

# Modul WEBSITE DASAR



Cara Membuat dari Zero To Hero

## Contents

Pertemuan 1 – Pengenalan HTML.....	6
1.1 HTML.....	6
1.1.1 Dokumen HTML.....	6
1.1.2 HTML.....	6
1.1.3 Cara Menuliskan Tag.....	7
1.1.4 Atribut dalam Tag.....	7
1.1.5 Value.....	7
1.1.6 Komentar.....	7
1.2 Tag-Tag Dasar HTML.....	8
1.2.1 Tag  .....	8
1.2.2 Tag Heading.....	9
1.2.3 Tag <P>.....	10
1.2.4 Tag <HR>.....	11
1.2.5 Tag <CENTER>.....	13
1.2.6 Tag <FONT>.....	14
1.2.7 Format Text.....	15
1.2.8 Pewarnaan Background <BODY>.....	15
1.3 Latihan.....	16
1.4 Link <A>.....	16
1.4.1 Link ke Situs Web Lain.....	18
1.4.2 Menentukan Warna Hyperlink.....	19
1.4.3 Membuka Jendela Baru.....	20
1.4.4 Membuat Link dalam satu Halaman.....	20
1.5 Gambar <IMG>.....	21
1.5.1 Atribut Gambar.....	21
1.6 List <OL> dan <UL>.....	22
1.6.1 List dengan Nomor / OrderedList / OL.....	23
1.6.2 List Tanpa Nomor / UnOrderedList / UL.....	24
1.7 Karakter Khusus.....	25
1.8 Latihan.....	26
Pertemuan 2 - HTML Lanjutan.....	28
2.1 Tabel <Table>.....	28
2.1.1 Memberi Garis pada Tabel.....	29
2.1.2 Meletakkan Judul Tabel.....	30

2.1.3	Mengatur Posisi Tabel.....	31
2.1.4	CellSpacing dan CellPadding.....	32
2.1.5	RowSpan dan ColSpan.....	33
2.1.6	Fungsi Atribut Width.....	34
2.1.7	Fungsi Atribut ALIGN dan VALIGN.....	35
2.1.8	Pewarnaan Pada Tabel.....	35
2.1.9	Atribut NOWRAP.....	36
2.2	HTML Layout <Div>.....	37
2.2.1	<Span>.....	38
2.3	Latihan.....	38
Pertemuan 3 – HTML Formulir.....		40
3.1	Dasar-Dasar Formulir.....	40
3.2	Pembuatan Formulir.....	40
3.2.1	Tipe Input.....	41
3.2.2	TextBox.....	41
3.2.3	Radio Button.....	42
3.2.4	Check Box.....	42
3.2.5	Password Box.....	43
3.2.6	Submit Button dan Reset Button.....	43
3.2.7	Seleksi / Check List.....	44
3.2.8	Text Area.....	44
3.2.9	Hidden Field.....	45
3.2.10	Image.....	45
3.3	Iframe.....	46
3.4	Latihan.....	47
Pertemuan 4 - Dasar CSS dan HTML 5.....		49
4.1	CSS.....	49
4.1.1	Keuntungan menggunakan CSS.....	49
4.1.2	Cara penulisan CSS.....	49
4.1.3	Aturan penulisan CSS.....	50
4.1.4	Macam-macam Selector.....	50
4.1.5	CSS Background.....	52
4.1.6	CSS Font.....	53
4.1.7	CSS Teks.....	54
4.1.8	CSS List.....	55

4.1.9	Margin dan Padding .....	55
4.2	Contoh Penggunaan CSS .....	56
4.2.1	Padding dan Margin .....	56
4.2.2	Pembuatan Menu .....	57
4.2.3	Membuat Layout .....	58
4.3	HTML 5 .....	59
4.3.1	Tag Video .....	62
4.3.2	Tag Audio .....	62
Pertemuan 5	- Dasar PHP .....	64
5.1	Pengertian PHP .....	64
5.1.1	Syntax PHP .....	65
5.1.2	Variable .....	65
5.2	Operator pada PHP .....	66
5.2.1	Operator Aritmatika .....	66
5.2.2	Unary Operator .....	66
5.2.3	Operator Perbandingan .....	66
5.2.4	String Concatenation Operator .....	67
5.2.5	Variable Assignment Shortcut .....	67
5.2.6	Incrementing/Decrementing Operators .....	68
5.3	Condition Statement .....	68
5.3.1	If statement .....	68
5.3.2	If...else Statement .....	68
5.3.3	If...else if...else Statement .....	69
5.3.4	Switch Statement .....	70
5.4	Perulangan (Looping) .....	71
5.4.1	While Loop .....	71
5.4.2	Do...while Statement .....	71
5.4.3	For Loop .....	72
5.4.4	Foreach Loop .....	72
5.5	Array .....	73
5.5.1	Inisialisasi array .....	74
5.6	Function .....	75
5.6.1	Parameter Function .....	76
5.6.2	Return Value .....	77
Pertemuan 6	- Form dan Input User .....	79

