rio astamal

DASAR-DASAR

WEB PROGRAMMING



Lisensi dibawah: Creative Commons Attribution-Share Alike 3.0



http://creativecommons.org













http://rioastamal.net/

http://lug.stikom.edu/





Versi 1.0-alpha2 Februari 2008

©Copyright 2008-2009 rio astamal (c0kr3x@gmail.com)

Lisensi dibawah: Creative Commons Attribution-Share Alike 3.0



http://creativecommons.org



Linux User Group STIKOMP Surabaya Pendahuluan

UNDER CONSTRUCTION....

BAB I HTML dan World Wide Web

1.1 Cara Kerja WWW

Ketika kita mengunjungi sebuah situs katakanlah, google.com, ada sebuah proses "dibalik layar" yang tidak anda ketahui. Proses tersebut cukup kompleks namun, jika digambarkan dalam sebuah diagram sederhana kurang lebih akan seperti berikut:



Secara kronologis urutan proses pada diagram diatas adalah:

- 1. User membuka alamat website pada browser
- 2. Browser mengirim HTTP-Request ke server
- 3. Server merespon HTTP-Request dari browser
- 4. Server mengirim HTTP-Response ke browser
- 5. Browser menampilkan halaman website kepada user

1.2 HTML

HTML/XHTML merupakan format tampilan yang digunakan untuk menampilkan halaman website. HTML terdiri dari simbol-simbol tertentu yang sering disebut dengan **tag**. Sebuah halaman website yang valid selalu diapit tag <html></html>. File HTML umumnya memiliki akhiran *.htm atau *.html.

Tag-tag pada HTML selalu diawali dengan <x>...</x>, dimana x tag HTML seperti , , <div>, dan lain-lain. Dibawah ini adalah tabel beberapa tag HTML yang sering digunakan.

Tag HTML	Keterangan
<html></html>	Tag untuk mengapit halaman HTML
<head></head>	Tag yang berisi informasi umum dari halaman
<title></title>	Judul Halaman *
<body></body>	Akan ditampilkan di browser
<style></style>	Untuk CSS *
	Untuk menebalkan teks
<div></div>	Untuk membuat layer
<a>	Untuk membuat hyperlink

	Untuk membuat paragraf
<hn></hn>	Untuk membuat header **
	Untuk inline style (manipulasi teks)
	Komentar

* Tag tersebut harus berada didalam tag <head>...</head>

** n dapat berupa angka dari 1 – 5, contoh <h1>...</h1>

Penerapan Teori HTML

- 1. Buka editor anda (gedit(Linux), Notepad++(Windows), dsb))
- 2. Ketik kode berikut:

```
1. <html>
2. <head>
3. <title>Website Pertamaku</title>
4. </head>
5. <body>
6. <h1>Hello F**king World!!</h1>
7. </body>
8. </html>
```

- 3. Simpan pada dengan nama hello-world.html
- 4. Buka browser anda(Firefox, IE, dsb) dan buka file yang baru saja anda buat.
- 5. Hasilnya akan seperti gambar 1.1



Gambar 1.1: hello-world.html

1.3 Manipulasi Font

Untuk memanipulasi font kita akan menggunakan **inline style/CSS** (akan dibahas dibagian berikutnya). Kita tidak akan menggunakan tag karena tag ini sudah "kuno" alias deprecated. W3C sebagai pengembang HTML sudah tidak menyarankan untuk menggunakan tag font lagi.

Sebagai gantinya digunakan tag dan inline style. Inline style adalah attribut style yang diberikan pada sebuah tag HTML. Contoh, untuk membuat tampilan teks tebal gunakan cara berikut:

```
<span style="font-weight:bold">Aku adalah teks tebal</span>
```

Output:

Aku adalah teks tebal

Pada contoh diatas kita memberikan attribut style, yang berisi "font-weight:bold". Maksudnya adalah kita akan mengaplikasikan style bold pada teks yang diapit oleh tag . Pemberian style tidak hanya terbatas pada satu bagian saja. anda dapat memberikan beberapa style sekaligus dengan memberi tanda pemisah berupa ";" untuk setiap style.

Aku
adalah teks tebal, bergaris bawah dan miring

Output

Aku adalah teks tebal, bergaris bawah, dan miring

Pada contoh diatas kita menerapkan tiga style pada tag span. Dapat dilihat kalau setiap style dipisahkan oleh titik koma. Intinya kita dapat memberikan banyak style sekaligus pada suatu tag.

Attribut style tidak terbatas pada tag span saja, hampir semua tag yang digunakan untuk presentasi teks dapat kita sisipi tag style.

Penerapan Teori Manipulasi Teks

- 1. Buka gedit/Notepad++
- 2. Ketik kode HTML berikut ini

```
1. <html>
2. <head>
3.
      <title>Manipulasi Teks</title>
4. </head>
5. <body>
6.
      <h1 style="color:red;text-decoration:underline">PENGUMUMAN!</h1>
7.
      <span style="color:#FF0000">
      Barang siapa yang menemukan dompet saya, akan saya beri
8.
      <span style="font-weight:bold;font-style:italic">ISTRI, MOBIL, RUMAH dan
9.
10.
      Segala SERTIFIKAT TANAH SAYA</span> DIJAMIN!!!.</span>
11.</body>
12.</html>
```

- 3. Simpan dengan nama manipulasi-teks.html
- 4. Buka dengan browser anda
- 5. Hasilnya akan seperti gambar 1.2



Gambar 1.2

1.4 Hyperlink

Anda tentu sering melngklik gambar atau tulisan yang mengarah ke halaman lain atau website lainnya. Yang anda klik itu adalah *hyperlink* atau lebih dikenal dengan istilah *link* saja. Untuk membuat link kita dapat menggunakan tag <a>. Dua attribut yang paling sering digunakan pada tag <a> adalah **href** dan **target**.

href digunakan untuk menentukan lokasi tujuan dari link. Dapat berupa relatif URL atau absolut URL.

target digunakan untuk menentukan apakah halaman akan dibuka pada jendela/window baru atau tidak. Defaultnya adalah *self* dimana link tidak dibuka pada window baru. Jika ingin membuka pada window baru gunakan nilai *_blank*.

Untuk lebih memahami cara kerja link, kita akan membuat dua file yaitu **profilku.html** dan **daftarsitus.html**.

- 1. Buka gedit/Notepad++
- 2. Ketik kode berikut:

```
1. <html>
2. <head>
3.
     <title>Profil JSL</title>
4. </head>
5. <body>
6.
     <h2>ProfilKu</h2>
7.
     8.
        <strong>Nama Lengkap:</strong> John Septian Lennon
9.
        <strong>Tempat/Tgl. Lahir: </strong>Lipermboh, 14 September 1940
10.
        <strong>Pekerjaan:</strong> Rocker Jalanan
11.
        <strong>Wafat: </strong> Tewas tertembak oleh penggemar di depan
12.
        kandang sapi pada 1980
13.
     14.
     Ingin lihat daftar situs favorit saya? <a href="daftar-situs.html">
15.
     klik disini</a>.
16.</body>
17.</html>
```

3. Simpan dengan nama **profilku.html**

4. Buat file baru lagi, kemudian ketik kode berikut:

```
1. <html>
2. <head>
3.
     <title>Daftar situs favorit JSL</title>
4. </head>
5. <body>
     <h2>Daftar Situs favoritku</h2>
6.
7.
     <01>
8.
        <a target=" blank" href="http://www.google.com/">Om Google</a>
        <a target="blank" href="http://www.milw0rm.com">
9.
10.
        Cacing Underground</a>
11.
     12.
     <a href="profilku.html">Lihat profil lengkap</a>
13.</body>
14.</html>
```

- 5. Simpan dilokasi yang sama dengan profilku.html dan beri nama daftar-situs.html
- 6. Buka file profilku.html dengan browser anda, dan coba klik link yang ada untuk memahami cara kerjanya



Gambar 1.3

1.5 Membuat Tabel

Untuk membuat tabel kita dapat menggunakan tag . Dalam beberapa tahun terakhir, penggunaan tabel sebagai layout website sudah tidak relevan lagi, dan digantikan oleh tag DIV. Saat ini penggunaan tabel dikhususkan hanya untuk menampilkan data secara tabular saja tidak untuk design halaman.

Beberapa tag yang sering digunakan saat membuat tabel adalah:

Tag	Keterangan
	Digunakan untuk membuat baris baru
*	Digunakan untuk membuat header tabel
*	Digunakan untuk membuat kolom

* Tag tersebut selalu berada didalam tag ...

Seperti halnya tag lain yang digunakan untuk presentasi teks, seperti **, anda dapat menggunakan attribut *style* untuk memformat tampilan tampilan tabel. Berikut ini adalah contoh

pembuatan tabel menggunakan HTML.

- 1. Buka teks editor (gedit/Notepad++)
- 2. Ketik kode berikut:

```
1. <html>
2. <head>
3.
  <title>TABEL IPS/IPK</title>
4. </head>
5. <body>
6.
  <h2>Tabel IP Sampai semester 4:</h2>
7.
   8.
    9.
      >SEMESTER
10.
      IPS
11.
    12.
    13.
      SEMESTER 1style="text-align:right">4.0
    14.
15.
    16.
      SEMESTER 2style="text-align:right">4.0
17.
    18.
    19.
      SEMESTER 3style="text-align:right">4.0
20.
    21.
    22.
      SEMESTER 4style="text-align:right">4.0
23.
    24.
    25.
      NILAI IPK >>>
26.
      4.0
27.
    28.
   29.</body>
30.</html>
```

- 3. Simpan denan nama *tabel.html*
- 4. Buka dengan browser anda, hasilnya akan nampak seperti gambar 1.4

Eile Edit View History Bookmarks Tools Help del.icio.us TABEL IPS/IPK - Mozilla Firefox			
< • 🔶 • 💽 🛞 🏠 📑 🙀 Diffile:///home/astadev/webpro/bab1/tabel.html 🔹 🕨 💽 • Google			
Tabel IP Sampai semester 4:			
SEMESTER	IPS		
SEMESTER 1	4.0		
SEMESTER 2	4.0		
SEMESTER 3	4.0		
SEMESTER 4	4.0		
NILAI IPK >>>	4.0		
Done			

Gambar 1.4

> merupakan special character untuk tanda ">". Dalam HTML special character diawali tanda ampersand "&" dan diakhiri ";". Contoh lain adalah < untuk "<", © untuk "©"

1.6 HTML Form

HTML Form sering digunakan untuk meminta inputan dari user yang kemudian diproses oleh server side scripting seperti PHP, JSP, dan sebagainya. Form HTML diapit oleh tag <form></form>. Dua attribut yang paling sering digunakan dalam penggunaan form adalah:

- Action: Nilai dari attribut ini menunjukkan lokasi dari file pemroses dari form. Dapat berupa relatif URI, contoh "folderX/file.php" atau absolut URI, contoh "http://contoh.com/file.php".
- **Method:** Nilai dari attribut menentukan metode data yang dikirimkan ke file pemroses, apakah melalui metode "*GET*" atau "*POST*".

Contoh penggunaan tag form sebagai berikut:

```
<form action="file.php" method="POST">
...
</form>
```

Tag form tidak banyak berguna tanpa adanya tag input lain. Tag form berfungsi hanya sebagai *wrapper* yang mengelompokkan data yang akan dikirim. Tag yang digunakan bersama tag form adalah tag *<input>*.

1.6.1 Macam-macam input

Untuk menampilkan textbox, checkbox, radio button, dan lainnya hanya diperlukan tag *<input>*. Yang membedakan output dari masing-masing tampilan adalah nilai dari attribut *type*. Contoh berikut merupakan penggunaan tag input untuk menampilkan textbox:

```
<input type="text" size="16" maxlength="16">
```

Nilai type	Keterangan
text	Untuk menampilkan textbox
password	Untuk menampilkan password field
file	Untuk menampilkan proses upload file (mirip seperti textbox namun dengan tombol Browse)
checkbox	Untuk menampilkan tombol checkbox (lebih dari satu pilihan)
radio	Untuk menampilkan tombol radio/option (hanya satu pilihan)
submit	Tombol untuk men-submit form (default button untuk submit)
button	sama dengan submit hanya saja bukan default button
reset	Untuk membersihkan tampilkan form

Berikut ini daftar nilai yang dapat digunakan pada attribut type:

hidden	Input tidak ditampilkan dibrowser.
--------	------------------------------------

Selain tag <input> masih ada tag lain yang biasa digunakan dalam form yaitu tag <*select*> dan <*textarea*>.

