
33    function addField($name, $label)
34    {
35        $this->fields [$this->jumField]['name'] = $name;
36        $this->fields [$this->jumField]['label'] = $label;
37        $this->jumField++;
38    }
39
40 ?>
```

Program 10.3

Nama File : pbo03.php

Deskripsi : Program memanfaatkan Program 10.2 untuk membuat form inputan sederhana.

```

1 <?php
2 include "pbo02.inc.php";
3 echo "<html><head><title>Mahasiswa</title></head><body>";
4 $form = new Form ("", "Input Form");
5 $form->addField ("txtnim", "Nim");
6 $form->addField ("txtnama", "Nama");
7 $form->addField ("txtalamat", "Alamat");
8 echo "<h3>Silahkan isi form berikut ini :</h3>";
9 $form->displayForm();
10 echo "</body></html>";
11 ?>
```

Penjelasan Program 10.3

Program 10.3 di atas merupakan contoh pemanggilan class yang didefinisikan pada Program 10.2 (pbo02.inc.php). Untuk itu pertama kali program **pbo02.inc.php** harus di-include-kan dengan menggunakan fungsi **include()**. Perhatikan program baris ke-2. Untuk menambahkan form inputan text box, hanya perlu dengan memanggil fungsi **addField()**. Hasil tampilan dari program di atas dapat dilihat pada gambar 10.1



Gambar 10.1. Tampilan Program 10.3

Memanfaatkan Class yang Tersedia

Dengan adanya konsep pemrograman berorientasi objek, pada programmer PHP di seluruh dunia dapat lebih mudah berbagi teknik programming. Kita bisa membuat suatu class dan programmer lain dapat dengan mudah menggunakannya tanpa perlu mengetahui proses jalannya class tersebut. Just use it! Di internet saat ini bertebaran begitu banyak class dengan fungsinya masing-masing yang tersedia secara cuma-cuma dan dapat dimodifikasi secara bebas. Situs yang berisi kumpulan-kumpulan class PHP yang cukup terkenal antara lain <http://www.phpclasses.org>. Silahkan berkunjung dan ambil class yang Anda inginkan. Selain itu, Anda juga bisa mendapatkan banyak class PHP di kumpulan repository PHP dengan nama PECL (*PHP Extention Community Library*). PECL dapat diakses di <http://pecl.php.net>.

Berikut ini beberapa aplikasi PHP yang dibuat dengan konsep OOP yang banyak digunakan dalam aplikasi berbasis web :

1. PEAR

PEAR merupakan singkatan dari PHP Extension and Application Repository. PEAR berisi kumpulan class PHP yang dapat digunakan sewaktu-waktu. Class yang cukup banyak digunakan dari PEAR adalah **PEAR::DB** yang merupakan *abstraction layer* antara aplikasi PHP dengan database. Dengan PEAR::DB program menjadi lebih fleksibel dan secara mudah dapat berganti jenis database apapun (MySQL, PosgreSQL, Oracle dsb). PEAR dapat di-download dan dilihat di situs <http://pear.php.net>

2. Smarty

Smarty merupakan sebuah *template-engine* untuk PHP. Dengan Smarty memungkinkan terjadi pemisahan antara logika program (PHP) dan tampilan halaman web. Jadi programmer dan *web-designer* dapat mengerjakan sebuah *project* aplikasi secara paralel (bersamaan). Smarty merupakan salah satu *template-engine* yang banyak digunakan saat ini. Selain memisahkan antara logika program dan layout program, smarty juga dapat mempercepat akses ke halaman web karena smarty juga menggunakan konsep *cache* halaman web. Beberapa situs yang menggunakan Smarty antara lain www.bl.ac.id, www.blogsome.com, www.tikiwiki.org dan sebagainya. Smarty dapat di-download di situs <http://smarty.php.net>

3. FPDF

FPDF merupakan class PHP yang memungkinkan untuk membuat (generate) file PDF dengan PHP murni, tanpa menggunakan *library* PDFlib. F dari **FPDF** merupakan singkatan dari **Free**: jadi Anda bisa mendapatkannya secara

cuma-cuma dan juga dapat memodifikasinya. Informasi mengenai FPDF bisa didapat di situsnya <http://www.fpdf.org>

4. JpGraph

JpGraph merupakan library PHP untuk pembuatan grafik, baik dalam bentuk grafik batang, pie, chart, line dsb. JpGraph membutuhkan *extension* PHP berupa PHP GD. JpGraph dapat didownload secara gratis di <http://www.aditus.nu/jgraph/>

Bab 11

Pengenalan DBMS dan MySQL

- TM Tentang Database, DBMS, dan RDBMS
- TM Tentang MySQL
- TM Instalasi MySQL di Windows
- TM Koneksi ke Server MySQL dengan MySQL Client
- TM Berbagai Tools Administrasi Server MySQL

Tentang Database, DBMS dan RDBMS

Basis data (atau **database**) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut(<http://id.wikipedia.org/wiki/Database>). Database digunakan untuk menyimpan informasi atau data yang terintegrasi dengan baik di dalam komputer.

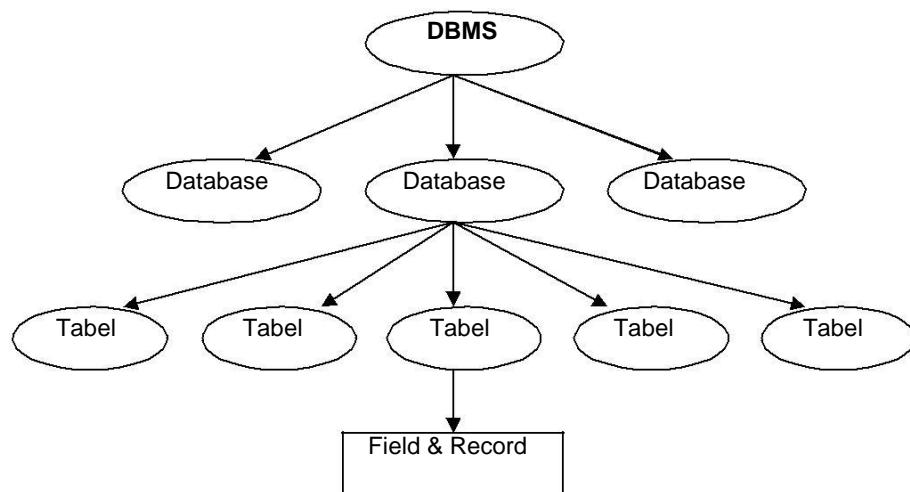
Untuk mengelola *database* diperlukan suatu perangkat lunak yang disebut **DBMS** (*Database Management System*). DBMS merupakan suatu perangkat lunak yang memungkinkan user (pengguna) untuk membuat, memelihara, mengontrol, dan mengakses database secara praktis dan efisien. Dengan DBMS, user akan lebih mudah mengontrol dan memanipulasi data yang ada.

Sedangkan **RDBMS** atau *Relationship Database Management System* merupakan salah satu jenis DBMS yang mendukung adanya *relationship* atau hubungan antar tabel. Di samping RDBMS, terdapat jenis DBMS lain, misalnya *Hierarchy DBMS*, *Object Oriented DBMS*, dsb.

Beberapa software atau perangkat lunak DBMS yang sering digunakan dalam aplikasi program antara lain :

- f DB2 - <http://www-306.ibm.com/software/data/db2/>
- f Microsoft SQL Server - <http://www.microsoft.com/sql/>
- f Oracle - <http://www.oracle.com>
- f Sybase - <http://www.sybase.com>