6.1	Form.....	79
6.2	Penggunaan GET .....	80
6.3	Penggunaan POST .....	81
6.4	Isset dan Empty.....	82
6.5	Halaman Login .....	84
6.6	Latihan .....	85
Pertemuan 7 – Session dan Include .....		87
7.1	Session .....	87
7.2	Penggunaan Session Pada Login .....	90
7.3	Cookies .....	92
7.4	SESSION VS COOKIE.....	93
7.5	Include(), Include_once(), require(), require_once() .....	93
Pertemuan 8 – PHP dan MySQL .....		97
8.1	MySQL .....	97
8.2	Struktur dan Syntax MySQL .....	97
8.3	Membuat Database, membuat table.....	97
8.4	Koneksi PHP dan MySQL.....	98
8.5	Database Query .....	98
8.6	Menampilkan data menggunakan Table .....	99
8.7	Input ke database .....	101
8.8	Menampilkan Detail Data .....	102
Pertemuan 9 – PHP dan MySQL 2 .....		104
9.1	Delete Data .....	104
9.2	Update Data .....	105
9.3	Login dengan Database.....	107
Pertemuan 10.....		111
10.1	Pengertian JavaScript.....	111
Pertemuan 11 .....		120
11.1	PROJECT.....	120

# Pertemuan 1

## Pengenalan HTML

### 1.1 HTML

#### 1.1.1 Dokumen HTML

- Dapat dibuat dengan menggunakan text editor(notepad) / Dreamweaver
- Dokumen harus disimpan dengan ekstensi **.html**
- Dokumen dibuka dengan menggunakan browser.
- Halaman harus direfresh apabila kode diupdate.

#### 1.1.2 HTML

- HTML singkatan dari *Hyper Text Markup Language*
- HTML merupakan *markup language*
- Markup language merupakan dokumen yang ditulis dengan kode apit (**tags**)
- Tags menentukan format, atau jenis isi pada dokumen
- Dokumen HTML disebut juga **halaman web**

#### Contoh Dokumen HTML Sederhana (**HelloWorld.html**)

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Latihan Pertama</TITLE>
</HEAD>
<BODY>
    <h1>My First Heading</h1>
    <p>My First paragraph.</p>
</BODY>
</HTML>
```

- Tag HTML merupakan kata kunci yang diapit **kurung siku**, seperti <html>
- Bagian yang terdapat dalam <HTML> dan </HTML> umumnya terbagi atas :
  1. kepala, ditandai dengan pasangan tag <HEAD> dan </HEAD>
  2. badan, ditandai dengan pasangan tag <BODY> dan </BODY>
- Pada bagian kepala, bisa ditentukan judul dokumen HTML. Judul ini ditulis dalam pasangan tag <TITLE> dan </TITLE>. Pada contoh di atas, judul dokumen HTML adalah Latihan Pertama. Judul ini diletakkan di bagian atas jendela browser.
- Tag pada HTML biasanya ditulis berpasangan <b> </b>
- Tag pertama merupakan **tag pembuka**, dan **tag berikutnya** merupakan tag penutup( biasanya diberi tanda *slash*)

### 1.1.3 Cara Menuliskan Tag

Nama tag ditulis di dalam tanda < dan >. Antara nama tag dan tanda < ditulis tanpa spasi sama sekali. Nama tag boleh ditulis dengan huruf kecil, huruf kapital, ataupun kombinasi antara huruf kecil dan huruf kapital. Dengan kata lain, penulisan :

```
<HTML>
```

```
<html>
```

```
<Html>
```

dianggap sama. Namun umumnya nama tag ditulis dengan huruf kecil saja atau huruf kapital saja.

### 1.1.4 Atribut dalam Tag

Beberapa tag mengandung atribut di dalamnya. Contoh :