PENERARAN TEORI HTML FORM

- 1. Buka gedit / Notedpad++
- 2. Ketik kode berikut:

```
1. <html>
2. <head>
3.
    <title>Registrasi</title>
4. </head>
5. <body>
6.
     <h2>Form Registrasi</h2>
7.
      <form action="" method="POST">
8.
         <label>Nama: </label>
9.
         <input type="text" size="30" name="nama"><br>
10.
         <label>Username: </label>
11.
         <input type="text" size="16" name="uname" maxlength="16"><br>
12.
         <label>Password: </label>
13.
         <input type="password" size="16" name="pass" maxlength="16"><br>
14.
         <label>Jenis Kelamin: </label>
15.
         <input type="radio" name="jk" value="pria" checked><span> Pria</span>
16.
         <input type="radio" name="jk" value="wanita"><span> Wanita</span>
17.
         <br>
18.
         <label>Hoby: </label><br>
19.
         <input type="checkbox" name="hob" value="spkbola"><span> Sepak Bola</span>
20.
         <input type="checkbox" name="hob" value="game"><span> Game</span>
21.
         <input type="checkbox" name="hob" value="tidur"><span> Tidur</span><br>
22.
         <label>Deskripsikan diri anda: <label><br>
23.
         <textarea style="height:100px;width:400px" name="desc"></textarea>
24.
         <hr><br>>
25.
         <label>Darimana anda mendengar kami?</label><br>
26.
         <select name="dengar">
27.
            <option value="kuburan">Dari dalam kubur</option>
28.
            <option value="google">Google</option>
29.
            <option value="mimpi">Mimpi</option>
30.
         </select><br>
31.
         <input type="submit" value="DAFTAR">
32.
         <input type="reset" value="RESET">
      </form>
33.
34.</body>
35.</html>
```

3. Simpan dengan nama *form.html*

4. Buka dengan browser file tersebut. Hasilnya memang tidak begitu bagus karena kita tidak melakukan *styling* pada form.

Gambar 1.6		
* * • • Registrasi • Mozilla Firefox •		
<u>E</u> ile <u>E</u> dit <u>V</u> iew Hi <u>s</u> tory <u>B</u> ookmarks <u>T</u> ools <u>H</u> elp del <u>.</u> icio.us		
Form Registrasi		
Nama: John Septian Lennon Username: john Password: ****** Jenis Kelamin: Pria O Wanita Hoby: Sepak Bola I Game I Tidur Deskripsikan diri anda: Wah saya sendiri bingung saya ini seperti apa ya???		
Darimana anda mendengar kami? Dari dalam kubur ▼ DAFTAR RESET		
Done		

Dalam kasus *real world* nilai yang ada pada attribut *name* dan *value*-lah yang akan dikirim ke file pemroses.

1.7 Menampilkan Gambar

Untuk menampilkan gambar pada halaman HTML dapat digunakan tag **. Beberapa attribut yang sering digunakan adalah *src, height,* dan *width*. Attribut *src* digunakan untuk menentukan alamat dari gambar yang akan ditampilkan, dapat berupa relatif URI atau absolut URI. Attribut *height* digunakan untuk menentukan tinggi, sedangkan *width* untuk menentukan lebar.

Secara default nilai yang ada pada height dan width adalah dalam pixel, kecuali anda menambahkan tanda "%". Maka lebar atau tinggi diukur menggunakan persen. Jika anda tidak menyertakan attribut height dan width, maka gambar tersebut akan ditampilkan sesuai dengan ukuran aslinya. Untuk lebih memahaminya ikuti langkah berikut:

- 1. Siapkan dua buah gambar dalam folder yang sama dengan file html
- 2. Dalam contoh ini saya menggunakan (1)stikom.jpg dan (2)join-revolution.jpg
- 3. Buka gedit/Notepad++
- 4. Ketik kode berikut:

```
1. <html>
2. <head>
3. <title>Join The Revolution</title>
4. </head>
5. <body>
6. <img src="stikom.jpg" align="left">
7. <strong style="color:red">SHOULD</strong>
8. <img src="join-revolution.jpg" align="center">
```

9. </body> 10.</html>

Join The Revolution - Mozilla Firefox

5. Simpan dengan nama gambar.html

Gambar 1.7

1.8 XHTML dan DOCTYPE

Apakah XHTML itu? apakah ia berbeda dengan HTML? Sebenarnya XHTML adalah HTML hanya saja XHTML mengikuti format XML sebagai standar penulisannya. Jadi setiap tag harus memiliki penutup.

Anda tentu pernah mengetik tag-tag tanpa penutup seperti
br>, , dan <input> pada pembahasan sebelumnya. Pada XHTML semua tag harus memiliki penutup, jika tidak maka halaman yang anda buat tidak sesuai dengan standar yang telah ditentukan dan dianggap tidak valid/ compliant dengan standar W3C.

Berikut ini adalah beberapa contoh penulisan yang valid di HTML tapi tidak di XHTML.

SALAH	BENAR
<pre>Hello World</pre>	Hello World
<input checked="" type="radio"/>	<input checked="checked" type="radio"/>
	 atau
	
<input type="button" value="GO >>>"/>	<input <br="" type="button"/> value="GO >>>" />
Hello	Hello

Mungkin anda bertanya, lalu bagaimana saya memberitahu browser bahwa halaman saya adalah XHTML? Semua itu terletak pada DOCTYPE halaman anda. DOCTYPE adalah pententu tipe

document yang ingin anda gunakan apakah HTML 4.01, XHTML 1.0 Transitional, XHTML 1.0 Strict dan lainnya.

DOCTYPE juga sering disebut *DTD* (*Document Type Definition*). Berikut ini adalah contoh penggunaan tipe dokumen HTML 4.01,

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD HTML 4.01//EN" "http://www.w3c.org/TR/html4/
strict.dtd">
```

Jika anda tidak menyebutkan DTD yang anda gunakan asumsi dari satu browser dengan browser yang lain mungkin berbeda-beda. Jadi jika memang anda ingin menggunakan HTML murni bukan XHTML gunakanlah DTD 4.01. Deklarasi DOCTYPE harus diletakkan paling awal sebelum tag https://www.communication.com

Untuk XHTML beberapa DTD yang sering digunakan adalah

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
2. <! DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> 3. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

Perbedaan *Transitional* dan *Strict* terdapat pada toleransi kesalahan. Pada transitional jika masih ada kesalahan standar penulisan masih diberikan toleransi, sedangkan Strict tidak memberikan toleransi sama sekali. Contoh halaman XHTML yang valid adalah sebagai berikut:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/TR/xhtml1">
    <head>
        <title>Every document must have a title</title>
        </head>
        <body>
            <span>Hello World</span>
        </body>
</html>
```

Dalam pembahasan selanjutnya kita akan selalu menggunakan XHTML bukan lagi HTML.

Deklarasi versi XML pada awal halaman tidak harus ditulis namun sangat dianjurkan untuk ditulis. Attribut "xmlns" harus ditulis untuk menentukan namespace yang digunakan.

1.9 CSS (Cascading Style Sheets)

CSS merupakan suatu teknologi yang digunakan untuk mempermudah pembuatan sebuah website. Dengan CSS kita dapat dengan cepat mengaplikasikan suatu style pada tag tertentu. Bahkan kita dapat meletakkan CSS pada suatu file sehingga dapat digunakan oleh banyak halaman sekaligus. Itu merupakan salah satu kelebihan CSS. CSS diapit oleh tag <style></style> dan berada diantara tag <head></head>. Kelebihan lainnya adalah anda dapat menyisipkan komentar pada CSS, hal ini

cukup penting jika jumlah CSS anda sangat banyak. Komentar pada CSS diapit oleh karakter /* */.

Pada pembahasan-pembahasan sebelumnya kita sering menggunakan attribut style untuk memformat tampilan atau sering disebut *inline style*. Apa yang ada pada attribut style tersebut sebenarnya adalah CSS yang valid, hanya saja letaknya didalam tag.

Hampir semua tag dapat dimanipulasi menggunakan CSS seperti <body>, , <div>, , , dan masih banyak lagi. Berikut ini adalah contoh sederhana penggunaan CSS pada tag body.

body {font-face: Verdana; font-size: 11px }

Pada contoh diatas kita memformat semua teks yang ada pada tag body menjadi berjenis huruf Verdana dengan ukuran 11 pixel. Tag *body* pada contoh disebut *selector* sedangkan attribut style didalamnya disebut *declaration*. Declaration ditandai dengan adanya kurung kurawal {...}.

Penerapan Teori CSS

- 1. Buka gedit / Notepad++
- 2. Ketik kode berikut:

```
1. <?xml version="1.0" encoding="UTF-8" ?>
2. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3.
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4. <html xmlns="http://www.w3.org/TR/xhtml1">
5. <head>
6.
     <title>CSS - Cascading Style Sheets</title>
      <style type="text/css">
7.
8.
         body { font-family: Verdana, Serif; font-size: 11px }
9.
         div {
10.
            border: 2px solid #cccccc;
11.
            padding: 4px
12.
         }
13.
        h2 {
14.
            text-decoration: underline;
15.
            font-style: italic;
16.
            font-size: 18px
17.
         }
18.
    </style>
19.</head>
20.<body>
21.
    <div>
22.
         <h2>Ini adalah contoh sederhana penggunaan CSS</h2>
23.
    </div>
24.</body>
25.</html>
```

- 3. Simpan dengan nama *css.html*
- 4. Lalu jalankan pada browser, hasilnya akan terlihat seperti gambar 1.8

CSS - Cascading Style Sheet - Mozilla Firefox	
<u>F</u> ile <u>E</u> dit <u>V</u> iew Hi <u>s</u> tory <u>B</u> ookmarks <u>T</u> ools <u>H</u> elp del <u>.</u> icio.us	$\langle \rangle$
🔄 🗸 🛶 🗸 🥑 🐼 🚮 📑 🚗 🗋 file:///home/astadev/webpro/bab1/css.html 🔻 🕨 💽 Google	Q
Ini adalah contoh sederhana penggunaan CSS	
Done	

Gambar 1.8: Contoh penggunaan CSS

Dapat anda lihat bahwa tag yang kita deklarasikan pada CSS secara otomatis style tag tersebut mengikuti aturan CSS yang kita buat. Dengan demikian kita dapat dengan mudah memodifikasi banyak elemen sekaligus hanya dari sebuah baris CSS.

1.9.1 CSS Class dan ID

Jika anda cukup kreatif, saya harap memang demikian :) maka anda mungkin berfikir kalau penggunaan cara seperti pembahasan sebelumnya tidak sepenuhnya baik. Mengapa? Bayangkan jika pada beberapa bagian pada halaman, anda tidak ingin mengaplikasikan style tersebut. Lalu apa yang anda lakukan? Merubahnya secara manual lewat inline style? Itu memang dapat dilakukan tapi tidak efisien.

Cara yang paling efektif dan efisien adalah dengan menggunakan *class* dalam CSS. Dengan menggunakan class kita dapat menentukan letak bagian yang harus kita aplikasikan suatu style. Penggunakan class diawali dengan tanda titik ".". Lihat contoh berikut:

```
.tebal-miring { font-weight: bold; font-style: italic }
div.error { color: red; font-weight: bold }
```

Baris pertama merupakan *general class* karena semua tag dapat menggunakannya. Sedangkan pada baris kedua adalah *regular class* karena class tersebut hanya berlaku pada tag div saja. Untuk mengaplikasikan style yang ada pada class kita harus memasukkan nama class tersebut pada attribut *class*.

Cara lain adalah dengan menggunakan nilai dari attribut **id** pada setiap tag. Nilai dari attribut id antara tag yang satu dengan tag yang lain tidak boleh ada yang sama. Tanda yang digunakan bukan titik melainkan tanda pagar "#".

```
#main { border: 1px solid #000000 }
div#header { padding: 4px }
```

OK, mari kita buat sebuah file untuk lebih memahaminya.

- 1. Buka gedit / Notepad++
- 2. Ketik kode berikut:

```
1. <?xml version="1.0" encoding="UTF-8" ?>
2. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
4. <html xmlns="http://www.w3.org/TR/xhtml1">
5. <head>
6.
     <title>CSS Class & amp; ID</title>
7.
     <style type="text/css">
8.
        body { font-family: Verdana, Serif; font-size: 11px }
9.
        #main {
10.
           border: 1px solid #cccccc;
11.
           padding: 4px
12.
        }
13.
        div.box {
14.
           border: 1px solid orange;
15.
           background-color: #FFF8B9;
16.
           padding: 6px 3px;
           color: #c60000;
17.
18.
           width: 200px
19.
        }
20.
        .green-force {
21.
           font-weight: bold;
22.
           color: green
23.
        }
24.
    </style>
25.</head>
26.<body>
27. <div id="main">
28.
        <div class="box">
29.
           I'm the WalRUS...!!<br />GOO GOO G'JOOB!!!
30.
        </div>
31.
        <span class="green-force">Class green-force pada tag span</span>
32.
        Class green-force pada tag p
33.
     </div>
34.</body>
35.</html>
```

- 3. Simpan dengan nama css-class.html
- 4. Hasilnya akan tampak seperti gambar 1.9 jika anda jalankan pada browser

🗴 🔸 🗕 CSS Class & ID - Mozilla Firefox	-
<u>F</u> ile <u>E</u> dit ⊻iew Hi <u>s</u> tory <u>B</u> ookmarks <u>T</u> ools <u>H</u> elp del <u>.</u> icio.us	$\langle \rangle$
🗢 🗣 👻 🚱 🚮 F 🙀 โ file:///home/astadev/webpro/bab1/css-class.html 🔻 🕨 💽 Google	Q
I'm the WalRUS!! GOO GOO G'JOOB!!!	
Class green-force pada tag span	
Class green-force pada tag p	
Done	

Gambar 1.9

1.9.2 Hyperlink Pseudoclass

Jika anda pernah mengunjungi suatu website dan anda menggerakkan mouse anda ke arah sebuah link lalu link tersebut berubah warna dan menjadi bergaris bawah atau sebaliknya. Apa yang anda lihat tersebut adalah hasil dari *hyperlink pseudoclass*-nya CSS.