- f Interbase - <http://www.borland.com/interbase>
- f Teradata - <http://www.teradata.com/>
- f Firebird - <http://www.firebirdsql.org/>
- f MySQL – <http://www.mysql.com>
- f PostgreSQL - <http://www.postgresql.org/>

Dalam konsep database, urutan atau hierarki database sangatlah penting. Urutan atau hierarki database digambarkan dalam gambar sbb :



Gambar 11.1. Hierarki Database

Tentang MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak seperti Apache yang merupakan software yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu **MySQL AB**. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty"

Widenius.

Beberapa kelebihan MySQL antara lain :

- f Free (bebas didownload)
- f Stabil dan tangguh
- f Fleksibel dengan berbagai pemrograman
- f Security yang baik
- f Dukungan dari banyak komunitas
- f Kemudahan *management database*.
- f Mendukung transaksi
- f Perkembangan software yang cukup cepat.

Instalasi MySQL di Windows

Persiapan

1. Download Source MySQL di <http://www.mysql.com/downloads/>

MySQL versi terakhir saat materi ini dibuat adalah MySQL 5.0.45. Silahkan Anda download versi terakhir tersebut dan simpan di komputer Anda. Pada dasarnya, instalasi untuk setiap versi MySQL tidak jauh berbeda.

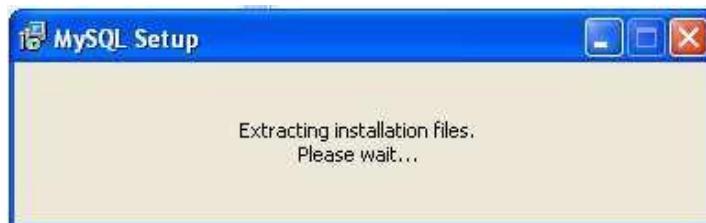
Proses Instalasi MySQL

1. Setelah Anda mendapatkan source MySQL, selanjutnya Anda perlu meng- ekstrak file tersebut ke komputer Anda.
2. Jalankan file **Setup.exe** yang ada di dalam folder source MySQL. Lihat gambar berikut !



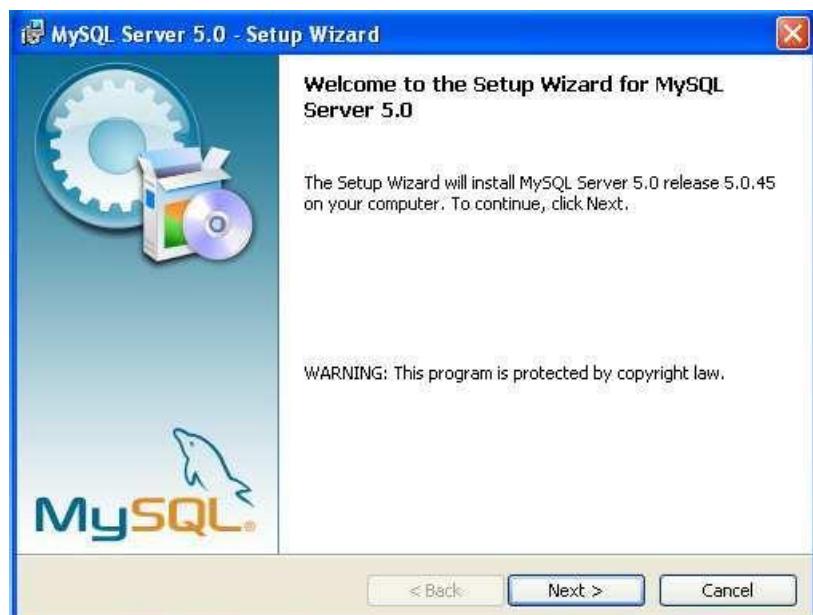
Gambar 2.1. File Setup.exe

3. MySQL Setup akan mengekstrak file instalasi MySQL seperti pada gambar berikut ini.



Gambar 2.2. Proses Instalasi Dimulai

4. Selanjutnya akan ditampilkan window **MySQL Server 5.0 Setup Wizard for MySQL**. Klik tombol **Next** untuk memulai proses instalasi.



Gambar 2.3. Memulai Proses Instalasi

-
5. Selanjutnya akan ditampilkan pilihan untuk memilih cara instalasi. Pilih **Typical** jika kita ingin menginstall MySQL yang umumnya digunakan.



Gambar 2.4. Pilih tipe instalasi

7. Selanjutnya akan ditampilkan window informasi konfigurasi MySQL, yaitu tipe instalasi dan folder tujuan instalasi. Klik **Install** untuk memulai proses instalasi.



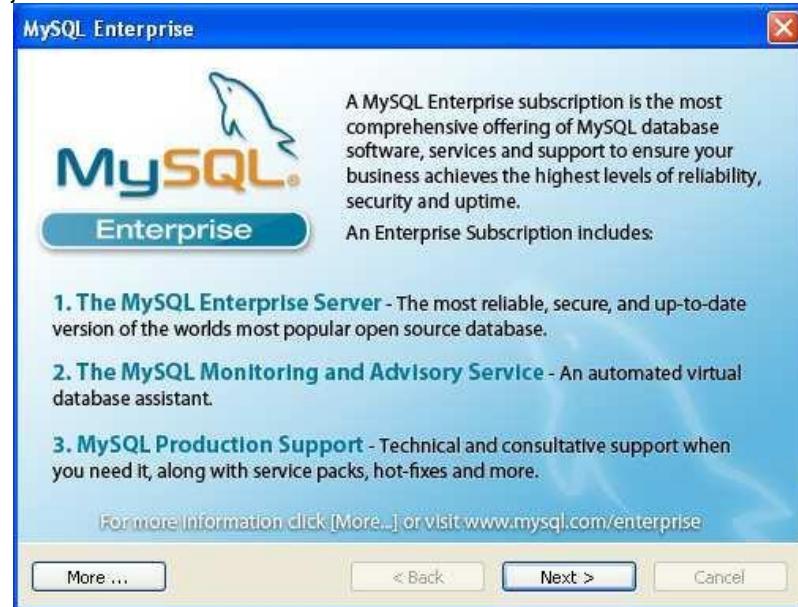
Gambar 2.5. Window Informasi Konfigurasi Instalasi

7. Proses instalasi dimulai.

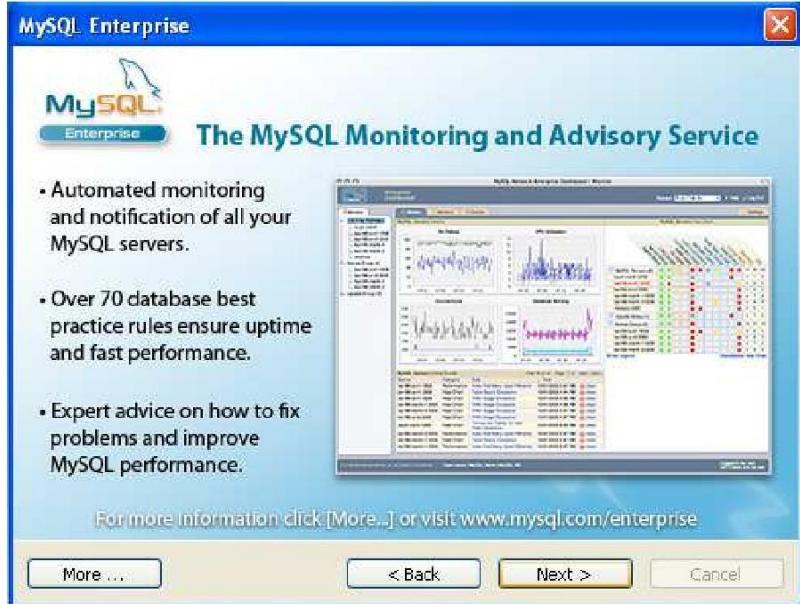


Gambar 2.6. Window Setup Type

8. Selanjutnya ditampilkan window informasi mengenai **MySQL Enterprise**. Klik **Next** untuk melanjutkan.



Gambar 2.7. Window MySQL Enterprise



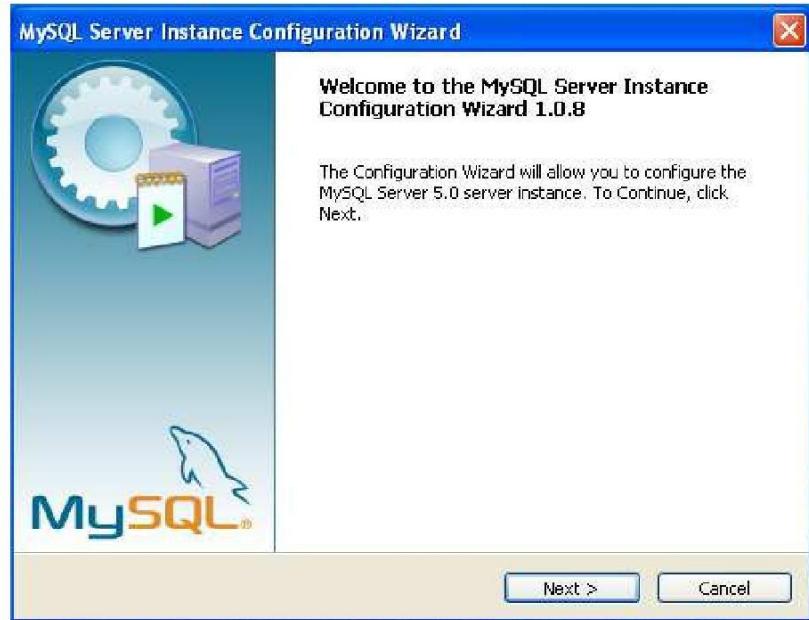
Gambar 2.8. Window informasi MySQL Monitoring

9. Proses instalasi selesai dan akan ditampilkan seperti pada gambar di bawah ini. Jika kita ingin langsung mengkonfigurasikan server MySQL (password, service dll) maka pilihkan checkbox **Configure the MySQL Server now** dan tekan tombol **Finish**.



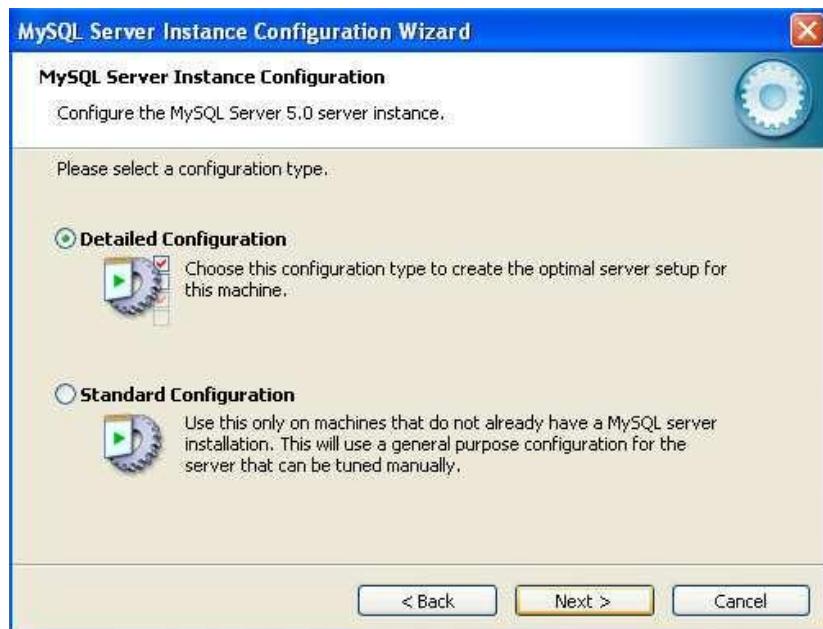
Gambar 2.9. Proses instalasi Selesai

10. Selanjutnya ditampilkan window **MySQL Server Instance Configuration Wizard**. Klik **Next** untuk melanjutkan.



Gambar 2.10. Window MySQL Server Instance Configuration Wizard

11. Selanjutnya terdapat pilihan tipe konfigurasi yang yang diinginkan, diinginkan, **Detailed Configuration** atau **Standard Configuration**. Pilih dan klik **Next** untuk melanjutkan.



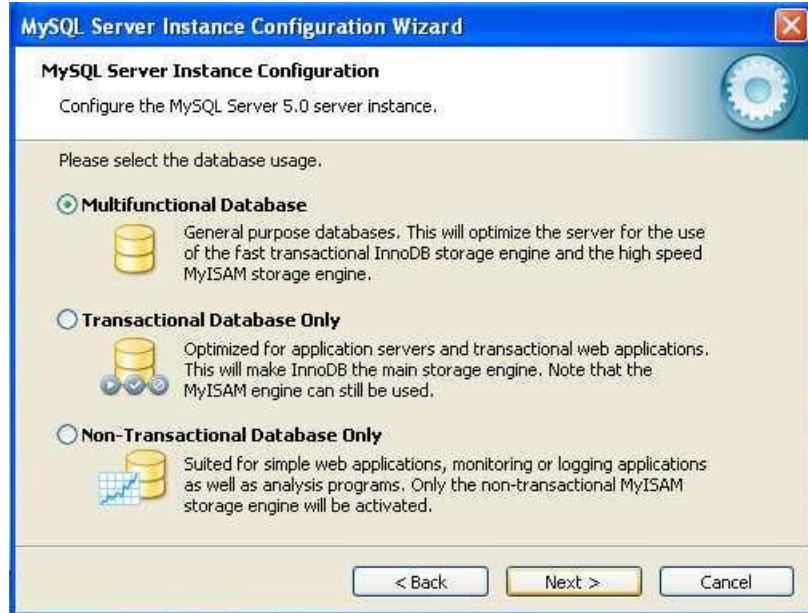
Gambar 2.11. Window Pilihan tipe konfigurasi

12. Selanjutnya terdapat pilihan tipe server yang diinginkan, **Developer**, **Server**, atau **Dedicated MySQL Server**. Pilih salah satu dan klik **Next** untuk melanjutkan.



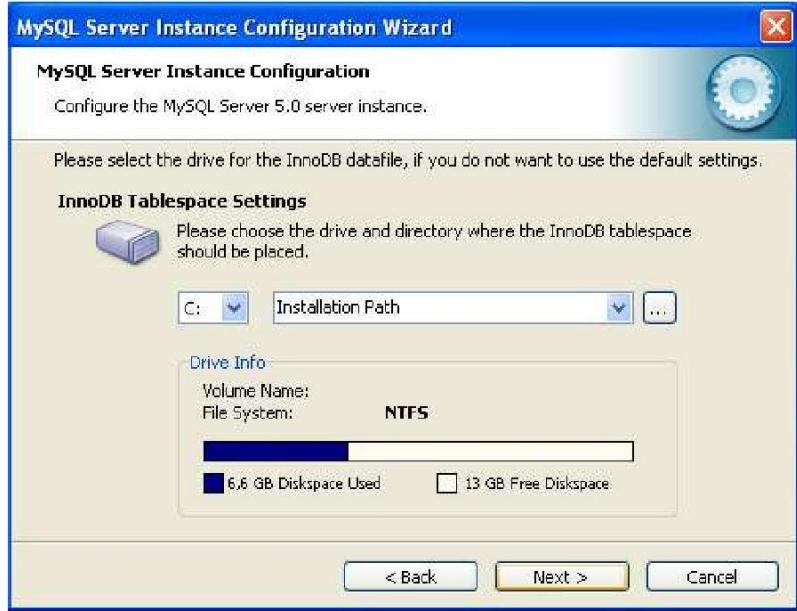
Gambar 2.12. Pilihan tipe server MySQL

13. Selanjutnya terdapat pilihan penggunaan database untuk **Multifunctional, Transactional Only** atau **Non-Transactional Only**. Pilih salah satu dan klik **Next** untuk melanjutkan.



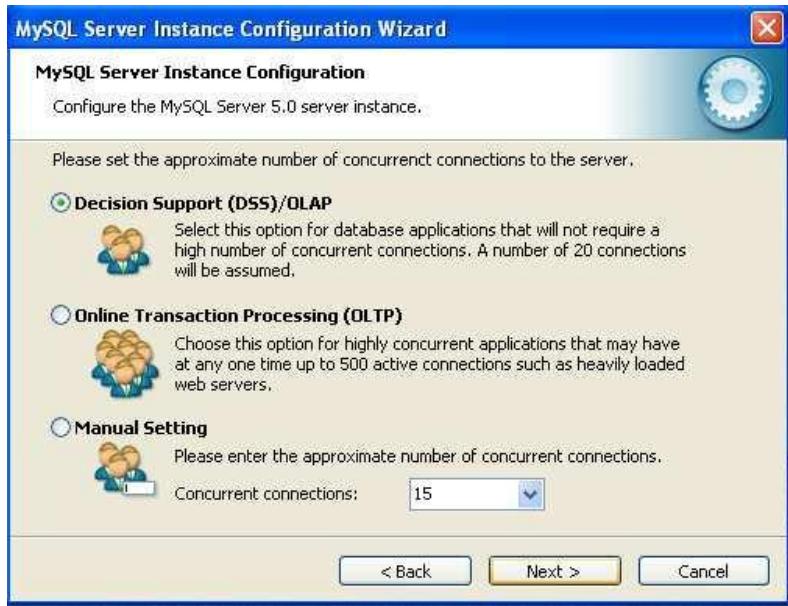
Gambar 2.13. Window Pilihan penggunaan Database.

14. Selanjutnya terdapat dimana diminta memilih tempat untuk InnoDB. Klik **Next** untuk melanjutkan.



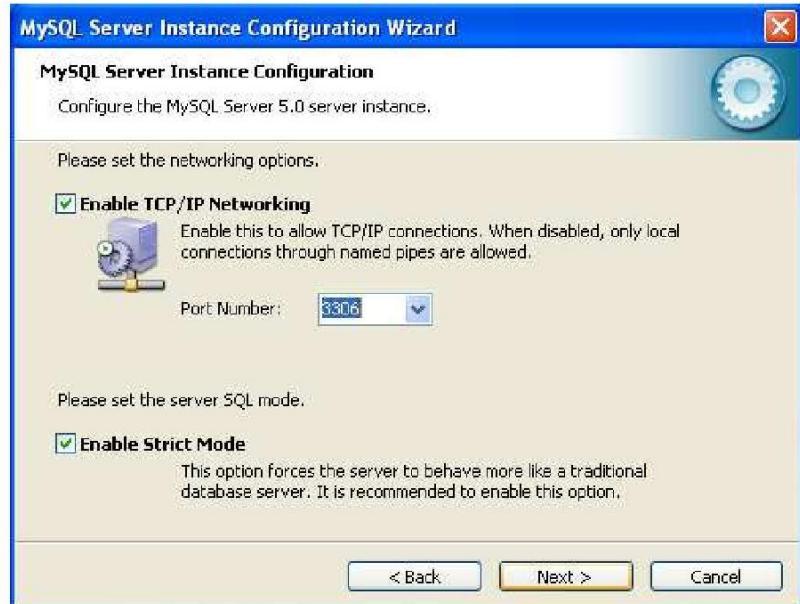
Gambar 2.14. Window InnoDB Tablespace Settings.

15. Selanjutnya terdapat pilihan perkiraan seberapa besar koneksi user ke server. Pilih salah satu dan klik **Next** untuk melanjutkan.



Gambar 2.15. Pilihan Perkiraan Seberapa Besar Koneksi User ke Server

16. Selanjutnya terdapat window untuk memilih nomor PORT yang digunakan untuk MySQL. **Next** untuk melanjutkan.



Gambar 2.16. Window pilihan port MySQL.

17. Selanjutnya terdapat pilihan nama service MySQL yang akan digunakan oleh Windows. Pilih salah satu dan klik **Next** untuk melanjutkan.



Gambar 2.17. Window pilihan Nama Service MySQL.

18. Selanjutnya diminta memodifikasi security. Pilih password untuk root (user tertinggi di MySQL) dan klik **Next** untuk melanjutkan.



Gambar 2.18. Window Security Setting.

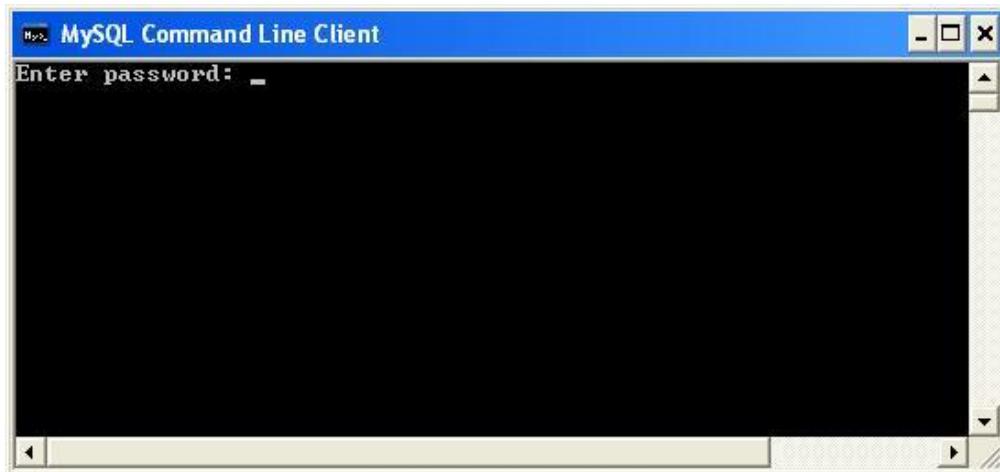
19. Proses konfigurasi selesai dan klik **Execute** untuk menyimpan konfigurasi dan menjalankan servis MySQL.



Gambar 2.19. Proses konfigurasi server MySQL selesai.

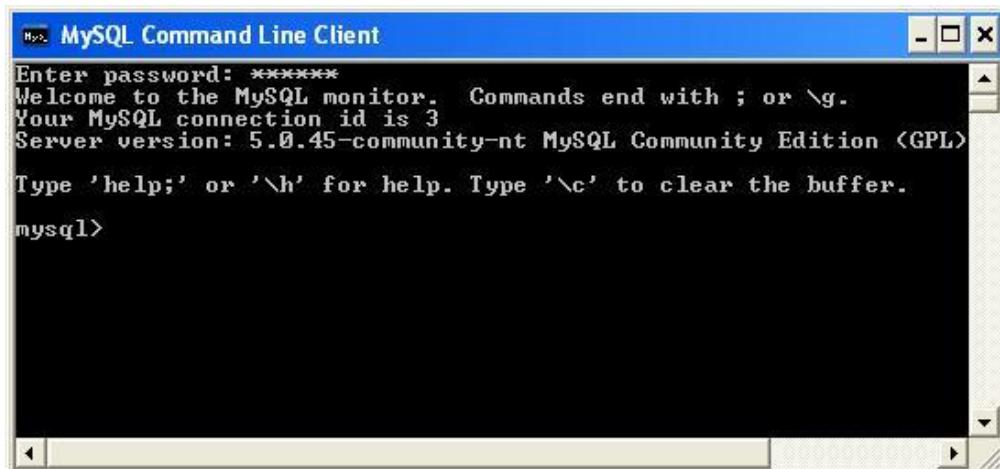
Koneksi ke Server MySQL dengan MySQL Client