Seperti yang sudah dibawas sebelumnya untuk membuat link kita menggunakan tag <a>. Oleh

karena itu tag iniliah yang akan kita manipulasi. Format penulisan hyperlink pseudoclass adalah a:nama_aksi, dimana nama_aksi dapat berupa:

- *link*: Tampilan link ketika tidak dipilih oleh user (biasanya tidak perlu ditulis)
- *visited*: Tampilan link setelah diklik
- active: Tampilan ketika link diklik
- *hover*: Tampilan ketika mouse melewati link (lebih sering digunakan daripada *active*)

Penasaran? langsung saja buat file untuk mencobanya.

- 1. Buka gedit / Notepad++
- 2. Ketik kode berikut:

```
1. <?xml version="1.0" encoding="UTF-8" ?>
2. <! DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3.
4. <html xmlns="http://www.w3.org/TR/xhtml1">
5. <head>
     <title>Hyperlink Pseudoclass</title>
6
7.
     <style type="text/css">
8.
         body { font-family: Verdana, Serif; font-size: 14px }
9.
         a:link {
10.
            color: green;
11.
            text-decoration: underline
12.
         }
13.
         a:visited {
14.
           color: green;
15.
            text-decoration: underline
16.
         }
17.
         a:active {
18.
           font-weight: bold;
19.
            text-decoration: none;
20.
            color: #c60000
21.
         }
22.
         a:hover {
23.
            font-style: italic;
24.
            font-weight: bold;
25.
            color: #c60000;
26.
            text-decoration: none
27.
         }
28.
     </style>
29.</head>
30.<body>
31.
    <div>
32.
       <a href="#">Link 1</a><br />
33.
        <a href="#">Link 2</a><br />
34.
        <a href="#">Link 3</a>
35.
    </div>
36.</body>
37.</html>
```

- 3. Simpan dengan nama *pseudoclass.html*
- 4. Buka browser and a lalu buka file tersebut

🔹 🔹 - Hyperlink Pseudoclass - Mozilla Firefox	-
<u>F</u> ile <u>E</u> dit <u>V</u> iew Hi <u>s</u> tory <u>B</u> ookmarks <u>T</u> ools <u>H</u> elp del <u>.</u> icio.us	$\langle \rangle$
ሩ • 🔷 • 🞯 🕼 📑 🚗 🗋 file:///home/astadev/webpro/bab1/pseudoclass.html 🔹 🕨 💽 • Google	Q
Link 1 Link 2 Link 3	
Done	
Gambar: 1.10	

1.9.3 External CSS

Sampai saat ini anda sudah tahu bagaimana mudahnya memodifikasi style dengan menggunakan CSS. Agar dalam pembuatan website kita lebih efisien maka sebaiknya kita meletakkan file CSS kita pada suatu file tersendiri. Sehingga kita dapat meload-nya pada halaman yang membutuhkan stye yang ada pada file CSS tersebut.

File CSS yang akan kita load tidak harus berada satu komputer dengan file HTML kita. File CSS tersebut dapat berada pada website lain, ini dimungkinkan karena kita dapat menggunakan URL pada saat pemanggilan file.

Ada dua metode untuk memanggil file CSS, yang pertama adalah dengan menggunakan tag *<link>* dan yang kedua adalah menggunakan statement @*import* didalam CSS. Cara yang lebih sering digunakan adalah menggunakan tag *<link>*. Tidak ada yang lebih jelas daripada *learning by doing*, karena itu langsung saja kita praktikkan.

```
1. Buka gedit / Notepad++
     2. Ketik kode berikut:
1. body { font-family: Verdana, Serif; font-size: 14px }
2. #main {
3.
     border: 1px solid #cccccc;
4.
     padding: 4px
5. }
6. div.error {
7.
     border: 1px solid #c60000;
8.
     border-left: 5px solid #c60000;
9.
     color: #c60000;
10. font-size: 14px;
11. padding: 4px 6px;
12. font-weight: bold;
13.
     width: 350px;
14.}
     3. Simpan dengan nama my.css
     4. Buat file baru lagi, kemudian ketik kode berikut:
1. <?xml version="1.0" encoding="UTF-8" ?>
2. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4. <html xmlns="http://www.w3.org/TR/xhtml1">
5. <head>
6.
    <title>External Stylesheet</title>
```

```
7.
      <link rel="stylesheet" type="text/css" href="my.css" />
8. </head>
9. <body>
10. <div id="main">
11.
        <h2>External Stylesheet/CSS</h2>
12.
         <div class="error">
13.
            WARNING!, memory otak anda sudah 90% penuh silahkan kosongkan pikiran-
14.
            pikiran kotor yang tidak diperlukan...!!!
15.
         </div>
16.
     </div>
17.</body>
18.</html>
```

- 5. Simpan dengan nama *external-css.html*
- 6. Jalankan pada browser, hasilnya akan terlihat seperti gambar 1.11

External Stylesheet - Mozilla Firefox File Edit ⊻iew History Bookmarks Tools Help del,icio.us	-
<ri> </ri>	Q
External Stylesheet/CSS WARNING!, memory otak anda sudah 90% penuh silahkan kosongkan pikiran- pikiran kotor yang tidak diperlukan!!!	
Done	

Gambar 1.11

1.10 Membuat Layout dengan DIV

Seperti yang sudah disinggung pada pembahasan tentang tabel, kalau pembuatan design website yang menggunakan tabel dapat dikatakan sudah tidak relevan lagi. Sebagai pengganti dari tabel adalah tag div. Tag div memang dikhususkan untuk membagi halaman kedalam beberapa segmen. Jika kita gabungkan dengan penggunakan attribut id dan class, maka div dapat kita gunakan untuk layout sebuah halaman.

Secara kasar setiap halaman yang "baik" pasti memiliki beberapa bagaian, misalnya halaman tersebut terdiri dari header, kemudian beberapa kolom untuk content utama, dan sebagainya. Semua itu terserah pada designernya.

Pada contoh kasus yang akan kita buat nantinya, kita akan membuat sebuah layout halaman yang terdiri dari:

- 1. Header utama halaman
- 2. 2 Kolom
- 3. 1 untuk content utama
- 4. 1 untuk berita atau lainnya
- 5. lebar halaman tidak lebih dari 800px (untuk berjaga-jaga, karena masih banyak user yang resolusi monitornya 800x600)
- 6. Footer halaman

1. Buka gedit / Notepad++

Design halaman yang akan dibuat tidak melibatkan penggunaan gambar, karena prioritas kita disini hanyalah bagaimana cara mengatur tata letak komponen menggunakan tag div. OK, langsung saja tanpa banyak "mendrible bola", lho kok?

```
2. Ketik kode berikut:
1. * { padding: 0; margin: 0 }
2. body \{
3.
      font-family: Verdana, Arial, Serif;
4.
      font-size: 11px;
5.
     color: #333;
6.
     background: #fafafa;
7. }
8. /* pembatas utama */
9. #container {
10. margin: 6px auto;
11. text-align: left;
12. clear: both;
13. background: #fff;
14.
    border:2px solid #666;
     width:778px;
15.
16.
    padding:0;
17.}
18.#header {
19. clear:both;
20.
     margin: 2px;
21.
     background: #FFEA8C;
22.
     border: 1px solid orange;
23.
     height: 95px;
24.
25.#header h1 {
26. font-size: 2em;
27.
     font-family: Arial;
28. color: #c60000;
29. margin: 14px 6px 4px 8px;
30.}
31.#menu {
32. clear: both;
33.
     padding: 0; margin: 0 0 25px 2px;
34.}
35.#menu ul {
36. float:left;
37. border: none;
38.
     list-style: none;
39.
     font: bold 14px Arial;
40.}
41.#menu ul li {
42.
     display: inline;
43.}
44.#menu ul li a {
45.
     padding-right: 16px;
46.
     border-bottom: 4px solid orange;
47.}
48.#menu ul li a:hover {
49. border-bottom: 4px solid #c60000;
50.}
51.#menu ul li a.aktif {
52. border-bottom: 4px solid #c60000;
```

```
53.}
54.#footer {
55.
     position: relative;
56.
     clear:both;
57. height: 40px;
58. border: none;
59.
    background: #cfcfcf;
60.
     color: #fff;
61.
    width: 100%;
62.
     text-align: center;
63.}
64. #footer span {
65.
     top: 10px;
66.
     position: absolute;
67.}
68.#content {
69. margin: 2px 2px 8px 2px;
70. float: left;
71.
     width: 485px;
72.
     border: 1px solid #ccc;
73.
     padding: 6px 10px;
74.}
75.#content h1 {
76.
     border-bottom: 2px dashed #ccc;
77.
     margin-bottom: 16px;
78.}
79.#side {
80.
     border: 1px solid #ccc;
81.
    float: right;
82. margin: 2px;
83. width: 250px;
84.
     padding: 2px;
85.}
86.#side h1 {
87.
    border-bottom: 1px solid #fafafa;
88.}
89.#side h2 {
90. background: #cfcfcf;
91. padding: 3px;
92.
     color: #333;
93.
    text-align: center;
94.}
95.#side p {
96. border: 1px solid #ccc;
97.
     padding: 4px;
98.}
99.a { text-decoration:none; color: #666}
100.a:visited { color: #666 }
     5. Simpan dengan nama layout.css
```

6. Buat file baru, lanjutkan dengan mengetik kode berikut:

```
1. <?xml version="1.0" encoding="UTF-8" ?>
2. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4. <html xmlns="http://www.w3.org/TR/xhtml1">
5. <head>
6. <title>Layout Menggunakan DIV</title>
7. <link rel="stylesheet" type="text/css" href="layout.css" />
```

```
8. </head>
9. <body>
10. <div id="container">
11.
        <div id="header">
12.
           <h1>R.I.P John</h1>
13.
         </div>
        <div id="menu">
14.
15.
           16.
              <a class="aktif" href="#">Home</a>
17.
              a href="#">Books</a>
18.
              <a href="#">Search</a>
19.
              <a href="#">About</a>
20.
              <a href="#">Contact Us</a>
21.
            22.
        </div>
23.
        <div id="content">
24.
           <h1>Working Class Hero</h1>
25.
           As soon as you're born they make you feel small<br />
26.By giving you no time instead of it all<br />
27.Till the pain is so big you feel nothing at all<br />
28.A working class hero is something to be<br />
29.<br />
30. They hurt you at home and they hit you at school<br />
31. They hated your clever and despised a fool<br />
32.Till you're so fucking crazy you can't follow their rules<br />
33.A working class hero is something to be<br />
34.<br />
35.When they've tortured and scared you for twenty hard years<br/>br />
36. Then they expect you to pick a career <br />
37.When you can't really function you're so full of fear<br />
38.A working class hero is something to be<br />
39.<br />
40.Keep you doped with religion and sex and TV<br />
41.And you think you're so clever and classless and free<br />
42.But you're still fucking peasants as far as I can see<br />
43.A working class hero is something to be<br />
44.A working class hero is something to be<br />
45.<br />
46. There's room at the top they are telling you still<br />
47.But first you must learn how to smile as you kill<br />
48.If you want to be like all the folks on the hill<br />
49.<br />
50.If you want to be a hero well just follow me 
51.
        </div>
52.
        <div id="side">
53.
           <h2>IMAGINE...</h2>
54.
           Imagine there's no heaven
55.It's easy if you try
56.No hell below us
57.Above us only sky
58.Imagine all the people
59.Living for today...
60.Imagine there's no countries
61.It isn't hard to do
62.Nothing to kill or die for
63.And no religion too
64.Imagine all the people
65.Living life in peace...
66.You may say I'm a dreamer
67.But I'm not the only one
68.I hope someday you'll join us
```

<pre>Imagine no possessions I wonder if you can No need for greed or hunger A brotherhood of man Imagine all the people Sharing all the world You may say I'm a dreamer But I'm not the only one I hope someday you'll join us And the world will live as one And the world will live as one <th>nakan DIV - Mozilla Firefox ppro/bab1/layout.html# ▼ ► Cogle</th></pre>	nakan DIV - Mozilla Firefox ppro/bab1/layout.html# ▼ ► Cogle
R.I.P John	
R.I.P John Home Books Search About Contact Us	
R.I.P John Home Books Search About Contact Us Working Class Hero	IMAGINE
R.I.P John Home Books Search About Contact Us Working Class Hero As soon as you're born they make you feel small By giving you no time instead of it all Till the pain is so big you feel nothing at all A working class hero is something to be They hurt you at home and they hit you at school They hated your clever and despised a fool Till you're so fucking crazy you can't follow their rules A working class hero is something to be When they've tortured and scared you for twenty hard years Then they expect you to pick a career When you can't really function you're so full of fear A working class hero is something to be Keep you doped with religion and sex and TV And you think you're so clever and classless and free But you're still fucking peasants as far as I can see A working class hero is something to be There's room at the top they are telling you still But first you must learn how to smile as you kill If you want to be like all the folks on the hill If you want to be a hero well just follow me	IMAGINE Imagine there's no heaven It's easy if you try No hell below us Above us only sky Imagine all the people Living for today Imagine there's no countries It isn't hard to do Nothing to kill or die for And no religion too Imagine all the people Living life in peace You may say I'm a dreamer But I'm not the only one I hope someday you'll join us And the world will be as one Imagine no possessions I wonder if you can No need for greed or hunger A brotherhood of man Imagine all the people Sharing all the world You may say I'm a dreamer But I'm not the only one I hope someday you'll join us And the world will live as one

Gambar 1.12

1.11 Javascript

Javascript adalah sebuah bahasa pemrograman yang khusus dirancang untuk website. Javascript hanya berjalan disisi klien, artinya penggunaan javascript hanya terbatas pada web-browser anda saja, ia tidak bisa melakukan manipulasi data pada sisi server. Saat ini javascript merupakan salah satu element terpenting dari teknologi web terkini. AJAX adalah salah satu contoh penggunakan javascript yang saat ini banyak digunakan oleh website Web 2.0.

Namun pada buku ini saya hanya membahas javascript sepintas saja, untuk tahu lebih jauh tentang javascript anda dapat mencari lewat google atau tunggu buku saya berikutnya hehehe...

1.11.1 Menggunakan Javascript

Secara sederhana sebuah script javascript diapit oleh tag <script></script>, namun untuk lebih memperjelas penggunaan javascript biasanya ditambahkan attribut *language* atau *type*, contoh seperti berikut:

```
<script language="javascript">
   // javascript kode disini...
</script>
atau
<script type="text/javascript">
   // javascript kode disini...
</script>
```

Kenapa demikian? Karena jika anda pengguna browser IE maka ia memiliki satu bahasa lagi selain javascript yaitu vbscript. Namun pada kenyataannya hampir tidak ada situs yang menggunakan vbscript, jadi secara default pun jika anda hanya menggunakan tag <script></script> maka IE akan menganggap script tersebut adalah javascript.

Tag <script> dapat anda letakkan pada level global yaitu pada tag <head></head> atau secara lokal dimana pun dalam tag <body></body>. OK langsung saja kita coba "script é wong jowo" ini.

- 1. Buka gedit / Notepad++
- 2. Ketik kode berikut:

```
1. <?xml version="1.0" encoding="UTF-8" ?>
2. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3.
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4. <html xmlns="http://www.w3.org/TR/xhtml1">
5. <head>
6.
     <title>Menggunakan Javascript</title>
7.
      <script language="javascript">
8.
         var nama = prompt("Masukkan nama: ", "nama anda");
9.
         var tanya = confirm(nama+", apakah anda laki-laki?");
10.
         if (tanya == true) {
11.
           alert("Halo "+nama+" kamu adalah laki-laki!");
12.
         } else {
13.
            alert("Halooo "+nama+"... kamu cewek ya...? atau jangan-jangan...?");
14.
         }
15.
    </script>
16.</head>
17.<body>
```

18.</body> 19.</html>

- 3. Simpan dengan nama *javascript.html*
- 4. Jalankan pada browser untuk melihat hasilnya.

* [JavaScript Application]				
?	Masukkan nama:			
	Rio Astamal			
	😴 Cancel 🛛 🛃 OK			

Gambar 1.13: Javascript prompt

Pada contoh diatas kita meletakkan javascript pada level global dengan meletakkannya pada tag <head></head>. *prompt, confirm, dan alert* adalah fungsi-fungsi built in javascript yang dapat kita gunakan untuk berinteraksi dengan user.

Jika anda familiar dengan bahasa seperti Java, C++, C, atau anak cucunya maka anda tidak akan terlalu berlama-lama menyesuaikan diri dengan syntax javascript. Hal itu karena syntax javascript sangat mirip dengan ketiga bahasa yang telah disebutkan diatas. Untuk mendeklarasikan variabel pada javascript kita dapat menggunakan keyword *var*.

Javascript versi terbaru sudah mendukung penggunaan tipe data dalam pendeklarasian variabel seperti int, string, dll. Namun untuk menjaga kompatibilitas sebaiknya tetap gunakan keyword var.

1.11.2 Manipulasi DOM

Salah satu fungsi utama javascript adalah kemampuannya untuk memanipulasi DOM(*Document Object Model*). DOM sendiri adalah suatu struktur didalam dokumen HTML yang berbentuk seperti tree yang komponen-komponennya adalah elemen HTML seperti tag. Karena kemampuan inilah maka dengan javascript kita dapat menciptakan DHTML Effect (*Dynamic HTML Effect*).

Salah satu contoh pengaplikasian DHTML Effect adalah *hide and show* suatu elemen HTML. Misal, jika suatu tombol diklik maka tampilan hilang atau tampil. Semua ini dimungkinkan karena javascript dapat melakukan modifikasi style CSS pada elemen secara langsung atau para geeky lebih suka menyebutnya: *on the fly* :).

Untuk lebih memahaminya sebaiknya kita langsung saja mencobanya lewat contoh.

- 1. Buka gedit/Notepad++
- 2. Ketik kode berikut:

1. <?xml version="1.0" encoding="UTF-8" ?>

```
2. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4. <html xmlns="http://www.w3.org/TR/xhtml1">
5. <head>
6.
      <title>Manipulasi DOM</title>
7.
      <script language="javascript">
8.
         function beriWarna() {
9.
            var kotak = document.getElementById('kotak');
10.
            var warna = document.getElementById('warna');
11.
            // ganti CSS on the fly
12.
            kotak.style.backgroundColor = '#' + warna.value;
13.
         }
14.
         function hideShow() {
15.
            var kotak = document.getElementById('kotak');
16.
            var status = kotak.style.display;
17.
            // set CSS display ke block atau none
18.
            if (status == 'block') {
19.
               kotak.style.display = 'none';
20.
            } else {
               kotak.style.display = 'block';
21.
22.
            }
23.
         }
24. </script>
25.
      <style type="text/css">
26.
         #kotak {
27.
            border:3px solid #000;
28.
            padding: 4px;
29.
           background-color: #ccc;
30.
            font-size: 18px;
31.
            font-weight: bold;
32.
           width: 300px; height: 200px;
33.
         }
34.
     </style>
35.</head>
36.<body>
37.
    <form>
38.
         <label>Warna</label>
39.
         #<input type="text" id="warna" name="warna" size="5" value="ccc" />&nbsp;
40.
         <input type="button" onclick="beriWarna()" value="Ganti Warna" />&nbsp;
41.
         <input type="button" onclick="hideShow()" value="Hide or Show" />
42.
     </form>
43.
     <br />
44.
    <div id="kotak" style="display:block">
45.
         SAYA BUKAN DUKUN, PESULAP ATAU BAHKAN BUNGLON TAPI SAYA BISA BERUBAH
         WARNA DAN MENGHILANG. HEBATKAH SAYA? TENTU TIDAK. YANG HEBAT ADALAH
46.
47.
         YANG MENULIS SAYA BUKAN BEGITU? :p
48.
      </div>
49.</body>
50.</html>
```

- 3. Simpan dengan nama dom.html
- 4. Buka file menggunakan browser untuk melihat hasilnya.

Hasilnya kurang lebih akan sama dengan gambar 1.14.

Penjalasan Singkat File dom.html

Banyak hal baru menyangkut penggunaan javascript, seperti penggunaan keyword function dan

metode *getElementById*. Jika tidak mengerti untuk sementara telan saja :p karena buku ini memang tidak mengharuskan anda menguasai javascript.

Fungsi pada javascript ditandai dengan keyword *function*. Fungsi merupakan pengemlompokkan sejumlah perintah untuk dieksekusi pada satu waktu ketika fungsi itu dipanggil. Pada contoh kita membuat dua fungsi yaitu *beriWarna()* dan *hideShow()*. Object yang akan kita modifikasi stylenya adalah object dengan elemen id bernama "kotak" dalam hal ini elemen tersebut adalah sebuah div.

```
<div id="kotak" style="display:block">
```

Agar object div tersebut dapat kita manipulasi maka kita perlu membuatkan sebuah wadah untuk menampung object tersebut dalam javascript. Untuk itu digunakanlah salah satu fungsi DOM yaitu *getElementById*. Setelah kita mendapatkan object tersebut maka kita bebas untuk memodifikasinya. Contoh sederhana adalah mengubah stylesheet dari object tersebut. Setiap tag yang dapat dimanipulasi dengan CSS memiliki atribut **style**. Bedanya jika pada CSS format penulisannya jika terdiri dari dua kata maka dipisah dengan tanda "-" contoh: background-color, font-family, font-size, dan sebagainya. Tapi pada javascript digunakan model Capitalis, contoh: backgroundColor, fontFamily, fontSize dan sebagainya.

Warna yang digunakan pada contoh adalah menggunakan hexadecimal. Contoh-contoh warna yang dapat anda gunakan sebagai bahan ujicoba adalah: 90EE90, FFFF00, FFC0CB, dan masih banyak lagi.



Gambar 1.14: Manipulasi DOM

Ringkasan Bab I

Pada bab ini kita telah mempelajari dasar-dasar penggunaan HTML, bagaimana mengunakan tagtagnya, memanipulasi teks dengan CSS dan mendesain layout sederhana menggunakan CSS dan DIV. Hal yang perlu diingat adalah kita akan menggunakan standar XHTML jadi setiap tag pasti memiliki penutup. Oleh karena itu untuk pembahasan selanjutnya DOCTYPE yang akan kita pakai adalah XHTML Transitional.

Pada bab ini kita juga sedikit membahas tentang penggunakan javascript secara sederhana untuk penyampaian informasi ke user berupa dialog box dan input box. Selain itu kita juga sedikit membahas bagaimana memanipulasi DOM menggunakan javascript. Fungsi DOM tersebut adalah *getElementById*.

Bab II Pengenalan PHP

2.1 Apa Itu PHP?

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

PHP merupakan hasil kerja seorang bernama Rasmus Lerdorf pada 1995. Namun kemudian PHP berkembang dan tidak hanya merupakan proyek pribadi Rasmus. PHP ditulis ulang dan dengan banyak menambahkan fungsi-fungsi baru oleh Zeev Suraski dan Andi Gutmants (disingkat Zend) dan lahirlah PHP 3 pada 1998.

PHP masih terus dikembangkan, pada tahun 2002 PHP hadir dengan versi ke 4. PHP 4 dilengkapi dengan Zend Engine dan mengalami banyak peningkatan performa. Dan yang paling akhir pada tahun 2005 PHP 5 hadir dengan dukungan Zend Engine 2. Banyak fitur baru khususnya pada OOP(Object Oriented Programming) dan XML

Berbeda dengan dengan PHP 4, adopsi PHP 5 dikalangan industri sangat lambat. Bahkan hingga sekarang masih cukup banyak hosting yang belum menyediakan PHP 5. Hal ini dikarenakan beberapa faktor, diantanranya kompatibilitas dengan versi PHP 4.

Pada saat buku ini ditulis versi terakhir dari PHP adalah versi 5.2.8 untuk PHP 5 dan versi 4.4.9 untuk PHP 4. PHP 6 juga sudah dikembangkan meskipun dalam tahap alpha.



2.2 Cara Kerja PHP

Cara kerja PHP yang akan kita bahas pada buku ini adalah PHP sebagai bahasa pemrograman untuk mengembangan aplikasi berbasis web. Karena selain untuk web-programming PHP juga dapat digunakan untuk mengembangan aplikasi berbasis desktop dan CLI (Command Line Interface).



Gambar 2.1: Cara kerja PHP

Secara sederhana cara kerja PHP dapat dilihat pada gambar 2.1 diatas. Jika diurut maka proses tersebut terdiri dari:

- 1. User meminta sebuah halaman PHP
- 2. Browser mengirim HTTP Request kepada WebServer, misal Apache
- 3. WebServer mengirim permintaan file PHP tersebut ke PHP processor. PHP processor dapat

berupa modul(bagian dari web-server) atau terpisah(sebagai CGI/Fast-CGI)

- 4. Permintaan diproses oleh PHP processor kemudian hasilnya dikirim kembali ke web-server
- 5. Web server memaket kembali hasil tersebut dengan menambahkan HTTP header dan dikirim kembali ke browser.
- 6. Browser memproses HTTP packet dan menampilkannya sebagai HTML kepada user.

2.3 Variabel pada PHP

Variabel merupakan elemen yang sangat penting dalam bahasa pemrograman. Hampir setiap bahasa pemrograman mengenal apa itu yang namanya variabel. Variabel sendiri merupakan suatu bentuk penyimpanan data sementara pada memori komputer yang akan diolah lebih lanjut. Contoh variabel pada PHP antaran lain:

```
$nama = 'LUG STIKOMP Surabaya';
$angka_1 = 1;
$angka_2 = 2;
$hasil = $angka_1 + $angka_2;
```

Ketentuan-ketentuan dalam membuat variabel:

- Variabel dapat terdiri dari huruf, angka dan underscore(_) dan tentu saja tanda dollar \$.
- Variabel tidak dapat diawali dengan angka.
- Variabel bersifat case sensitive artinya membedakan huruf kecil dan huruf besar.
- Variabel \$nama tidak sama dengan \$NaMa. Untuk itu hati-hati dalam penulisan nama variabel.