MySQL menyediakan tools untuk melakukan koneksi ke server MySQL, yaitu MySQL Command-Line Client. Tools tersebut dapat diakses dari menu **Start > All Programs > MySQL > MySQL Server 5> MySQL Command Line Client**. Tampilannya kurang lebih tampak pada gambar berikut ini:



Gambar 12.13. MySQL Command Line Client

Untuk melakukan koneksi ke server MySQL, Anda cukup mengetikkan password koneksi MySQL. Password ini didefinisikan pada saat proses instalasi. Jika passwordnya benar, maka akan ditampilkan window sbb :



Gambar 2.20. Koneksi ke Server MySQL dengan User root

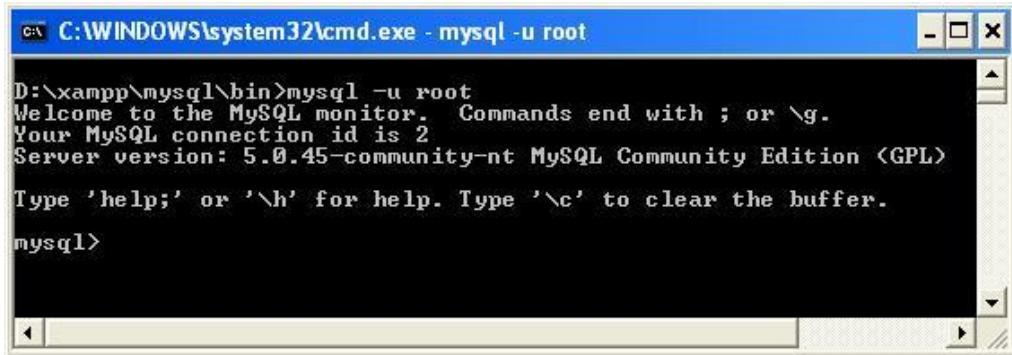
Setelah koneksi ke server MySQL berhasil dilakukan, maka akan ditampilkan prompt mysql> seperti pada gambar 12.14. Query atau perintah-perintah MySQL dapat dituliskan pada prompt MySQL ini. Akhiri setiap query dengan titik-koma (;). Selanjutnya untuk keluar dari server MySQL dapat dilakukan dengan mengetikkan perintah quit atau \q pada *prompt mysql>*.

Berbagai MySQL Client untuk Administrasi Server MySQL

Berikut ini beberapa tools yang biasa digunakan dalam berikut untuk mempermudah administrasi MySQL, ini hanya digunakan dalam berikut untuk mempermudah administrasi MySQL, jadi tidak harus digunakan.

1. MySQL Command Line Client

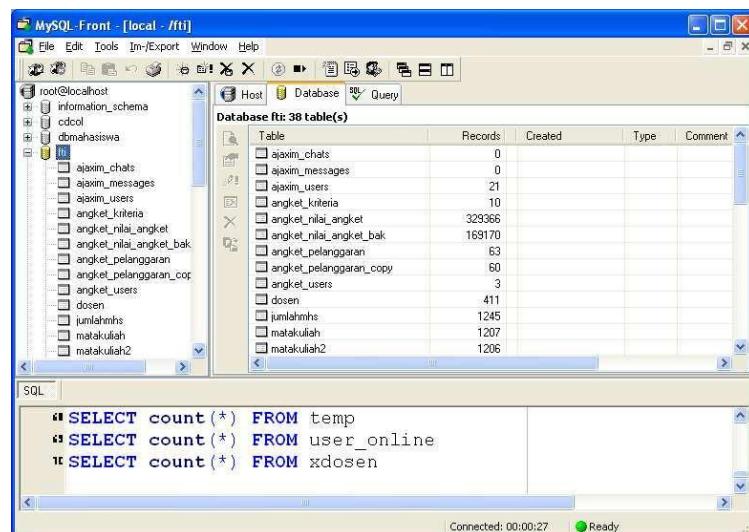
MySQL Command Line Client merupakan tools default MySQL yang sudah disertakan dalam file instalasi MySQL. Aplikasi ini dapat digunakan untuk melakukan koneksi ke MySQL melalui *text-based mode*.



Gambar 2.21. Tampilan MySQL command line client

2. MySQL-Front

MySQL-Front merupakan front-end MySQL berbasis Windows yang cukup banyak digunakan. MySQL-Front memiliki user interface yang cukup mudah digunakan, bahkan oleh user pemula. Pada awalnya MySQL-Front merupakan software yang free, namun mulai versi 3.0 ini menjadi software yang bersifat shareware dengan selama 30 hari. Jika Anda ingin mencoba software ini, cobalah MySQL-Front versi 2.5 karena selain masih bebas untuk didownload, versi 2.5 cukup stabil dan sudah teruji. Situs resmi MySQL-Front beralamat di <http://www.mysqlfront.de>



Gambar 2.22. Tampilan MySQL Front

3. PHPMyAdmin

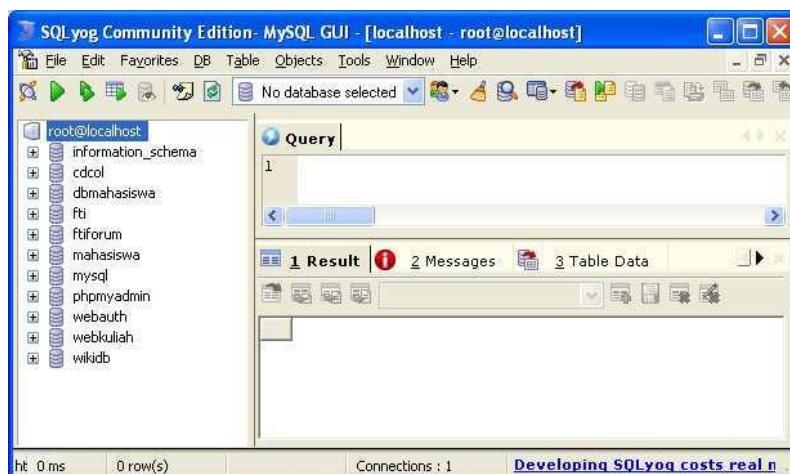
PHPMyAdmin merupakan front-end MySQL berbasis web. PHPMyAdmin dibuat dengan menggunakan PHP. Saat ini, PHPMyAdmin banyak digunakan dalam hampir semua penyedia hosting yang ada di internet. PHPMyAdmin mendukung berbagai fitur administrasi MySQL termasuk manipulasi database, tabel, index dan juga dapat mengeksport data ke dalam berbagai format data. PHPMyAdmin juga tersedia dalam 50 bahasa lebih, termasuk bahasa Indonesia. PHPMyAdmin dapat didownload secara gratis di <http://www.phpmyadmin.net>



Gambar 2.23. Tampilan halaman PHPMyAdmin

4. SQLYog

SQLYog merupakan salah satu front-end MySQL yang cukup populer saat ini. Dengan dukungan fitur yang cukup banyak dan lengkap, SQL Yog tersedia versi commercial dan community (free). SQLYog dapat didownload di situsnya <http://www.webyog.com>



Gambar 2.24. Tampilan layar SQLYog

5. MySQL Administrator dan MySQL Query Browser

MySQL Administrator dan MySQL Query Browser merupakan tools administrasi MySQL yang tersedia di situs resmi

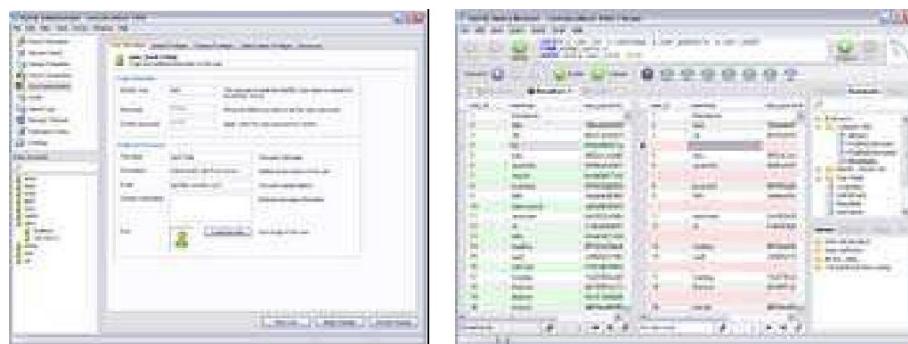
MySQL (<http://www.mysql.com>). Keduanya dapat didownload di alamat <http://www.mysql.com/products/tools/>.

Beberapa fitur **MySQL Administrator**, antara lain:

- Administrasi user.
- Halaman monitoring server.
- Optimisasi MySQL
- Informasi umum keadaan server
- Status *replication*.
- *Cross-platform*.

Beberapa fitur **MySQL Query Browser**, antara lain:

- Tampilan dan menu yang mudah (*user-friendly*).
- Mendukung beberapa window hasil (*result preview*) sekaligus.
- Kemudahan dalam menulis query dengan *visual tools*.
- Manipulasi database.
- Membuat dan manipulasi tabel.
- SQL statements *debugging*.



Gambar 2.25. Tampilan layar MySQL Administrator dan MySQL Control Center

Bab 12

Pengenalan Structured Query Language

- TM Apa Itu SQL ?
- TM Membuat, Menampilkan, Membuka dan Menghapus Database
- TM Membuat, Mengubah dan Menghapus Table
- TM Menambah Record dengan INSERT
- TM Mengedit Record dengan UPDATE
- TM Menghapus Record dengan DELETE
- TM Menampilkan Record dengan SELECT

Apa Itu SQL ?

SQL merupakan singkatan dari *Structured Query Language*. SQL atau juga sering disebut sebagai query merupakan suatu bahasa (*language*) yang digunakan untuk mengakses database. SQL dikenalkan pertama kali dalam IBM pada tahun 1970 dan sebuah standar ISO dan ANSI ditetapkan untuk SQL. Standar ini tidak tergantung pada mesin yang digunakan (IBM, Microsoft atau Oracle). Hampir semua software database mengenal atau mengerti SQL. Jadi, perintah SQL pada semua software database hampir sama.

Terdapat 2 (dua) jenis perintah SQL, yaitu :

1. DDL atau Data Definition Language

DDL merupakan perintah SQL yang berhubungan dengan pendefinisian suatu struktur database, dalam hal ini *database* dan *table*. Beberapa perintah dasar yang termasuk DDL ini antara lain :

- f CREATE
- f ALTER
- f RENAME
- f DROP

2. DML atau Data Manipulation Language

DML merupakan perintah SQL yang berhubungan dengan manipulasi atau pengolahan data atau *record* dalam table. Perintah SQL yang termasuk dalam DML antara lain :

- f SELECT
- f INSERT
- f UPDATE
- f DELETE

Membuat, Menampilkan, Membuka dan Menghapus Database

Membuat Database

Sintaks umum SQL untuk membuat suatu database adalah sebagai berikut :