Pada variabel anda dapat mengubah nilainya, contoh $angka_1 = 1$ dapat anda ubah menjadi $angka_1 = 2$ saat suatu script dijalankan. Ini berbeda dengan konstanta(akan dibahas berikutnya).

2.3.1 Tipe Variabel

Pada PHP kita tidak perlu mendeklarasikan tipe variabel secara eksplisit, istilah kerennya *dynamic typing*. Karena PHP secara otomatis dapat menentukan tipe variabel berdasarkan nilai yang ada pada variabel tersebut. Berikut ini adalah beberapa tipe data yang dikudung oleh PHP.

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

Tabel 2.1: Macam-macam Tipe Data

2.3.2 Konstanta

Hampir sama dengan variabel, konstanta juga digunakan untuk penyimpanan nilai sementara. Namun perbedaan konstanta dengan variabel adalah pada konstanta anda tidak dapat mengubah nilainya jika sudah dideklarasikan. Cara pendeklarasiannya pun berbeda dengan variabel. Pada konstanta digunakan keyword define untuk mendeklarasikan variabel. Konstanta juga tidak diawali dengan tanda \$ (dollar).

```
define('HARGA', 1500);
define('NAMA', 'LUG STIKOMP Surabaya');
```

Dapat dilihat pada kode diatas bahwa kita selalu gunakan huruf KAPITAL untuk konstanta. Hal ini tidak harus dilakukan namun semacam peraturan tidak tertulis jika konstanta maka sebaiknya gunakan huruf kapital.

2.3.3 Penerapan Teori Variabel dan Konstanta

- 1. Jika belum silahkan buat folder *bab2* didalam folder *htdocs/webpro*.
- 2. Pastikan Apache sudah berjalan.
- 3. Buka gedit (Linux)/Notepad++ (Windows)
- 4. Ketik kode berikut:

```
1. <! DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2.
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3. <html xmlns="http://www.w3.org/TR/xhtml1">
4. <head>
5.
     <title>Variabel dan Konstanta</title>
6. </head>
7. <body>
8. <?php
9.
      $menu = 'Menu: Nasi Pecel';
10.
    define('HARGA', 2500);
11.
12. $menu = 'Menu: Nasi Rawon';
13. define('HARGA', 5000);
14.
15.
    echo $menu . '<br />';
16.
    echo HARGA;
17.?>
18.</body>
19.</html>
```

- 5. Simpan dengan nama variabel.php pada folder bab2
- 6. Arahkan browser and a ke alamat http://localhost/webpro/bab2/variabel.html
- 7. Output dari file tersebut adalah

```
Menu: Nasi Rawon
2500
```

Klik kanan browser anda lalu klik View Source (untuk Firefox) untuk melihat kode HTML dari halaman tersebut. Perhatikan bahwa tag <?php dan ?> yang ada hanyalah kode HTML murni.

Penjelasan Script variabel.php

Jika anda perhatikan, sama seperti pada bab sebelumnya kita hanya menulis sebuah file html biasa.

Namun ada sedikit perbedaan, dimana pada file variabel.php terdapat sebuah tag baru yaitu <?php dan ?>. Script dalam tag inilah yang akan diproses oleh PHP processor. Semua tulisan diluar tag <? php dan ?> akan diabaikan oleh PHP.

Cara seperti yang kita lakukan pada file variabel.php adalah menempelkan PHP pada HTML. Jika kumpulan mp3 anda adalah western song mungkin anda lebih suka mendengarnya dengan sebutan embedding PHP inside HTML :).

Jika anda masih ingat teori sebelumnya tentang variabel, dapat dilihat bahwa variabel \$menu kita deklarasi ulang dan nilainya diubah menjadi 'Menu: Nasi Rawon' (Baris: 12). Namun hal tersebut tidak berlaku pada konstanta, pada baris 13 kita deklarasikan ulang konstanta HARGA. Nilai dari konstanta tersebut tidak berubah.

Hal itu ditunjukkan ketika kita mencetak nilai dari \$menu dan HARGA. \$menu berubah menjadi 'Nasi Rawon' dan konstanta tetap 2500.

Pada PHP untuk mencetak sesuatu dapat digunakan perintah echo. Contoh penggunaan dapat dilihat pada baris 15 dan 16. Pada konstanta tidak perlu diberi tanda \$.

TIPS:

Selain perintah **echo** anda juga dapat menggunakan perintah **print** untuk mencetak output.

2.4 Komentar

Komentar pada script bertujuan untuk memberitahu pembacanya, baik orang lain ataupun anda sendiri. Biasanya komentar digunakan untuk menjelaskan tujuan dituliskannya script tersebut, siapa penulisnya, kapan ditulis dan sebagainya. Komentar juga berguna bagi anda sendiri ketika suatu waktu anda lupa mengapa anda menulis file ini, fungsi-fungsinya untuk apa dan banyak lainnya.

PHP akan mengabaikan semua text didalam komentar. Jadi tidak akan berpengaruh pada jalannya suatu script.

PHP mengenal tiga jenis gaya komentar. Pertama model bahasa C (banyak baris/multi line), biasanya dapat diletakkan pada bagian atas script. Lihatlah script dibawah ini.

Yang kedua adalah model C++, seperti yang ditunjukkan dibawah ini.

echo \$nama; // cetak nama

Yang ketiga adalah model shell script.

echo HARGA; # *cetak harga* Sekarang coba modifikasi file variabel.php tersebut dengan menambahkan komentar lalu lihat hasilnya. Apakah masih sama atau tidak.

2.5 Operator

Operator merupakan simbol yang dapat digunakan untuk memanipulasi nilai dan variabel. Pada bagian sebelumnya kita sudah menggunakan beberapa operator diantaranya =, ==, <, ., +, * dan lainnya. Selanjutnya kita kan membahas operator-oprator ini dan operator lainnya lebih detil lagi .

2.5.1 Operator Aritmatik

Pasti anda sudah banyak mengenal berbagai macam operator aritmatik. Jika anda pernhah mengenyam pendidikan yang bernama Sekolah Dasar atau SD pasti anda sering menulis operator-operator ini pada buku anda :).

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

Tabel 2.2: Operator Aritmatik

Yang mungkin bapak atau Ibu guru SD kita lupa mengajarkan adalah tentang modulus. Modulus merupakan sisa pembagian dari dua bilangan. Lihat contoh berikut untuk lebih jelas.

```
1. <?php
2.
3. $bil1 = 10;
4. $bil2 = 5;
5. $bil3 = 3;
6.
7. echo $bil1 % $bil2; // output 0
8. echo $bil1 % $bil3; // output 1
9. echo $bil2 % $bil3; // output 2
10.
11.?>
```

2.5.2 Operator Kombinasi

Saya sendiri bingung menjelaskan operator ini, saya takut kata-kata saya menyesatkan pikiran anda :). Jadi lebih baik kita langsung lihat contoh saja biar lebih jelas.

Tabel 2.3: Operator Kombinasi

Operator	Penggunaan	Sama dengan
+=	\$a += \$b	\$a = \$a + \$b
_=	\$a -= \$b	\$a = \$a - \$b

*=	\$a *= \$b	\$a = \$a * \$b
/=	\$a /= \$b	\$a = \$a / \$b
%=	\$a %= \$b	\$a = \$a % \$b

Penggunaan operator kombinasi memang favorit para programmer karena memang mereka ini orang-orang yang cenderung malas. Waduh, saya juga termasuk kelihatannya :).

2.5.3 Operator Perbandingan

Operator perbandingan digunakan untuk membandingkan dua nilai. Penggunaan ekspressi ini juga akan menghasilkan nilai true atau false terganting dari perbandingan.

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

 Tabel 2.4:
 Operator perbandingan

Operator perbandingan biasanya digunakan pada struktur kontrol seperti percabangan dan perulangan.

2.5.4 Operator Logika

Operator logika digunakan untuk melakukan tes terhadap sebuah kondisi logika dari dua atau lebih perbandingan. Sebagai contoh, kita mungkin menemukan kasus nilai dari variabel a berada diantara 0 sampai 10. Untuk mengetes kondisi $a \ge 0$ dan a <= 10, kita gunakan operator AND.

Operator	Nama	Penggunaaan	Keterangan
!	NOT	!\$a	TRUE jika \$a adalah FALSE, berlaku sebaliknya
11	OR	\$a \$b	TRUE jika \$a atau \$b bernilai TRUE atau keduanya bernilai TRUE
& &	AND	\$a && \$b	TRUE jika \$a dan \$b bernilai TRUE
xor	XOR	\$a xor \$b	TRUE jika \$a atau \$b bernilai TRUE, tapi tidak keduanya

Tabel 2.5: Operator Logika

2.5.5 Operator Increment/Decrement

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

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

Tabel 2.5: Operator Increment/Decrement

Mungkin akan lebih jelas jika anda melihat contoh dibawah ini.

```
1. <?php
2.
3. echo "<h3>Post-increment</h3>";
4. a = 5;
5. echo "Seharusnya 5: " . $a++ . "<br />";
6. echo "Seharusnya 6: " . $a . "<br />";
7.
8. echo "<h3>Pre-increment</h3>";
9. a = 5;
10.echo "Seharusnya 6: " . ++$a . "<br />";
11.echo "Seharusnya 6: " . $a . "<br />";
12.
13.echo "<h3>Post-decrement</h3>";
14. $a = 5;
15.echo "Seharusnya 5: " . $a-- . "<br />";
16.echo "Seharusnya 4: " . $a . "<br />";
17.
18.echo "<h3>Pre-decrement</h3>";
19.$a = 5;
20.echo "Seharusnya 4: " . --$a . "<br />";
21.echo "Seharusnya 4: " . $a . "<br />";
22.
23.?>
```

2.5.6 Operator String

Sebenarnya pada contoh-contoh sebelumnya operator string ini sudah sering kita gunakan. Operator string yang dimaksud adalah tanda "." (titik). Tanda titik ini dapat digunakan untuk menggabung string.

```
1. <?php
2.
3. $a = 'LUG';
4. $b = 'STIKOMP';
5. $c = $a . ' ' . $b; // hasil: LUG STIKOMP
6.
7. // atau dengan kombinasi
8. $a .= ' STIKOMP'; // hasil: LUG STIKOMP
9.
10.?>
```

2.6 Escape Character

Pada saat menulis kode anda akan sangat sering menjumpai suatu kondisi dimana kita harus mencetak tanda petik baik petik satu maupun petik dua. Permasalahannya tanda tersebut sudah

digunakan sebagai penanda untuk mencetak string. Lalu, bagaimana pemecahannya? Ada beberapa solusi untuk mencetak petik didalam petik itu sendiri:

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

Berikut ini adalah contoh penyelesaian dari kedua solusi diatas.

<?php

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

2.7 Special Character

Pembahasan ini sebenarnya tidak begitu krusial jika dihubungkan dengan pembuatan tampilan web. Namun akan sangat berguna dalam proses pencarian kesalahan ketika kita mendesain suatu halaman website. Bingung? Sama saya juga bingung dengan apa yang saya tulis :p.

Pada intinya HTML mengabaikan semua karakter whitespace (spasi, tab, baris baru) dan menggantinya hanya dengan satu spasi ketika ditampilkan. Special character disni mencakup

- Baris baru (\n)
- Tab (\t)

Sebenarnya masih banyak karakter spesial lainnya seperti \r, \0, \b dan lain-lain. Untuk lebih jelasnya cobalah contoh berikut.

```
1. <?php
2.
3. echo "INI\t\tBANYAK\t\tSPASI<br />";
4. echo "INI BARIS SATU<br />";
5. echo "INI BARIS DUA<br />";
6. echo "INI BARIS TIGA<br />";
7.
8. echo "\n\n<br />\n\n";
9. echo "INI BARIS SATU<br />\nINI BARIS DUA<br />\nINI BARIS TIGA";
```

10. 11.?>

Output dari kode diatas kurang lebih seperti berikut:

INI BANYAK SPASI INI BARIS SATU INI BARIS DUA INI BARIS TIGA INI BARIS DUA INI BARIS DUA INI BARIS TIGA

Jika dilihat dari browser tidak ada yang berbeda dari group 1 dan group 2 yang ada pada kode diatas. Tapi jika anda melihat source HTML dengan cara klik kanan \rightarrow View Source (Firefox) akan terlihat perbedaannya.



TIPS:

Untuk mencetak special character seperti \n, \t, \r dan sebagainya harus menggunakan petik dua sebagai penutup string.

Ringkasan Bab II

Pada Bab II kita telah membahas sejarah singkat tentang PHP yang diciptakan oleh Rasmus Lerdorf dan kemudian dikembangkan oleh Zend. Kita juga telah membahas hal-hal dasar seputar PHP yang meliputi:

- Variabel
- Konstanta
- Operator
- Special Character

Bab III Struktur Kontrol

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

- if...
- if...else...
- if...elseif...else...
- switch...case...break...

Sedangkan struktur perulangan terdiri dari:

- for...
- while...
- do...while...
- foreach...