```
CREATE DATABASE [IF NOT EXISTS] nama_database;
```

Bentuk perintah di atas akan membuat sebuah database baru dengan nama *nama_database*. Aturan penamaan sebuah database sama seperti aturan penamaan sebuah variabel, dimana secara umum nama database boleh terdiri dari huruf, angka dan *under-score* (_). Jika database yang akan dibuat sudah ada, maka akan muncul pesan error. Namun jika ingin otomatis menghapus database yang lama jika sudah ada, aktifkan option IF NOT EXISTS. Setiap kita membuat database baru, maka sebenarnya MySQL akan membuat suatu folder (direktori) sesuai dengan nama databasenya yang ditempatkan secara *default* di C:\mysql\data. Di dalam folder tersebut nantinya akan terdapat file-file yang berhubungan dengan tabel dalam database.

Berikut ini contoh perintah untuk membuat database baru dengan nama “**mahasiswa**” :

|| CREATE DATABASE mahasiswa;

Jika query di atas berhasil dieksekusi dan database berhasil dibuat, maka akan ditampilkan pesan sebagai berikut :

|| Query OK, 1 row affected (0.02 sec)

Membuat Database

Untuk melihat database yang baru saja dibuat atau yang sudah ada, dapat menggunakan perintah sebagai berikut :

|| SHOW DATABASES;

Hasil dari perintah di atas akan menampilkan semua database yang sudah ada di MySQL. Berikut ini contoh hasil dari query di atas :

+	+
Database	
-----	+
mahasiswa	
mysql	
test	
-----	+
3	rows in set (0.02 sec)

Membuka Database

Sebelum melakukan manipulasi tabel dan record yang berada di dalamnya, kita harus membuka atau mengaktifkan databasenya terlebih dahulu. Untuk membuka database “**mahasiswa**”, berikut ini querynya :

|| USE mahasiswa;

Jika perintah atau query di atas berhasil, maka akan ditampilkan pesan sebagai berikut :

|| Database changed

Menghapus Database

Untuk menghapus suatu database, sintaks umumnya adalah sbb :

|| DROP DATABASE [IF EXISTS] nama_database;

Bentuk perintah di atas akan menghapus database dengan nama **nama_database**. Jika databasenya ada maka database dan juga seluruh tabel di dalamnya akan dihapus. Jadi berhati-hatilah dengan perintah ini! Jika nama database yang akan dihapus tidak ditemukan, maka akan ditampilkan pesan error. Aktifkan option IF EXISTS untuk memastikan bahwa suatu database benar-benar ada.

Berikut ini contoh perintah untuk menghapus database dengan nama “**mahasiswa**” :