3.1 Struktur Kondisi

3.1.1 Statement if...

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

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

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

```
1. <?php
2.
3. $nama = 'LUG STIKOMP';
4. if ($nama == 'LUG STIKOMP') {
5. echo 'OK, anda boleh masuk.';
6. }
7.
8. ?>
```

Pada contoh diatas kita menggunakan variabel nama sebagai kondisi, dimana jika nilai variabel \$nama sama dengan LUG STIKOMP maka blok perintah akan dieksekusi. Ingat, bahwa operator perbandingan untuk sama dengan adalah == bukan =.

Contoh lain jika kita menggunakan angka untuk perbandingan adalah:

```
1. <?php
```

2.

```
3. $usia = 21;
4. $black_list = FALSE;
5. if ($usia >= 21 && $black_list == FALSE) {
6. echo 'OK, anda sudah dewasa. Silahkan masuk.';
7. }
8.
9. ?>
```

Pada contoh ke dua ini, blok if hanya akan dieksekusi jika nilai dari usia *lebih dari atau sama dengan* 21 DAN nilai dari *black_list* sama dengan false.

TIPS: Anda dapat meletakkan if di dalam if (if bersarang). Ini juga berlaku terhadap semua struktur kontrol lain.

3.1.2 Statement if...else...

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

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

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

```
1. <?php
2.
3. $peghasilan = 5000000;
4. if ($penghasilan >= 3000000) {
5. echo 'Pak Toyib: "Nak john, kamu boleh menikahi putri saya :)"';
6. } else {
7. echo 'Pak Toyib: "Mau kamu kasih makan apa putri saya nanti!!!"';
8. }
9.
10.?>
```

Alur dari script diatas sudah jelas, dimana jika nilai dari \$penghasilan lebih dari 3.000.000 maka john diperbolehkan nikah dengan putrinya Pak Toyib :). Namun jika kurang dari 3.000.000 alamat kena marah Pak Toyib :(.

3.1.3 Statemen if...elseif...else...

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

```
if (kondisi_1) {
```

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

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

3.1.4 Penerapan Teori if...elseif...else...

Hal yang paling sering dilakukan oleh PHP sebagai bahasa server-side scripting adalah memproses HTML Form. Pada contoh ini adalah inti awal anda memahami tugas PHP sebagai pemroses HTML Form. Kita akan membuat dua file yang pertama adalah file HTML murni sebagai input data bagi user(HTML Form). Dan file yang kedua adalah file PHP yang bertugas memproses data yang dikirim oleh file HTML .

- 1. Jika belum silahkan buat folder *bab3* pada folder *htdocs/webpro*
- 2. Pastikan Apache sudah berjalan
- 3. Buka gedit / Notepad++
- 4. Simpan file tersebut dengan nama form-toko.html
- 5. Ketik kode berikut:

```
1. <! DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3. <html xmlns="http://www.w3.org/TR/xhtml1">
4. <head>
     <title>Pembelian Online</title>
5.
6. </head>
7. <body>
    <h3>LUG ONLINE STORE</h3><hr />
8.
9.
    <form action="form-proses.php" method="post">
      <label>Sticker Linux (Rp7.500,-)</label><br />
10.
        <label>Jumlah: </label>
11.
        <input type="text" name="sticker" value="0" size="4" /><br /><br />
12.
13.
14.
        <label>Kaos Linux (Rp35.000,-)</label><br />
15.
        <label>Jumlah: </label>
        <input type="text" name="kaos" value="0" size="4" /><br /><br />
16.
17.
18.
        <label>Jacket Linux (Rp35.000,-)</label><br />
19.
        <label>Jumlah: </label>
        <input type="text" name="jacket" value="0" size="4" /><br /><br />
20.
21.
        <input type="submit" value="P r o s e s" />
22.
     </form>
23.
24.</body>
25.</html>
```

- 6. Save kembali file tersebut
- 7. Sekarang buat file baru
- 8. Simpan dengan nama *form-proses.php* masih difolder yang sama.

9. Ketik kode berikut:

```
1. <?php
2.
3. // buat variabel untuk menampung data dari file form-toko.html
4. $sticker = $_POST['sticker']; // data sticker
5. $kaos = $_POST['kaos']; // data kaos
6. $jacket = $ POST['jacket']; // data jacket
7.
8. // harga dari masing-masing produk
9. define('HARGA STCIKER', 7500);
10.define('HARGA KAOS', 35000);
11.define('HARGA JACKET', 55000);
12.
13.// total harga
14.$total = (HARGA STICKER * $sticker) + (HARGA KAOS * $kaos) +
15.
           (HARGA JACKET * $jacket);
16.
17.$diskon = 0;
18.$pesan_diskon = '0%';
19.
20.// cek total untuk menentukan diskon
21.if ($total >= 50000 && $total < 75000) {
22. $diskon = 0.05; // diskon 5%
23. $pesan diskon = '5%';
24.} elseif ($total >= 75000 && $total < 100000) {
    diskon = 0.1;
25.
                     // diskon 10%
    $pesan diskon = '10%';
26.
27.} elseif ($total >= 100000) {
    diskon = 0.15;
                     // diskon 15%
28.
     $pesan_diskon = '15%';
29.
30.}
31.$subtotal = $total - ($total * $diskon);
32.
33.?>
34.<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
35. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
36.<html xmlns="http://www.w3.org/TR/xhtml1">
37.<head>
38.
   <title>Pembelian Online</title>
39.
     <style type="text/css">
40.
        .tanda { background-color: #cfcfcf; }
41.
     </style>
42.</head>
43. < body >
44.
    <strong>LUG ONLINE STORE - DATA PEMBELIAN</strong><hr />
45.
     46.
        47.
           BarangJumlahTotal
        48.
49.
        50.
           Sticker
51.
           <?php echo $sticker;?>
52.
           <?php echo ($sticker * HARGA STCIKER);?>
53.
        54.
        Kaos
55.
           <?php echo $kaos;?>
56.
           <?php echo ($kaos * HARGA KAOS);?>
57.
        58.
59.
        Jacket
60.
```

61.	php echo \$jacket;?
62.	php echo (\$jacket * HARGA JACKET);?
63.	
64.	
65.	TOTAL
66.	php echo \$total;?
67.	
68.	
69.	Diskon
70.	php echo \$pesan_diskon;?
71.	
72.	
73.	SUB-TOTAL
74.	php echo \$subtotal;?
75.	
76. <td>able></td>	able>
77. <td>7></td>	7>
78. <td>></td>	>

- 10. Simpan kembali file tersebut
- 11. Buka browser anda dan arahkan ke http://localhost/webpro/bab3/form-toko.html
- 12. Hasilnya terlihat seperti gambar 3.1 dan 3.2

Eile Edit View History Beekmarks Tools	
	🗙 🗖 🗕 Pembelian Online - Mozilla Firef 🗸
	Eile Edit View History Bookmarks Tools Help
Sticker Linux (Rp7.500,-) Jumlah: 0	Barang Jumlah Total
Kaos Linux (Rp35.000,-) Jumlah: 0	Sticker 5 37500 Kaos 2 70000
Jacket Linux (Rp55.000,-) Jumlah: 0	TOTAL 70000 Diskon 5%
Proses	SUB-TOTAL 66500
Done ///	Done //.

Gambar 3.1

```
Gambar 3.2
```

Penjelasan singkat form-toko.html dan form-proses.php

Kita mulai dari file form-toko.html terlebih dulu. Pada file ini yang perlu diperhatikan adalah data yang terletak pada <form> dan </form>. Perhatikan cuplikan script form berikut:

<form action="form-proses.php" method="post">

Perhatikan nilai dari action, file yang diberi "kehormatan" untuk menjalankan tugas memproses data yang dikirim oleh form adalah file *form-proses*.php. Letak file tersebut satu folder dengan form-toko.html. Metode yang digunakan adalah POST bukan GET.

<input type="text" name="sticker" value="0" size="4" />

Arti dari kode diatas adalah data yang akan kita kirimkan ke file *form-proses*.php bernama *sticker*. Jadi pada file *form-proses.php* kita mengambilnya dengan cara:

```
$sticker = $ POST['sticker']; // data sticker
```

Kode diatas berarti kita menangkap data yang dikirim dengan metode post dan data tersebut bernama *sticker*. Isi dari **\$_POST** *harus* sama dengan isi dari *name* pada form. Nama variabel penampung tidak harus sama akan tetapi lebih mudah jika sama dengan data yang di-post.



TIPS: *PHP tidak memperdulikan tata letak kode anda. Jadi desainlah kode anda senyaman mungkin untuk dibaca.*

3.1.5 Statement switch...case...break

Sebenarnya statement switch...case...break sama dengan if...elseif... dimana kita dapat memilih lebih dari dua kondisi atau lebih. Selain itu pada switch kita dapat memilih kondisi tidak hanya pada nilai true atau false saja tetapi kita juga dapat memilih dari berbagai tipe lain seperti integer, boolean, string dan lain-lain. Syntax penulisan statement switch adalah sebagai berikut.

```
switch ($kondisi) {
  case kondisi_1:
    // kode yang dieksekusi
  break;
  case kondisi_2:
    // kode yang dieksekusi
  break;
  case kondisi_3:
    // kode yang dieksekusi
  break;
  default:
    // kode yang dieksekusi jika tidak ada yang memenuhi
  break;
}
```

Block default pada switch sama dengan else pada statement if...elseif...else.... Yaitu blok yang akan dieksekusi jika semua kondisi tidak terpenuhi. Sebagai contoh anda dapat mengganti kode pada form-proses.php baris 21 – 30 dengan kode berikut:

```
21.switch ($total) {
22. case $total >= 50000 && $total < 75000:
        $diskon = 0.05; // diskon 5%
23.
         $pesan_diskon = '5%';
24.
25. break;
    case $total >= 75000 && $total < 100000:</pre>
26.
27.
         $diskon = 0.05; // diskon 5%
         $pesan_diskon = '5%';
28.
    break;
29.
    case $total >= 100000;
30.
         $diskon = 0.05; // diskon 5%
31.
         $pesan diskon = '5%';
32.
```

33. break; 34.}

3.2 Struktur Perulangan (Looping)

Salah satu kelebihan komputer adalah melakukan pekerjaan selama berulang-ulang tanpa rasa bosan sedikit pun. Berbeda dengan manusia, tidak dapat dibayangkan jika komputer tiba-tiba bosan ketika menjalankan perintah :D. Dalam pemrograman perulangan atau looping adalah eksekusi suatu blok kode selama suatu kondisi looping masih terpenuhi. Ini berbeda dengan struktur kontrol yang hanya mengeksekusi blok kode satu kali saja.

Agar pengertian looping mudah dicerna, seperti makanan saja dicerna :p, kita akan membuat sebuah file HTML static yang kemudian akan kita buat duplikatnya tetapi menggunakan loop.

- 1. Buka gedit / Notepad++
- 2. Buat file baru
- 3. Simpan dengan nama bensin.html letakkan pada folder htdocs/webpro/bab3
- 4. Ketik kode berikut:

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2.
3. <html xmlns="http://www.w3.org/TR/xhtml1">
4. <head>
5.
   <title>Bensin Static</title>
6. </head>
7. <body>
8.
   9.
     10.
       Bensin (Liter)Harga (Rp)
11.
     12.
     522500
13.
     14.
15.
     16.
       1045000
17.
     18.
     19.
       1567500
     20.
21.
     22.
       2090000
23.
     24.
     25.
       25112500
26.
     27.
   28.</body>
29.</html>
```

- 6. Save kembali file bensin.html
- 7. Buka file tersebut pada browser, http://localhost/webpro/bab3/bensin.html
- 8. Hasilnya akan terlihat seperti gambar 3.3

X		Ben	sin	Stat	ic - N	1ozill	\sim
<u>F</u> ile	<u>E</u> dit	⊻iew	Нi	<u>s</u> tory	<u>B</u> ool	kmark	s <u>T</u> o
Be	nsir	n (Lite	r)	Harg	ga (R	p)	i
5				2250	00		н
10				4500	00		н
15				6750	00		н
20				9000	00		н
25				1125	500		•
Don	e						11.

Gambar 3.3

3.2.1 Statement for...

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

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

Keterangan:

- **ekspresi1:** dieksekusi satu kali saat loop mulai dieksekusi, biasanya untuk inisialisasi variabel counter
- **kondisi:** dicek setiap loop apakah kondisi masih true, jika ya maka blok kode tetap dieksekusi, berlaku sebaliknya jika false
- **ekspresi2:** dieksekusi setiap akhir loop, disini biasanya digunakan untuk mengubah nilai variabel counter

3.2.2 Penerapan Teori for...

Seperti yang pernah disinggung sebelumnya, bahwa kita akan kan menduplikasi file bensin.html tetapi dengan menggunakan perulangan, dalam hal ini kita gunakan for.

- 1. Buka gedit/Notepad++
- 2. Buat file baru
- 3. Simpan dengan nama bensin_for.php pada folder htdocs/webpro/bab3
- 4. Ketik kode berikut:

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3. <html xmlns="http://www.w3.org/TR/xhtml1">
4. <head>
5. <title>Bensin For</title>
6. </head>
7. <body>
8. 
9.
```

```
10.
          Bensin (Liter)Harga (Rp)
       11.
12.
       <?php
13.
          $harga = 4500; // semoga tidak naik lagi :p
14.
15.
          // lakukan loop dengan kelipatan 5 sampai dengan 25
          for ($liter = 5; $liter <= 25; $liter += 5) {</pre>
16.
17.
            echo "\n";
18.
            echo "$liter" . ($harga * $liter) . "";
19.
            echo "\n";
20.
          }
21.
       ?>
22.
    23.</body>
24.</html>
```

- 6. Simpan kembali file bensin_for.php
- 7. Arahkan browser and a ke http://localhost/webpro/bab3/bensin_for.php
- 8. Hasilnya akan sama persis seperti gambar 3.3

Penjelasan Singkat File bensin_for.php

Nilai variabel \$liter awal kita set 5, kondisi yang ingin cek adalah jika \$liter <= 25. Selama masih terpenuhi(kondisi true) maka blok kode akan tetap dieksekusi. Setiap selesai eksekusi nilai dari variabel counter yaitu \$liter kita tambah dengan 5.

Looping akan berhenti jika sudah mencapai step 6. Karena pada saat itu nilai \$liter sudah mencapai 30 dan itu tidak memenuhi syarat kondisi dimana \$liter <= 25.

3.2.3 Statement while...

Berbeda dengan for pada while kita tidak bisa meletakkan inisialisasi variabel, kondisi, dan counter pada satu baris saja. Syntax penulisan while adalah sebagai berikut.

```
while (kondisi) {
    // blok kode yang dieksekusi
}
```

Jadi selama kondisi masih true maka blok kode akan terus dieksekusi. While biasa digunakan jika kita tidak tahu batas akhir kapan suatu perulangan harus dihentikan.

3.2.4 Penerapan Teori while...

Kita masih tetap menggunakan file bensin.html sebagai acuan output yang akan diselesaikan menggunakan while.

- 1. Buka gedit / Notepad++
- 2. Buat file baru
- 3. Simpan dengan nama bensin_while.php pada folder htdocs/webpro/bab3
- 4. Ketik kode berikut:

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
3. <html xmlns="http://www.w3.org/TR/xhtml1">
```

```
4. <head>
    <title>Bensin While</title>
5.
6. </head>
7. <body>
    8.
       9.
10.
         Bensin (Liter)Harga (Rp)
       11.
       <?php
12.
13.
         $harga = 4500; // semoga tidak naik lagi :p
14.
         fiter = 5;
                       // inisialisasi $liter diluar loop
15.
16.
         // lakukan loop dengan kelipatan 5 sampai dengan 25
         while ($liter <= 25) {
17.
           echo "\n";
18.
           echo "$liter". ($harga * $liter) . "\n";
19.
           echo "\n";
20.
21.
           $liter += 5;
                       // tambah counter
22.
         }
23.
       ?>
24.
    25.</body>
26.</html>
```

- 5. Simpan kembali file tersebut
- 6. Coba lihat hasilnya, http://localhost/webpro/bab3/bensin_while.php
- 7. Hasilnya akan terlihat seperti gambar 3.3 sebelumnya

3.2.5 Statement do...while

Berbede dengan dua loop sebelumnya, dimana pada for dan while pengecekan kondisi dilakukan pada awal sebelum blok kode dieksekusi. Pada do...while pengecekan dilakukan diakhir eksekusi kode. Jadi setidak-tidaknya blok kode do...while *pasti* dieksekusi minimal satu kali. Syntax dasar dari do...while adalah sebagai berikut.

```
do {
    // kode yang dieksekusi
} while (kondisi)
```

Blok kode do akan dieksekusi selama kondisi while masih bernilai true.

3.2.6 Penerapan Teori do...while

- 1. Buka gedit / Notepad++
- 2. Buat file baru
- 3. Simpan dengan nama bensin_do_while.php pada folder htdocs/webpro/bab3
- 4. Ketik kode berikut:

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3. <html xmlns="http://www.w3.org/TR/xhtml1">
```

```
3. <ntml xmlns="nttp://www.w3.org/TR/Xnt
4. <head>
```

```
5. <title>Bensin Do While</title>
```

- 6. </head>
- 7. <body>

```
8.
```

```
9.
       10.
          Bensin (Liter)Harga (Rp)
11.
       12.
       <?php
13.
         $harga = 4500; // semoga tidak naik lagi :p
14.
                        // inisialisasi $liter diluar loop
         fiter = 5;
15.
16.
         // lakukan loop dengan kelipatan 5 sampai dengan 25
17.
         do {
18.
            echo "\n";
19.
            echo "$liter". ($harga * $liter) . "\n";
20.
            echo "\n";
21.
            $liter += 5; // tambah counter
22.
         } while ($liter <= 25)</pre>
23.
       ?>
24.
    25.</body>
26.</html>
```

- 5. Simpan kembali file tersebut
- 6. Arahkan browser ke http://localhost/webpro/bab3/bensin_while.php untuk melihat hasilnya

Jika anda masih penasaran dengan kata-kata "minimal dieksekusi satu kali" coba ganti nilai \$liter dari 5 menjadi 30 saat inisialisasi. Kode pasti akan dieksekusi satu kali kemudian selesai. Hal ini tidak berlaku untuk for dan while.

Jika ditanya kapan kita menggunakan do...while, susah juga jawabnya. Yang jelas secara naluri nanti anda akan menggunakannya tanpa disuruh-suruh orang lain :p.

TIPS: Looping for biasa digunakan jika kita telah mengetahui batas awal dan akhir dari suatu perulangan.

Untuk sementara kita tidak menbahas dulu *foreach* karena loop ini didesain khusus untuk penggunaan didalam array. Jadi pembahasan foreach akan kita lakukan saat membahas tentang array.

Ringkasan Bab III

Pada bab III ini kita telah membahas aspek terpenting dari bahasa pemrograman struktur kontrol. Struktur kontrol terdari dari struktur percabangan (branching) dan perlulangan (looping). Pada PHP percabangan terdiri dari:

- if...
- if...else...
- if...elseif...else...
- switch...case...break...

Sedangkan struktur perulangan terdiri dari:

- for...
- while...
- do...while...
- foreach...

Bab IV File dan Array

4.1 Berinteraksi dengan File

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

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

4.1.1 Menyimpan File

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

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

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



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

Mode yang dapat digunakan pada fopen adalah:

TIPS:

Agar kompatibel antar sistem operasi yang memiliki "ending line" yang berbeda Linux (n), MacOS(r), dan Windows(rn) tambahkan opsi 'b', misal 'wb'. (b = binary mode)

Kita akan menggunakan contoh form-toko.html yang terdapat pada bab 3. Format penyimpanan yang kita gunakan adalah:

jml_kaos Sticker -#- harga_kaos -#- jml_kaos Kaos-#- harga_kaos -#- jml_jacket Jacket -#- harga_jacket -#- total -#- diskon -#- sub-total

OK, langsung aja kita coba untuk menulis file.

4.1.3 Penerapan Teori Penyimpanan File

Sebelum memulai pastikan anda sudah membuat folder baru di direktori *htdocs* anda dengan nama *bab4*. Kita akan menyimpan file tersebut didalam folder bab4 dengan nama *faktur.txt*. Ikuti langkah-langkah berikut:

- 1. Copy file form-toko.html dan form-proses dari folder bab3 ke bab4
- 2. Modifikasi file form-proses.php(pada bab4) agar menampilkan link ke halaman faktur.

```
1. <?php
```

```
2. // buat variabel untuk menampung data dari file form-toko.html
3. $sticker = $ POST['sticker']; // data sticker
4. $kaos = $ POST['kaos']; // data kaos
5. $jacket = $_POST['jacket']; // data jacket
6. // harga dari masing-masing produk
7. define('HARGA_STICKER', 7500);
8. define('HARGA_KAOS', 35000);
9. define('HARGA_JACKET', 55000);
10. // total harga
11. $total = (HARGA STICKER * $sticker) + (HARGA KAOS * $kaos) +
12.
             (HARGA JACKET * $jacket);
13. \frac{13}{0};
14. pesan diskon = '0%';
15. // cek total untuk menentukan diskon
16. if ($total >= 50000 && $total < 75000) {
      diskon = 0.05;
                       // diskon 5%
17.
     $pesan diskon = '5%';
18.
19. } elseif ($total >= 75000 && $total < 100000) {
20.
      diskon = 0.1;
                        // diskon 10%
      $pesan_diskon = '10%';
21.
22. } elseif ($total >= 100000) {
23.
      diskon = 0.15;
                       // diskon 15%
24.
      $pesan diskon = '15%';
25. }
26. $subtotal = $total - ($total * $diskon);
27. // tulis ke file, mode yang digunakan adalah 'a' => append
28. // file faktur.txt akan dibuat otomatis jika belum ada
29. $fp = fopen('faktur.txt', 'a');
30. $isi_file = "$sticker Sticker -#- " . (HARGA_STICKER * $sticker) . " -#- " .
                "$kaos Kaos -#- " . (HARGA_KAOS * $kaos) . " -#- "
31.
32.
                "$jacket Jacket -#- " . (HARGA JACKET * $jacket) . " -#- " .
                "$total -#- $pesan diskon -#- $subtotal\n";
33.
34. fwrite($fp, $isi file);
35. fclose($fp);
```

...

© Copyright 2009 Rio Astamal – Hal.51

36.	?>						
37.	html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN</td						
38.	"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">						
39.	<html xmlns="http://www.w3.org/TR/xhtml1"></html>						
40.	<head></head>						
41.	<title>Pembelian Online</title>						
42.	<style type="text/css"></td></tr><tr><td>43.</td><td><pre>.tanda { background-color: #cfcfcf; }</pre></td></tr><tr><td>44.</td><td></style>						
45.							
46.	<body></body>						
47.	LUG ONLINE STORE - DATA PEMBELIAN <hr/>						
48.							
49.							
50.	BarangJumlahTotal						
51.							
52.							
53.	Sticker						
54.	>?php echo \$sticker;?>						
55.	php echo (\$sticker * HARGA_STCIKER);?						
56.							
5/.							
50.	td > dos						
59. 60							
61							
62	<pre></pre>						
63.	lacket						
64.							
65.							
66.							
67.							
68.	TOTAL						
69.	php echo \$total;?						
70.							
71.							
72.	Diskon						
73.	php echo \$pesan_diskon;?						
74.							
75.							
76.	SUB-TOTAL						
77.	php echo \$subtotal;?						
78.							
79.							
80.	Lihat Faktur						
81.							
82.							

- 3. Simpan kembali file form-proses.php
- 4. Buat sebuah dokumen baru dengan nama faktur.php
- 5. Isi dari file *faktur.php* adalah:

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3. <html xmlns="http://www.w3.org/TR/xhtml1">
4. <head>
5. <title>Pembelian Online</title>
6. </head>
7. <body>
8. <h2>Data Faktur Pembelian</h2>
9.
```

```
10. <?php
11. // baca seluruh isi file
12. readfile('faktur.txt');
13. ?>
14. 
15. <a href="form-toko.html">Kembali ke Form</a>
16. </body>
17. </html>
```

- 6. Simpan kembali file *faktur.php*
- 7. Buka alamat http://localhost/webpro/bab4/form-toko.html untuk melihat proses penyimpanan file.

×□	Pemb	elian On	line - Moz	zilla Fir	efox	1
<u>F</u> ile <u>E</u> dit	: <u>∨</u> iew H	i <u>s</u> tory <u>B</u> o	okmarks	<u>T</u> ools	<u>H</u> elp	- 11 11 11 11
LUG ONLINE STORE - DATA PEMBELIAN						
Barang	y Jumlah	Total				
Sticker	6	0				
Kaos	4	140000				
Jacket	2	110000				
TOTAL		295000				
Diskon		15%				
SUB-TOTAL		250750				
Lihat Faktur						
Done						

Gambar 4.1: Tambahan link Lihat Faktur

🗙 🗖 🚝 form.zip byth Pembelian Online - Mozilla Firefox	
<u>F</u> ile <u>E</u> dit <u>V</u> iew Hi <u>s</u> tory <u>B</u> ookmarks <u>T</u> ools <u>H</u> elp	14. 14. 14.
Data Faktur Pembelian	
5 Sticker -#- 37500 -#- 5 Kaos -#- 175000 -#- 3 Jacket -#- 165000 -#- 377500 -#- 15% -#- 320875 10 Sticker -#- 75000 -#- 5 Kaos -#- 175000 -#- 3 Jacket -#- 165000 -#- 415000 -#- 15% -#- 352750 6 Sticker -#- 45000 -#- 4 Kaos -#- 140000 -#- 2 Jacket -#- 110000 -#- 295000 -#- 15% -#- 250750	,
Kembali ke Form	
Done	2

Gambar 4.2: Data beberapa faktur transaksi yang tercatat

4.1.4 Penjelasan Singkat Penerapan Teori Penyimpanan File

Pada file form-proses.php modifikasi yang dilakukan dapat dilihat mulai dari baris 27 - 35 dan juga baris ke-80. Pada baris 27 - 35 yang kita lakukan adalah melakukan penulisan sebuah file dengan nama *faktur.txt*. Karena kita tidak menyertakan path untuk direktori file maka diasumsikan file *faktur.txt* akan ditulis pada direktori yang sama.

Mode file yang kita gunakan adalah 'a', agar data faktur-faktur sebelumnya yang sudah tertulis tidak tertimpa/overwrite melainkan kita tambahkan diakhir(append). Separator yang digunakan untuk setiap item adalah "-#-". Format ini akan tetap kita gunakan, karena pada pembahasan array kita akan mengextract setiap item berdasarkan separator tersebut.

Pada file *faktur.php* pembacaan file dilakukan dengan memanggil fungsi readfile(). Fungsi readfile akan membaca sebuah file dan langsung menampilkannya ke output. Karena setiap baris dipasahkan dengan karakter "\n" maka kita perlu tag preformatted text agar file ditampilkan apa adanya.

4.1.5 Fungsi-fungsi Lain

Beberapa fungsi lain yang dapat digunakan untuk manajemen file diantarnya baca dan hapus antara lain fungsi file_get_contents(), fread(), dan unlink().

a. file_get_contents()

Fungsi file_get_contents() digunakan untuk membaca sebuah file dan menyimpannya sebagai string. Sebagai contoh kita dapat mengganti fungsi dari readfile() pada file *faktur.php* dengan perintah berikut:

\$data = file_get_contents('faktur.txt'); echo \$data;

b. fread()

Pada fungsi fread() terdapat dua parameter yang harus disertakan yaitu pointer/resource yang dibuka dengan fopen dan panjang bytes dari data yang akan dibaca. Contoh pengganti readfile() pada file *faktur.php* adalah sebagai berikut:

```
$fp = fopen('faktur.txt', 'r');
$ukuran = filesize('faktur.txt');
$data = fread($fp, $ukuran);
echo $data;
fclose($fp);
atau
$fp = fopen('faktur.txt', 'r');
$data = '';
while (!feof($fp)) {
    $data .= fread($fp, 2048);
}
fclose($fp);
echo $data;
```

Fungsi filesize() digunakan untuk mengetahui ukuran sebuah file dalam bytes. Sedangkan fungsi feof() digunakan untuk megecek apakah akhir sebuah file sudah tercapai EOF(End of File). Pada contoh diatas kita membaca sebanyak 2048 bytes atau 2kb setiap loop.

c. fgets()

Penggunaan fungsi fgets() hampir sama dengan fread() pada metode yang kedua.

```
$fp = fopen('faktur.txt', 'r');
$data = '';
while (!feof($fp)) {
    $data .= fgets($fp, 2048);
}
fclose($fp);
echo $data;
```

d. unlink()

Berbeda dengan beberapa fungsi sebelumnya, fungsi unlink() digunakan untuk menghapus sebuah file. Contoh penggunaan unlink adalah sebagai berikut:

```
if (unlink('faktur.txt'))
   echo "File faktur.txt berhasil dihapus";
else
   echo "File faktur.txt gagal dihapus";
```

Fungsi unlink() akan mengembalikan nilai boolean mengenai status penghapusan dari sebuah file. TRUE jika file berhasil dihapus dan FALSE jika gagal dihapus.

TIPS: Masal

Masalah yang sering muncul jika menyangkut pembuatan dan penghapusan file adalah masalah permission(hak akses). Hal ini umum terjadi pada sistem operasi berbasis UNIX. Jadi perhatikan permission dari direktori/file yang menjadi objek.

4.2 Array pada PHP

Array merupakan salah satu aspek terpenting dalam bahasa pemrograman. Sebenarnya apa sih array itu? Secara sederhana *array* merupakan suatu variabel yang dapat menyimpan beberapa nilai sekaligus. Berbeda dengan variabel biasa yang hanya dapat menampung sebuah nilai saja. Array juga dapat berisi array lainnya (array didalam array), array seperti ini sering disebut array *multi-dimensi*.

Pada PHP dikenal dua tipe array berdasarkan cara pengaksesannya yaitu *array index numerik* dan *array index string (associative array)*. Bingung? Yes akhirnya saya bisa membuat anda bingung :p. OK, biar anda tidak stress langsung saja kita bahas yang pertama dulu yaitu *arrai index numerik*.

4.2.1 Array dengan Index Numerik

Kita akan tetap menggunakan contoh produk pada baba sebelumnya yaitu sticker, kaos, dan jacket dalam pembahsan contoh. Sebuah array dengan index numerik diberi index mulai dari 0,1,2,3,... dan seterusnya. Untuk membuat sebuah array dengan index numeric terdapat beberapa cara diantaranya: \rightarrow Cara 1

\$produk = array('sticker', 'kaos', 'jacket');

Perintah diatas berarti kita membuat sebuah array dengan nama \$produk dengan tiga buah elemen didalamnya. Index akan diberikan otomatis dimulai dari 0 yaitu sticker, 1 adalah kaos dan 2 adalah jacket.

\rightarrow Cara 2

```
$produk[0] = 'sticker';
$produk[1] = 'kaos';
$produk[2] = 'jacket';
```

Cara diatas adalah sebuah cara manual yaitu dengan memberi index langsung pada array.

\rightarrow Cara 3

```
$produk[] = 'sticker';
$produk[] = 'kaos';
$produk[] = 'jacket';
```

Cara ketiga hampir sama dengan yang pertama, hanya berbeda cara penulisan saja. Cara ini sangat berguna saat kita akan memberi sebuah nilai pada array saat looping dimana index akan bertambah secara otomatis.

4.2.3 Mengakses Isi Array

Untuk mengakses sebuah array hal yang diperlukan adalah nama array tersebut dan index-nya. Dimana penulisan index diapit oleh tanda kurung siku []. Perhatikan contoh berikut:

echo \$produk[0] . ' ' . \$produk[1] . ' ' . \$produk[2];

Output dari perintah diatas adalah:

sticker kaos jacket ——— Output 1

Untuk array dengan jumlah yang "bisa" dihitung dengan jari pengaksesan secara manual mungkin tidak masalah. Namun jika jumlah array mencapai puluhan bahkan ratusan? Solusinya ya pakai looping.

4.2.4 Looping Array Index Numerik

Pada bab-bab awal kita sudah membahas penggunaan looping. Looping sangat berguna untuk mengakses array dalam jumlah besar. Pada bagian ini kita juga akan membahas foreach() satu-satunya perintah looping yang belum dibahas. Output dari perintah-perintah looping berikut akan mengacu pada *output 1*. Hal ini ditujukan agar memudahkan anda memahami cara kerja array dan looping itu sendiri.

a. Looping Array dengan for()

```
// set variabel index ke 0
```

```
for ($i=0; $i<3; $i++) {
    echo "$produk[$i] ";
}</pre>
```

Hasil yang ditampilkan akan sama dengan output 1.

b. Looping Array dengan while()

```
// set variabel index ke 0
$i = 0;
while ($i < 3) {
    echo "$produk[$i] ";
    $i++; // increment $i
}</pre>
```

c. Looping Array dengan foreach()

Nah, inilah saat yang tepat kita berkenalan dengan si foreach(). Looping ini memang khusus dirancang untuk digunakan dengan array. Jadi jika memungkinkan sebaiknya gunakan loop tipe ini dibanding yang lain jika mengakses array.

Syntax dasar dari foreach adalah:

```
foreach (ekspresi1 as [eskpresi2 => ] ekspresi3) {
    // kode yang akan dieksekusi
}
```

dimana:

- ekspresi1: variabel array yang akan dimanipulasi
- ekpsresi2(opsional): variabel baru tempat menyimpan index array
- ekspresi3: variabel baru penampung nilai dari elemen array

Untuk menghasilkan output seperti output 1, maka penulisan foreach yang digunakan adalah:

```
foreach ($produk as $prd) {
   echo "$prd ";
}
```

Pada contoh diatas kita hanya menuliskan ekspresi1 dan ekspresi3. Penggunaan ekspresi2 dapat anda lihat pada contoh berikut:

```
foreach ($produk as $index => $prd) {
    echo "$index : $prd<br />";
}
Output yang ditampilkan adalah:
0 : stip
```

Output yang ditampilkan adalah:

0 : sticker 1 : kaos 2 : jacket

4.2.5 Penerapan Teori Array Index Numerik

Untuk contoh kali ini kita akan coba menampilkan isi dari faktur.txt tapi dengan tampilan yang lebih

baik. Tidak hanya sekedar isinya secara langsung. Ikuti langkah-langkag berikut:

- 1. Buka gedit(Linux) / Notepad++ (Windows)
- 2. Buat file baru didalam direktori htdocs/bab4 dengan nama faktur2.php
- 3. Ketik kode berikut:

```
1. <! DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2. "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3. <html xmlns="http://www.w3.org/TR/xhtml1">
4. <head>
5.
    <title>Pembelian Online</title>
6. </head>
7. <body>
8.
    <h2>Data Faktur Pembelian</h2>
9.
    10.
11.
        No.
12.
        Jml Sticker
13.
        Total Sticker
14.
        Jml Kaos
15.
        Total Kaos
16.
        Jml Jacket
17.
        Total Jacket
18.
        Total
19.
        Diskon
20.
        Sub-Total
     </t.r>
21.
     <?php
22.
23.
        // pecah setiap baris menjadi array dengan fungsi file()
24.
        $baris = file('faktur.txt');
25.
        // baca array
        foreach ($baris as $nomor => $isi) {
26.
27.
          echo "\n";
           echo "". ++$nomor . "\n"; // increment karena dimulai dari 0
28.
           // extract data pada setiap baris dengan separator '-#-'
29.
30.
           $data = explode('-#-', $isi); // $data sekarang berupa array
31.
           foreach ($data as $data item) {
32.
             echo "$data item\n";
33.
           }
34.
           echo "\n";
35.
        }
36.
     ?>
37.
     38. </body>
39. </html>
```

- 4. Simpan kembali file faktur2.php
- 5. Arahkan browser and a pada alamat http://localhost/webpro/bab4/faktur2.php
- 6. Hasilnya akan terlihat seperti gambar 4.3.

TIPS: Anda dapat bereksperimen dengan perintah explode(), misalnya tidak menggunakan fungsi file untuk memecah baris melainkan dengan explode() dengan separator \n.

🗙 🗖 💶 🚽 Pembelian Online - Mozilla Firefox 🥥									
<u>F</u> ile <u>E</u> dit <u>V</u> iew Hi <u>s</u> tory <u>B</u> ookmarks <u>T</u> ools <u>H</u> elp ⇔									
Data Faktur Pembelian									
No.	Jml Sticker	Total Sticker	Jml Kaos	Total Kaos	Jml Jacket	Total Jacket	Total	Diskon	Sub-Total
1	5 Sticker	37500	5 Kaos	175000	3 Jacket	165000	377500	15%	320875
2	10 Sticker	75000	5 Kaos	175000	3 Jacket	165000	415000	15%	352750
3	6 Sticker	45000	4 Kaos	140000	2 Jacket	110000	295000	15%	250750
Done									



4.2.6 Array dengan Index String (Associative Array)

Pada kasus tertentu penggunaan associative array lebih baik dibandingkan dengan array index numerik. Pada associative array, pengaksesan bukan dengan index angka melainkan string sesuai dengan keinginan kita. Sebagai contoh kita akan membuat sebuah array \$produk seperti sebelumnya namun dengan associative array.

\rightarrow Cara 1

```
$produk = array( 'stc' => 'sticker', 'ks' => 'kaos', 'jkt' => 'jacket');
```

Pada contoh diatas kita memberi index pada sticker dengan *stc*, kaos dengan *ks* dan jacket dengan *jkt*.

\rightarrow Cara 2

```
$produk = array('stc' => 'sticker');
$produk = array('ks' => 'kaos');
$produk = array('jkt' => 'jacket');
```

\rightarrow Cara 3

```
$produk['stc'] = 'sticker';
$produk['ks'] = 'kaos';
$produk['jkt'] = 'jacket';
```

4.2.7 Mengakses Associative Array

Sebenarnya pengaksesan associative array sama dengan array index numerik. Hal yang diperlukan tentu nama dari array dan nama index yang akan diakses. Perhatikan contoh berikut, output tampilan dibawah akan sama dengan *output 1* pada bahasan sebelumnya.

```
echo $produk['stc'] . ' ' . $produk['ks'] . ' ' . $produk['jkt'];
```

atau

echo "{\$produk['stc']} {\$produk['ks'} {\$produk['jkt']}";

Jika akan mencetak associative array dalam suatu string maka gunakan tanda { } (kurung kurawal) sebagai pembungkus array.

4.2.8 Looping pada Associative Array

Karena pada associative array tidak menggunakan angka sebagai index, maka kita tidak dapat menggunakan looping for pada array tipe ini.

a. Looping dengan foreach()

Associative array memang sangat cocok digunakan dengan loop foreach. Kesederhanaan syntax foreach membuat lebih mudah dibaca.

```
foreach ($produk as $index => $prd) {
    echo "$index :: $prd<br />";
}
```

Output dari script diatas adalah sebagai berikut:

stc :: sticker	
ks :: kaos	
jkt :: jacket	

to be continue...