```
DROP DATABASE mahasiswa;
```

Membuat, Mengubah dan Menghapus TABLE

Membuat Table

Bentuk umum SQL untuk membuat suatu *table* secara sederhana sebagai berikut :

```
CREATE TABLE nama_tabel (
field1 tipe(panjang), field2
tipe(panjang),
...
fieldn tipe(panjang), PRIMARY
KEY (field_key);
```

Bentuk umum di atas merupakan bentuk umum pembuatan tabel yang sudah disederhanakan. Penamaan tabel dan field memiliki aturan yang sama dengan penamaan database.

MySQL menyediakan berbagai tipe data dengan spesifikasi dan panjang masing-masing. Tipe data untuk field dalam MySQL diantaranya ditampilkan pada tabel berikut ini :

JENIS TIPE	TIPE	KETERANGAN
NUMERIK	TINYINT	-128 s/d 127 SIGNED 0 s/d 255 UNSIGNED
	SMALLINT	-32768 s/d 32767 SIGNED 0 s/d 65535 UNSIGNED.
	MEDIUMINT	-8388608 s/d 8388607 SIGNED 0 s/d 16777215 UNSIGNED
	INT	-2147483648 s/d 2147483647 SIGNED 0 s/d 4294967295 UNSIGNED.
	BIGINT	-9223372036854775808 s/d 9223372036854775807 SIGNED 0 s/d 18446744073709551615 UNSIGNED.
	FLOAT	Bilangan pecahan presisi tunggal
	DOUBLE	Bilangan pecahan presisi ganda
	DECIMAL	Bilangan dengan desimal
DATE/TIME	DATE	Tanggal dengan format YYYY-MM-DD
	DATETIME	Tanggal dan waktu dengan format : YYYY-MM-DD HH:MM:SS
	TIMESTAMP	Tanggal dan waktu dengan format : YYYYMMDDHHMMSS
	TIME	Waktu dengan format HH:MM:SS
	YEAR	Tahun dengan format YYYY
STRING	CHAR	0 – 255 karakter
	VARCHAR	0 – 255 karakter
	TINYTEXT	String dengan panjang maksimum 255 karakter
	TEXT	String dengan panjang maksimum 65535 karakter
	BLOB	String dengan panjang maksimum 65535

	MEDIUMTEXT	karakter String dengan panjang maksimum <small>16777215 karakter</small>
	MEDIUMBLOB	String dengan panjang maksimum <small>16777215 karakter</small>
	LONGTEXT	String dengan panjang maksimum <small>4294967295 karakter</small>
	LONGBLOB	String dengan panjang maksimum <small>4294967295 karakter</small>
KHUSUS	ENUM	Tipe data dengan isi tertentu
	SET	Tipe data dengan isi tertentu

Sebagai contoh, kita akan membuat tabel baru dengan struktur sebagai berikut :

Nama tabel : mhs

No	Nama Field	Tipe	Panjang
1	nim	Varchar	10
2	nama	Varchar	30
3	tglahir	Date	-
4	alamat	Text	-

Untuk membuat tabel tersebut di atas, query atau perintah SQL-nya adalah sebagai berikut :

```
CREATE TABLE mhs (
    nim varchar(10) NOT NULL, nama
    varchar(30) NOT NULL, tglahir date,
    alamat text,
    PRIMARY KEY(nim) );
```

Jika query untuk membuat tabel di atas berhasil dijalankan, maka akan ditampilkan pesan sebagai berikut :

Query OK, 0 rows affected (0.16 sec)

Pada perintah di atas, beberapa hal yang perlu diperhatikan :

- f CREATE TABLE merupakan perintah dasar dari pembuatan table.
- f mhs merupakan nama tabel yang akan dibuat.
- f Nim, nama, tglahir dan alamat merupakan nama field
- f Varchar, date dan text merupakan tipe data dari field
- f NOT NULL merupakan option untuk menyatakan bahwa suatu field tidak boleh kosong.
- f PRIMARY KEY merupakan perintah untuk menentukan field mana yang akan dijadikan primary key pada tabel.
- f 10 dan 30 di belakang tipe data merupakan panjang maksimal dari suatu field
- f Untuk tipe data date dan text (dan beberapa tipe data lainnya) panjang karakter maksimalnya tidak perlu ditentukan.
- f Jangan lupa akhiri perintah dengan titik-koma (;

Selanjutnya untuk melihat tabel mhs sudah benar-benar sudah ada atau belum, ketikkan perintah berikut ini :

SHOW TABLES ;

Perintah di atas akan menampilkan seluruh tabel yang sudah ada dalam suatu database. Contoh hasil dari perintah di atas adalah sebagai berikut :

```
+          +
| Tables_in_mahasiswa |
+          +
| mhs           |
+
1 rows in set (0.01 sec)
```

Untuk melihat struktur tabel “mhs” secara lebih detail, cobalah perintah atau query sebagai berikut :

DESC mhs ;

DESC merupakan singkatan dari **DESCRIBE** (dalam query bisa ditulis lengkap atau hanya 4 karakter pertama) dan **mhs** adalah nama tabel yang akan dilihat strukturnya. Dari perintah di atas, akan ditampilkan struktur tabel **mhs** sebagai berikut :

Field	Type	Null	Key	Default	Extra
nim	varchar(10)		PRI		
nama	varchar(30)				
tgllahir	date	YES		NULL	
alamat	text	YES		NULL	

4 rows in set (0.00 sec)

Dari struktur tabel mhs yang ditampilkan di atas, dapat diketahui bahwa :

- f Terdapat 4 (empat) field dengan tipe masing-masing.
- f Primary Key dari tabel mhs adalah nim. Lihat kolom **Key** pada field nim.
- f Untuk field nim dan nama defaultnya tidak boleh kosong. Lihatlah kolom **Null** dan **Default** pada field nim dan nama
- f Untuk field tgllahir dan alamat defaultnya boleh kosong. Lihatlah kolom **Null** dan **Default** pada field tgllahir dan alamat.

Mengubah Struktur Table dengan ALTER

Untuk mengubah struktur suatu tabel, bentuk umum perintah SQL-nya sebagai berikut :

```
ALTER TABLE nama_tabel alter_options;
```

dimana :

f **ALTER TABLE** merupakan perintah dasar untuk mengubah tabel.

f **nama_tabel** merupakan nama tabel yang akan diubah strukturnya.

f **alter_options** merupakan pilihan perubahan tabel. Option yang bisa digunakan, beberapa di antaranya sebagai berikut :

» **ADD definisi_field_baru**

Option ini digunakan
field baru
“definisi_field_baru” (nama field, tipe dan option lain).

untuk
dengan
menambahkan

» **ADD INDEX nama_index**

- Option ini digunakan untuk menambahkan index dengan nama “**nama_index**” pada tabel.
- » ADD PRIMARY KEY (field_kunci)
Option untuk menambahkan primary key pada tabel
 - » CHANGE field_yang_diubah definisi_field_baru
Option untuk mengubah field_yang_diubah menjadi definisi_field_baru
 - » MODIFY definisi_field
Option untuk mengubah suatu field menjadi definisi_field
 - » DROP nama_field
Option untuk menghapus field nama_field
 - » RENAME TO nama_tabel_baru
Option untuk mengganti nama tabel

Beberapa contoh variasi perintah ALTER untuk mengubah struktur suatu tabel antara lain :

1. Menambahkan field “**agama**” ke tabel **mhs**

```
ALTER TABLE mhs ADD agama varchar(15) NOT NULL;
```

2. Menambahkan primary key pada suatu tabel

```
ALTER TABLE mhs ADD PRIMARY KEY(nim);
```

3. Mengubah **panjang** field **agama** menjadi 10 karakter dalam tabel **mhs**

```
ALTER TABLE mhs CHANGE agama agama varchar(10);
```

4. Mengubah **tipe** field **agama** menjadi char(2) dalam tabel **mhs**

```
ALTER TABLE mhs MODIFY agama char(2) NOT NULL;
```

5. Menghapus field **agama** dari tabel **mhs**

```
ALTER TABLE mhs DROP agama;
```

Mengubah Nama Tabel

Untuk mengubah nama suatu tabel, dapat menggunakan perintah SQL sbb :

```
RENAME TABLE mhs TO mahasiswa;  
ALTER TABLE mhs RENAME TO mahasiswa;
```

Perintah di atas akan mengubah tabel **mhs** menjadi **mahasiswa**.

Menghapus Tabel

Untuk menghapus sebuah tabel, bentuk umum dari perintah SQL adalah sebagai berikut :

```
DROP TABLE nama_tabel;
```

Contohnya kita akan menghapus tabel dengan nama “**mahasiswa**” maka perintah SQL-nya adalah :

```
DROP TABLE mahasiswa;
```


Menambah Record dengan INSERT

Bentuk umum perintah SQL untuk menambahkan record atau data ke dalam suatu tabel adalah sebagai berikut :

```
INSERT INTO nama_tabel VALUES ('nilai1','nilai2',...);
```

atau dapat dengan bentuk sebagai berikut :

```
INSERT INTO nama_tabel(field1,field2,...)  
VALUES ('nilai1','nilai2',...);
```

atau dapat juga dengan bentuk sebagai berikut :

```
INSERT INTO nama_tabel  
SET field1='nilai1', field2='nilai2',...;
```

Sebagai contoh, kita akan menambahkan sebuah record ke dalam tabel **mhs** yang telah kita buat sebelumnya. Berikut ini perintah SQL untuk menambahkan sebuah record ke dalam tabel **mhs** :

```
INSERT INTO mhs VALUES ('0411500121','Achmad Solichin',  
'1982-06-05','Jakarta Selatan');
```

Jika perintah SQL di atas berhasil dieksekusi maka akan ditampilkan pesan sebagai berikut :

```
Query OK, 1 row affected (0.00 sec)
```

Setelah perintah SQL di atas berhasil dieksekusi, maka record atau data dalam tabel **mhs** akan bertambah. Jalankan perintah berikut ini untuk melihat isi tabel **mhs** !

```
SELECT * FROM mhs;
```

Dan berikut ini hasil dari perintah SQL di atas :

nim	nama	tglahir	alamat
0411500121	Achmad Solichin	1982-06-05	Jakarta Selatan
rowinset (0.19 sec)			

Latihan

Tambahkan 10 data (record) baru ke tabel **mhs** sehingga isi tabel **mhs** menjadi sebagai berikut !

nim	nama	tglahir	alamat
0411500121	Achmad Solichin	1982-06-05	Jakarta Selatan
0411500123	Chotimatul M	1983-03-12	Jakarta Selatan
0422500111	Bajuri	1983-03-25	Tangerang
0444500011	Oneng	1980-05-22	Jakarta Utara
0433500115	Unyil	1980-08-29	Tangerang

0411500116	Ujang	1984-10-06	Jakarta Barat	
0422500316	Jebleh	1984-10-06	Cengkareng	
0433500333	Dono	1984-10-06	Jakarta Selatan	
0422500433	Dini	1986-12-10	Jakarta Selatan	
0411500331	Dana	1986-07-11	Jakarta Selatan	
0444500315	Dani	1985-01-01	Jakarta Barat	
+-----		+-----	+-----	+

Mengedit Record dengan UPDATE

Proses update bisa sewaktu-waktu dilakukan jika terdapat data atau record dalam suatu tabel yang perlu diperbaiki. Proses update ini tidak menambahkan data (record) baru, tetapi memperbaiki data yang lama. Perubahan yang terjadi dalam proses update bersifat permanen, artinya setelah perintah dijalankan tidak dapat di-cancel (*undo*).

Bentuk umum perintah SQL untuk mengedit suatu *record* atau data dari suatu tabel adalah sebagai berikut :

```
UPDATE nama_tabel SET field1='nilaibaru' [WHERE
kondisi];
```

Pada perintah untuk update di atas :

- f UPDATE merupakan perintah dasar untuk mengubah *record* tabel.
- f *nama_tabel* merupakan nama tabel yang akan diubah *recordnya*.
- f Perintah SET diikuti dengan *field-field* yang akan diubah yang mana diikuti juga dengan perubahan isi dari masing-masing *field*. Untuk mengubah nilai dari beberapa *field* sekaligus, gunakan koma (,) untuk memisahkan masing-masing *field*.
- f Perintah WHERE diikuti oleh kondisi tertentu yang menentukan record mana yang akan diedit (diubah). Perintah WHERE ini boleh ada boleh juga tidak. Jika WHERE tidak ditambahkan pada perintah update maka semua *record* dalam tabel bersangkutan akan berubah.

Perhatikan beberapa contoh perintah UPDATE tabel **mhs** berikut ini !

1. Mengubah alamat menjadi "Tangerang" untuk mahasiswa yang mempunyai nim 0411500121

```
UPDATE mhs SET alamat='Tangerang' WHERE
nim='0411500121';
```

Dan jika query di atas berhasil dieksekusi maka akan ditampilkan hasil sebagai berikut :

```
Query OK, 1 row affected (0.27 sec)
Rows matched: 1    Changed: 1    Warnings: 0
```

2. Mengubah tanggal lahir menjadi "12 Maret 1983" dan alamat menjadi "Bandung" untuk mahasiswa yang mempunyai nim 0422500316

```
UPDATE mhs SET tgllahir='1982-06-05', alamat='Jakarta
Selatan' WHERE nim='0422500316';
```

Menghapus Record dengan DELETE

Proses delete dilakukan jika terdapat data atau record dalam suatu tabel yang perlu dihapus atau dihilangkan. Perubahan yang terjadi dalam proses *delete* bersifat permanen, artinya setelah perintah dijalankan tidak dapat di-*cancel* (*undo*). Jadi berhati-hatilah dengan perintah *delete* !

Bentuk umum perintah SQL untuk menghapus suatu *record* atau data dari tabel adalah sebagai berikut :

```
DELETE FROM nama_tabel [WHERE kondisi];
```

Pada perintah untuk *delete* di atas :

- f **DELETE FROM** merupakan perintah dasar untuk menghapus suatu *record* dari tabel.
- f **nama_tabel** merupakan nama tabel yang akan dihapus *recordnya*.
- f Perintah **WHERE** diikuti oleh kondisi tertentu yang menentukan record mana yang akan dihapus (didelete). Perintah **WHERE** ini boleh ada boleh juga tidak. Namun demikian, jika **WHERE** tidak ditambahkan pada perintah delete maka semua *record* dalam tabel bersangkutan akan **terhapus**. Jadi jangan lupa menambahkan **WHERE** jika kita tidak bermaksud mengosongkan tabel

Perhatikan beberapa contoh perintah **DELETE** dari tabel **mhs** berikut ini !

1. Menghapus data mahasiswa yang mempunyai nim 0411500331

```
DELETE FROM mhs WHERE nim='0411500331';
```

Dan jika query di atas berhasil dieksekusi maka akan ditampilkan hasil sebagai berikut :

```
Query OK, 1 row affected (0.11 sec)
```

2. Menghapus semua mahasiswa yang beralamat di "Bandung"

```
DELETE FROM mhs WHERE alamat='Bandung';
```

Menampilkan Record dengan SELECT

Perintah **SELECT** digunakan untuk menampilkan sesuatu. Sesuatu di sini bisa berupa sejumlah data dari tabel dan bisa juga berupa suatu ekspresi. Dengan **SELECT** kita bisa mengatur tampilan atau keluaran sesuai tampilan yang diinginkan.

Bentuk dasar perintah **SELECT** data dari tabel adalah sebagai berikut :

```
SELECT [field | *] FROM nama_tabel [WHERE kondisi];
```

Perhatikan beberapa contoh perintah **SELECT** dari tabel **mhs** berikut ini !

1. Menampilkan seluruh data atau record (*) dari tabel **mhs**

```
SELECT * FROM mhs;
```

Dan jika query di atas berhasil dieksekusi maka akan ditampilkan hasil sebagai berikut :

+ nim	+ nama	+ tglahir	+ alamat	+
0411500121 Achmad Solichin	1982-06-05 Jakarta Selatan			
0411500123 Chotimatal M	1983-03-12 Jakarta Selatan			
0422500111 Bajuri	1983-03-25 Tangerang			
0444500011 Oneng	1980-05-22 Jakarta Utara			
0433500115 Unyil	1980-08-29 Tangerang			
0411500116 Ujang	1984-10-06 Jakarta Barat			
0422500316 Jebleh	1982-06-05 Jakarta Selatan			
0433500333 Dono	1984-10-06 Jakarta Selatan			
0422500433 Dini	1986-12-10 Jakarta Selatan			
0444500315 Dani	1985-01-01 Jakarta Barat			

2. Menampilkan **field nim** dan **nama** dari seluruh mahasiswa dalam tabel **mhs**

```
SELECT nim, nama FROM mhs;
```

Jika query di atas berhasil dieksekusi maka akan ditampilkan hasil sebagai berikut :

-----+-----+ nim nama
0411500121 Achmad Solichin
0411500123 Chotimatul M
0422500111 Bajuri
0444500011 Oneng
0433500115 Unyil
0411500116 Ujang
0422500316 Jebleh
0433500333 Dono
0422500433 Dini
0444500315 Dani

3. Menampilkan data mahasiswa yang mempunyai nim **0411500123**

```
SELECT * FROM mhs WHERE nim = '0411500123';
```

Hasil query di atas adalah sbb :

nim	nama	tgllahir	alamat
+-----	+-----	+-----	+-----
0411500123 Chotimatul M 1983-03-12 Jakarta Selatan			
+-----	+-----	+-----	+-----
1	rowinset (0.06 sec)		

4. Menampilkan data semua mahasiswa yang beralamat di luar **Jakarta Selatan**


```
SELECT * FROM mhs WHERE alamat != 'Jakarta Selatan';
```

Hasil query di atas adalah sbb :

nim	nama	tglahir	alamat
0422500111	Bajuri	1983-03-25	Tangerang
0444500011	Oneng	1980-05-22	Jakarta Utara
0433500115	Unyil	1980-08-29	Tangerang
0411500116	Ujang	1984-10-06	Jakarta Barat
0444500315	Dani	1985-01-01	Jakarta Barat

5 rows in set (0.02 sec)

Berikut ini operator **perbandingan** yang dapat digunakan untuk membandingkan dua buah nilai dalam MySQL :

- f **Operator =**, akan bernilai TRUE jika nilai yang dibandingkan sama.
- f **Operator !=** atau **<>**, akan bernilai TRUE jika nilai yang dibandingkan TIDAK SAMA (berbeda).
- f **Operator >**, akan bernilai TRUE jika nilai yang pertama lebih besar dari nilai kedua.
- f **Operator >=**, akan bernilai TRUE jika nilai yang pertama lebih besar atau sama dengan nilai kedua.
- f **Operator <**, akan bernilai TRUE jika nilai yang pertama lebih kecil dari nilai kedua.
- f **Operator <=**, akan bernilai TRUE jika nilai yang pertama lebih kecil atau sama dengan nilai kedua.

5. Menampilkan data semua mahasiswa yang beralamat di **Jakarta Selatan** dan lahir pada tahun **1982**.

```
SELECT * FROM mhs WHERE alamat = 'Jakarta Selatan' &&  
YEAR(tglahir) = '1982';
```

Hasil query di atas adalah sbb :

nim	nama	tglahir	alamat
0411500121	Achmad Solichin	1982-06-05	Jakarta Selatan
0422500316	Jebleh	1982-06-05	Jakarta Selatan

2 rows in set (0.00 sec)

Berikut ini operator **penghubung** yang dapat digunakan untuk menghubungkan antara dua kondisi dalam MySQL :

- f **Operator &&** atau **AND**, akan menghubungkan dua kondisi dimana akan bernilai TRUE jika kedua kondisi bernilai TRUE.
- f **Operator ||** atau **OR**, akan menghubungkan dua kondisi dimana akan bernilai TRUE jika salah satu atau kedua kondisi bernilai TRUE.
- f **Operator !**, akan me-reverse nilai suatu kondisi logika.

Keterangan

Fungsi **YEAR** pada query di atas akan menghasilkan nilai TAHUN dari suatu tanggal. Selain fungsi **YEAR**, juga terdapat fungsi **MONTH** yang akan menghasilkan nama BULAN dari tanggal, fungsi **DAY** yang akan menghasilkan

hari dari suatu tanggal, dan masih banyak fungsi lain yang berhubungan dengan tanggal.

- Menampilkan **nim**, **nama** dan **umur** dari semua mahasiswa.

```
SELECT nim, nama, YEAR(now())-YEAR(tgllahir) AS umur  
FROM mhs;
```

Hasil query di atas adalah sbb :

nim	nama	umur
0411500121	Achmad Solichin	24
0411500123	Chotimatul M	23
0422500111	Bajuri	23
0444500011	Oneng	26
0433500115	Unyil	26
0411500116	Ujang	22
0422500316	Jebleh	24
0433500333	Dono	22
0422500433	Dini	20
0444500315	Dani	21

Keterangan

Pada query di atas terdapat fungsi **YEAR** yang akan mengambil tahun dari suatu tanggal. Selanjutnya fungsi **now()** akan me-return tanggal dan waktu sistem saat query dieksekusi. Proses perhitungan umur dialiaskan dengan nama '**umur**'. Untuk mengaliaskan gunakan perintah **AS** yang diikuti nama alias.

- Menampilkan semua mahasiswa **jurusan TI**

```
SELECT * FROM mhs WHERE SUBSTRING(nim,3,2)='11';
```

Hasil query di atas adalah sbb :

nim	nama	tgllahir	alamat
0411500121	Achmad Solichin	1982-06-05	Jakarta Selatan
0411500123	Chotimatul M	1983-03-12	Jakarta Selatan
0411500116	Ujang	1984-10-06	Jakarta Barat

Keterangan

Pada query di atas terdapat fungsi **SUBSTRING** yang berguna untuk memotong suatu string. Format fungsi SUBSTRING adalah sebagai berikut :

SUBSTRING(field, awal, panjang)

- Menampilkan semua data mahasiswa secara urut berdasarkan **nama** dengan perintah **ORDER BY**

```
SELECT * FROM mhs ORDER BY nama;
```

Hasil query di atas adalah sbb :

nim	nama	tglahir	alamat
0411500121	Achmad Solichin	1982-06-05	Jakarta Selatan
0422500111	Bajuri	1983-03-25	Tangerang
0411500123	Chotimatul M	1983-03-12	Jakarta Selatan
0444500315	Dani	1985-01-01	Jakarta Barat
0422500433	Dini	1986-12-10	Jakarta Selatan
0433500333	Dono	1984-10-06	Jakarta Selatan
0422500316	Jebleh	1982-06-05	Jakarta Selatan
0444500011	Oneng	1980-05-22	Jakarta Utara
0411500116	Ujang	1984-10-06	Jakarta Barat
0433500115	Unyil	1980-08-29	Tangerang

10 rows in set (0.01 sec)

9. semua data mahasiswa secara urut berdasarkan **nim** secara

```
SELECT * FROM mhs ORDER BY nim DESC;
```

Hasil query di atas adalah sbb :

nim	nama	tglahir	alamat
0444500315	Dani	1985-01-01	Jakarta Barat
0444500011	Oneng	1980-05-22	Jakarta Utara
0433500333	Dono	1984-10-06	Jakarta Selatan
0433500115	Unyil	1980-08-29	Tangerang
0422500433	Dini	1986-12-10	Jakarta Selatan
0422500316	Jebleh	1982-06-05	Jakarta Selatan
0422500111	Bajuri	1983-03-25	Tangerang
0411500123	Chotimatul M	1983-03-12	Jakarta Selatan
0411500121	Achmad Solichin	1982-06-05	Jakarta Selatan
0411500116	Ujang	1984-10-06	Jakarta Barat

10. Menampilkan 5 record (data) pertama dari tabel **mhs** secara urut berdasarkan **nim** dengan **LIMIT**

```
SELECT * FROM mhs ORDER BY nim LIMIT 0,5;
```

Hasil query di atas adalah sbb :

nim	nama	tglahir	alamat
0411500116	Ujang	1984-10-06	Jakarta Barat
	0411500121 Achmad Solichin	1982-06-05 Jakarta Selatan	
0411500123	Chotimatul M	1983-03-12	Jakarta Selatan
0422500111	Bajuri	1983-03-25	Tangerang
0422500316	Jebleh	1982-06-05	Jakarta Selatan

5 rows in set (0.13 sec)

Keterangan

Pada query di atas bentuk **LIMIT** digunakan untuk membatasi hasil tampilan. LIMIT banyak digunakan untuk menampilkan data yang relatif banyak. Format fungsi LIMIT adalah sebagai berikut :

LIMIT awal, jumlah_record

Bab 13

Aplikasi Berita dengan PHP dan MySQL

- | | |
|----|---|
| TM | Merancang Struktur Database |
| TM | Membuat File Koneksi Database |
| TM | Membuat Halaman Input Berita |
| TM | Menampilkan Berita Terbaru di Halaman Depan |
| TM | Menampilkan Berita Lengkap |
| TM | Membuat Halaman Arsip Berita |
| TM | Membuat Halaman Edit Berita |
| TM | Membuat Halaman Delete Berita |
| TM | CSS sebagai Pemanis Tampilan |

Merancang Struktur Database

Untuk membuat aplikasi berita secara sederhana, langkah pertama adalah merancang tabel-tabel database yang diperlukan. Secara sederhana, struktur tabel dapat dibuat dengan spesifikasi sebagai berikut :

Tabel “berita”

Nama Field	Tipe	Panjang	Keterangan
id_berita	Int	5	Primary Key, Auto Increment
id_kategori	Int	3	Foreign Key
judul	Varchar	100	Judul berita
headline	Text	-	Berita singkat
isi	Text	-	Berita lengkap
pengirim	Varchar	15	Pengirim berita
tanggal	Datetime	-	Tanggal dan waktu kirim berita

Tabel “kategori”

Nama Field	Tipe	Panjang	Keterangan
id_kategori	Int	3	Primary Key, Auto Increment
nm_kategori	Varchar	30	Nama kategori
deskripsi	Varchar	200	Deskripsi kategori

Berikut ini perintah (SQL) untuk membuat tabel “berita”

```
CREATE TABLE berita (
    id_berita int(5) unsigned NOT NULL auto_increment,
    id_kategori int(3) unsigned NOT NULL default '0',
    judul varchar(100) NOT NULL default '',
    headline text NOT NULL,
    isi text NOT NULL,
    pengirim varchar(15) NOT NULL default '',
    tanggal datetime NOT NULL default '0000-00-00 00:00:00',
    PRIMARY KEY (id_berita)
) TYPE=MyISAM;
```

Berikut ini perintah (SQL) untuk membuat tabel “kategori”

```
CREATE TABLE kategori (
    id_kategori int(3) unsigned NOT NULL auto_increment,
    nm_kategori varchar(30) NOT NULL default '',
    deskripsi varchar(200) NOT NULL default '',
    PRIMARY KEY (id_kategori),
) TYPE=MyISAM;
```

Catatan : Isikan sejumlah data ke tabel **kategori** langsung di database

Membuat File Koneksi Database

Program 1

Nama File : *koneksi.php*
Deskripsi : Program koneksi ke database.

```
1 <?php
2 $host = "localhost";
3 $user = "root";
4 $pass = "";
5 $dbnm = "pw2";
6
7 $conn = mysql_connect ($host, $user, $pass);
8 if ($conn) {
9     $buka = mysql_select_db ($dbnm);
10    if (!$buka) {
11        die ("Database tidak dapat dibuka");
12    }
13 } else {
14     die ("Server MySQL tidak terhubung");
15 }
16
17 ?>
```

Membuat Halaman Input Berita

Program 2

Nama File : *input_berita.php*
Deskripsi : Program input berita.

```
1 <?php
2 include "koneksi.php";
3
4 //proses input berita
5 if (isset($_POST['Input'])) {
6     $judul = addslashes (strip_tags ($_POST['judul']));
7     $kategori = $_POST['kategori'];
8     $headline = addslashes (strip_tags ($_POST['headline']));
9     $isi_berita = addslashes (strip_tags ($_POST['isi']));
10    $pengirim = addslashes (strip_tags ($_POST['pengirim']));
11    //insert ke tabel
12    $query = "INSERT INTO berita
13        VALUES('$kategori','$judul','$headline','$isi_berita','$pengirim', now())";
14
15    $sql = mysql_query ($query);
16    if ($sql) {
17        echo "<h2><font color=blue>Berita telah berhasil
ditambahkan</font></h2>";
18    } else {
19        echo "<h2><font color=red>Berita gagal
ditambahkan</font></h2>";
20    }
21 }
22 ?>
23 <html>
24     <head><title>Input Berita</title>
25     <link rel="stylesheet" href="style.css">
26     </head>
27     <body>
28         <a href="index.php">Halaman Depan</a> |
29         <a href="arsip_berita.php">Arsip Berita</a> |
30         <a href="input_berita.php">Input Berita</a>
31         <br><br>
```

```

30      <FORM ACTION="" METHOD="POST" NAME="input">
31  <table cellpadding="0" cellspacing="0" border="0" width="700">
32      <tr>
33          <td colspan="2"><h2>Input Berita</h2></td>
34      </tr>
35      <tr>
36          <td width="200">Judul Berita</td>
37          <td>: <input type="text" name="judul"
size="30"></td>
38      </tr>
39      <tr>
40          <td>Kategori</td>
41          <td>:
42          <select name="kategori">
43              <?
44                  $query = "SELECT id_kategori, nm_kategori
FROM kategori ORDER BY nm_kategori";
45                  $sql = mysql_query ($query);
46                  while ($hasil = mysql_fetch_array ($sql)) {
47                      echo "<option
value='".$hasil[id_kategori].">'$hasil[nm_kategori]</option>";
48                  }
49              ?>
50          </select></td>
51      </tr>
52      <tr>
53          <td>Headline Berita</td>
54          <td>: <textarea name="headline" cols="50" rows="4">
</textarea></td>
55      </tr>
56      <tr>
57          <td>Isi Berita</td>
58          <td>: <textarea name="isi" cols="50" rows="10">
</textarea></td>
59      </tr>
60      <tr>
61          <td>Pengirim</td>
62          <td>: <input type="text" name="pengirim" size="20">
</td>
63      </tr>
64      <tr>
65          <td>&nbsp;</td>
66          <td>&nbsp;&nbsp;<input type="submit" name="Input"
value="Input Berita">&nbsp;
                <input type="reset" name="reset" value="Cancel">
</td>
67      </tr>
68      </table>
69      </FORM>
70  </body>
71 </html>
72

```

Menampilkan Berita Terbaru di Halaman Depan

Program 3

Nama File : index.php

Deskripsi : Program menampilkan 5 berita terbaru di halaman depan.

```

1  <?php
2 include "koneksi.php";
3
4 ?>
5 <html>
6     <head><title>Index Berita</title>

```

```

7 <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10 <a href="index.php">Halaman Depan</a> |
11 <a href="arsip_berita.php">Arsip Berita</a> |
12 <a href="input_berita.php">Input Berita</a>
13 <br><br>
14 <h2>Halaman Depan ~ Lima Berita Terbaru</h2>
15 <?
16 $query = "SELECT A.id_berita, B.nm_kategori, A.judul, A.headline, A.pengirim,
A.tanggal
17 FROM berita A, kategori B WHERE
A.id_kategori=B.id_kategori
18 ORDER BY A.id_berita DESC LIMIT 0,5";
19 $sql = mysql_query ($query);
20 while ($hasil = mysql_fetch_array ($sql)) {
21     $id_berita = $hasil['id_berita'];
22     $kategori = stripslashes ($hasil['nm_kategori']);
23     $judul = stripslashes ($hasil['judul']);
24     $headline = nl2br(stripslashes
($hasil['headline']));
25     $pengirim = stripslashes ($hasil['pengirim']);
26     $tanggal = stripslashes ($hasil['tanggal']);
27     //
28     //tampilkan berita
29     echo "<font size=4><a
href='berita_lengkap.php?id=$id_berita'>$judul</a></font><br>";
30     echo "<small>Berita dikirimkan oleh <b>$pengirim</b>
pada tanggal <b>$tanggal</b> dalam kategori
<b>$kategori</b></small>";
31     echo "<p>$headline</p>";
32     echo "<hr>";
33 }
34 ?
35 </body>
36 </html>
37

```

Menampilkan Berita Lengkap

Program 4

Nama File : berita_lengkap.php

Deskripsi : Program menampilkan berita secara lengkap.

```

1 <?php
2 include "koneksi.php";
3
4 if (isset($_GET['id'])) {
5     $id_berita = $_GET['id'];
6 } else {
7     die ("Error. No Id Selected! ");
8 }
9 ?>
10 <html>
11     <head><title>Berita Lengkap</title>
12     <link rel="stylesheet" href="style.css">
13     </head>
14     <body>
15     <a href="index.php">Halaman Depan</a> |
16     <a href="arsip_berita.php">Arsip Berita</a> |
17     <a href="input_berita.php">Input Berita</a>
18     <br><br>
19     <h2>Berita Lengkap</h2>
20     <?
21     $query = "SELECT A.id_berita, B.nm_kategori, A.judul,
A.isi, A.pengirim, A.tanggal
22         FROM berita A, kategori B WHERE

```

```

A.id_kategori=B.id_kategori && A.id_berita='$id_berita";
23   $sql = mysql_query ($query);
24   $hasil = mysql_fetch_array ($sql);
25   $id_berita = $hasil['id_berita'];
26   $kategori = stripslashes ($hasil['nm_kategori']);
27   $judul = stripslashes ($hasil['judul']);
28   $isi = nl2br(stripslashes ($hasil['isi']));
29   $pengirim = stripslashes ($hasil['pengirim']);
30   $tanggal = stripslashes ($hasil['tanggal']);
31   //
32   //tampilkan berita
33   echo "<font size=5 color=blue>$judul</font><br>";
34   echo "<small>Berita dikirimkan oleh <b>$pengirim</b>
      pada tanggal <b>$tanggal</b> dalam kategori
<b>$kategori</b></small>";
35   echo "<p>$isi</p>";
36   ?
37   </body>
38 </html>
39

```

Membuat Halaman Arsip Berita

Program 5

Nama File : *arsip_berita.php*
 Deskripsi : *Program arsip keseluruhan berita.*

```

1  <?php
2  include "koneksi.php";
3
4  ?>
5  <html>
6    <head><title>Arsip Berita</title>
7    <link rel="stylesheet" href="style.css">
8    <script language="javascript">
9      function tanya() {
10        if (confirm ("Apakah Anda yakin akan menghapus
berita ini ?")) {
11          return true;
12        } else {
13          return false;
14        }
15      }
16    </script>
17    </head>
18    <body>
19      <a href="index.php">Halaman Depan</a> |
20      <a href="arsip_berita.php">Arsip Berita</a> |
21      <a href="input_berita.php">Input Berita</a>
22      <br><br>
23      <h2>Arsip Berita</h2>
24      <ol>
25        <?
26
27 $query = "SELECT A.id_berita, B.nm_kategori, A.judul, A.pengirim, A.tanggal
28   FROM berita A, kategori B WHERE
A.id_kategori=B.id_kategori
29   ORDER BY A.id_berita DESC";
30   $sql = mysql_query ($query);
31   while ($hasil = mysql_fetch_array ($sql)) {
32     $id_berita = $hasil['id_berita'];
33     $kategori = stripslashes ($hasil['nm_kategori']);
34     $judul = stripslashes ($hasil['judul']);
35     $pengirim = stripslashes ($hasil['pengirim']);
36     $tanggal = stripslashes ($hasil['tanggal']);
37     //

```

```
38 //tampilkan arsip berita
39 echo "<li><a
40 href='berita_lengkap.php?id=$id_berita'>$judul</a><br>".  
41 echo "<small>Berita dikirimkan oleh <b>$pengirim</b>
42 pada tanggal <b>$stanggal</b> dalam kategori  
<b>$kategori</b><br>";
43 echo "<b>Action : </b><a
44 href='edit_berita.php?id=$id_berita'>Edit</a> | ";
45 echo "<a href='delete_berita.php?id=$id_berita'
46 onClick='return tanya()'>Delete</a>";
47 echo "</small></li><br><br>";
48 }
49 ?>
50 </ol>
51 </body>
52 </html>
```

Membuat Halaman Edit Berita

Program 6

Nama File : edit_berita.php
Deskripsi : Program edit berita.

```

1 <?php
2 include "koneksi.php";
3
4 if (isset($_GET['id'])) {
5     $id_berita = $_GET['id'];
6 } else {
7     die ("Error. No Id Selected! ");
8 }
9
10 $query = "SELECT id_berita, id_kategori, judul, headline, isi,
11 pengirim, tanggal
12             FROM berita WHERE id_berita='$id_berita'";
13 $sql = mysql_query ($query);
14 $hasil = mysql_fetch_array ($sql);
15 $id_berita = $hasil['id_berita'];
16 $id_kategori = stripslashes ($hasil['id_kategori']);
17 $judul = stripslashes ($hasil['judul']);
18 $headline = stripslashes ($hasil['headline']);
19 $isi = stripslashes ($hasil['isi']);
20 $pengirim = stripslashes ($hasil['pengirim']);
21 $tanggal = stripslashes ($hasil['tanggal']);
22
23 //proses edit berita
24 if (isset($_POST['Edit'])) {
25     $id_berita = $_POST['hidberita'];
26     $judul = addslashes (strip_tags ($_POST['judul']));
27     $kategori = $_POST['kategori'];
28     $headline = addslashes (strip_tags ($_POST['headline']));
29     $isi_berita = addslashes (strip_tags ($_POST['isi']));
30     $pengirim = addslashes (strip_tags ($_POST['pengirim']));
31     //update berita
32     $query = "UPDATE berita SET
33         id_kategori='$kategori', judul='$judul', headline='$headline',
34         isi='$isi_berita', pengirim='$pengirim' WHERE
35         id_berita=$id_berita";
36     $sql = mysql_query ($query);
37     if ($sql) {
38         echo "<h2><font color=blue>Berita telah berhasil diedit</font></h2>";
39     } else {
40         echo "<h2><font color=red>Berita gagal diedit</font></h2>";

```

```

39     }
40   }
41 ?>
42 <html>
43   <head><title>Edit Berita</title>
44   <link rel="stylesheet" href="style.css">
45   </head>
46   <body>
47     <a href="index.php">Halaman Depan</a> |
48     <a href="arsip_berita.php">Arsip Berita</a> |
49     <a href="input_berita.php">Input Berita</a>
50     <br><br>
51     <FORM ACTION="" METHOD="POST" NAME="input">
52     <table cellpadding="0" cellspacing="0" border="0" width="700">

53       <tr>
54         <td colspan="2"><h2>Input Berita</h2></td>
55       </tr>
56       <tr>
57         <td width="200">Judul Berita</td>
58         <td><input type="text" name="judul" size="30" value=<? echo $judul ?>"></td>
59       </tr>
60       <tr>
61         <td>Kategori</td>
62         <td>
63           <select name="kategori">
64             <?
65               $query = "SELECT id_kategori,
nm_kategori FROM kategori ORDER BY nm_kategori";
66               $sql = mysql_query ($query);
67               while ($hasil = mysql_fetch_array
($sql)) {
68                 $selected = ($hasil['id_kategori']==
69 $id_kategori) ? "selected" : "";
70                 echo "<option
value='$hasil[id_kategori]'$selected>$hasil[nm_kategori]</option>";
71               }
72             ?>
73           </select></td>
74         </tr>
75         <tr>
76           <td>Headline Berita</td>
77           <td><textarea name="headline" cols="50"
rows="4"><?=$headline?></textarea></td>
78         </tr>
79         <tr>
80           <td>Isi Berita</td>
81           <td><textarea name="isi" cols="50"
rows="10"><?=$isi?></textarea></td>
82         </tr>
83         <tr>
84           <td>Pengirim</td>
85           <td><input type="text" name="pengirim"
size="20" value=<?=$pengirim?>"></td>
86         </tr>
87         <tr>
88           <td>&nbsp;</td>
89           <td>&nbsp;&nbsp;
<input type="hidden" name="hidberita"
value=<?=$id_berita?>">
90           <input type="submit" name="Edit" value="Edit
Berita">&nbsp;
91           <input type="reset" name="reset"
value="Cancel"></td>
92         </tr>
93       </table>
94     </FORM>

```

```
95   </body>
96 </html>
```

Membuat Halaman Delete Berita

Program 7

Nama File : delete_berita.php
Deskripsi : Program untuk menghapus berita.

```
1  <?php
2  include "koneksi.php";
3
4  if (isset($_GET['id'])) {
5      $id_berita = $_GET['id'];
6  } else {
7      die ("Error. No Id Selected! ");
8 }
9 ?>
10 <html>
11     <head><title>Delete Berita</title>
12     <link rel="stylesheet" href="style.css">
13     </head>
14     <body>
15         <a href="index.php">Halaman Depan</a> |
16         <a href="arsip_berita.php">Arsip Berita</a> |
17         <a href="input_berita.php">Input Berita</a>
18         <br><br>
19         <?
20         //proses delete berita
21         if (!empty($id_berita) && $id_berita != "") {
22
23             $query = "DELETE FROM berita WHERE
24                 id_berita='$id_berita'";
25             $sql = mysql_query ($query);
26             if ($sql) {
27                 echo "<h2><font color=blue>Berita telah berhasil
28 dihapus</font></h2>";
29             } else {
30                 echo "<h2><font color=red>Berita gagal
31 dihapus</font></h2>";
32             }
33         }
34     ?>
35     </body>
36 </html>
```

CSS sebagai Pemanis Tampilan

Program 8

Nama File : style.css
Deskripsi : File CSS sebagai pemanis tampilan.

```
1 body {
2     font-family:verdana;
3     font-size:11px;
4 }
5 a {
6     color:#0000ff;
```

```
7     text-decoration:underline;
8 }
9 a:hover {
10    color:#ffffff;
11    background-color:#8AC452;
12    text-decoration:none;
13 }
14 input, textarea, select, option {
15    font-family:verdana;
16 }
```