```
<P ALIGN="CENTER">
```

```
<FONT SIZE="3" FACE="arial" COLOR="red">Merdeka atau Mati</font>
```

Pada contoh ini, P adalah nama tag sedangkan ALIGN adalah nama atribut. Pada umumnya, atribut melibatkan nilai. Pada contoh di atas, "CENTER" adalah nilai untuk atribut ALIGN.

### 1.1.5 Value

Value adalah nilai yang diberikan pada sebuah atribut. Value ada beberapa jenis, bisa value text seperti left, right, center, red, blue, bisa juga angka 1, 2, 3, bisa juga bentuk pixel : 100, 200, atau bentuk persen 100%, 30%, ada juga yang bentuk hexa seperti warna yang ditulis color="#000" dll.

Untuk value juga harus relevan, tidak bisa sembarangan. Contoh : align (ini untuk mengatur posisi) tidak boleh ditulis align="1" atau sebaliknya size="center".

### 1.1.6 Komentar

Komentar adalah bagian dari kode HTML yang diabaikan oleh browser. Kegunaan komentar dalam kode HTML adalah sebagai keterangan. Sebuah komentar diawali dengan **<!-- dan diakhiri dengan -->**

Contoh :

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Komentar</TITLE>
</HEAD>
<BODY>
<!-- Ini juga komentar -->
Selamat Belajar HTML
</BODY>
</HTML>
```

## 1.2 Tag-Tag Dasar HTML

### 1.2.1 Tag <BR>

Fungsi tag <BR> (Line Break) adalah untuk membuat baris baru atau berpindah baris, tag <BR> tidak perlu ditutup. **Tagbar.html**

Contoh :

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Tag BR</TITLE>
</HEAD>
<BODY>
    Selamat Datang! <BR>
    Selamat Belajar HTML<BR>
    Semoga Sukses !
</BODY>
</HTML>
```

### 1.2.2 Tag Heading

HTML menyediakan 6 buah tag yang digunakan untuk mengatur ukuran teks yang dijadikan sebagai judul dalam badan dokumen. Tag-tag judul ini berupa :

- `<H1> ... </H1>`
- `<H2> ... </H2>`
- `<H3> ... </H3>`
- `<H4> ... </H4>`
- `<H5> ... </H5>`
- `<H6> ... </H6>`

Contoh : **heading.html**

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Tag H1-H6</TITLE>
</HEAD>
<BODY>
  <H1>Belajar HTML </H1>
  <H2>Belajar HTML </H2>
  <H3> Belajar HTML </H3>
  <H4> Belajar HTML </H4>
  <H5> Belajar HTML </H5>
  <H6> Belajar HTML </H6>
</BODY>
</HTML>
```

Tag judul atau heading ini juga memiliki atribut ALIGN. Nilai yang dapat diberikan pada ALIGN adalah :

- **LEFT (default)**, mengatur teks rata kiri terhadap halaman
- **RIGHT**, mengatur teks rata kanan terhadap halaman
- **CENTER**, menempatkan judul di tengah-tengah layar pada baris yang bersangkutan
- **JUSTIFY**, mengatur teks rata kiri dan rata kanan. Efeknya terlihat untuk teks yang sangat panjang.

Contoh : **headingalign.html**

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Atribut ALIGN</TITLE>
</HEAD>
<BODY>
  <H1>Pesona Tanaman Hias</H1>
  <H1 ALIGN = "LEFT">Pesona Tanaman Hias</H1>
  <H1 ALIGN = "RIGHT">Pesona Tanaman Hias</H1>
  <H1 ALIGN = "CENTER">Pesona Tanaman Hias</H1>
</BODY>
</HTML>
```

### 1.2.3 Tag <P>

Tag <P> digunakan untuk membuat paragraf. **Paragraf.html**

Contoh :

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Contoh Paragraf</TITLE>
</HEAD>
<BODY>
  Pesona Tanaman Hias<P>
  Keindahan tanaman hias<BR>
  membawa suatu pesona tersendiri<BR>
  Warna-warni bunga<BR>
  Aneka rupa corak daun<BR>
  membuat orang dengan senang hati mengoleksinya
</BODY>
</HTML>
```

Di dalam tag <P> bisa disertakan atribut **ALIGN** yang berguna untuk mengatur peletakan teks di dalam masing-masing baris. Salah satu nilai yang dapat diberikan pada ALIGN adalah CENTER, yang menempatkan teks berada di tengah-tengah layar pada baris yang bersangkutan. Selain CENTER, nilai lain yang dapat diisikan pada ALIGN yaitu :

- **LEFT**, mengatur teks rata kiri terhadap halaman
- **RIGHT**, mengatur teks rata kanan terhadap halaman
- **CENTER**, Mengatur teks rata tengah terhadap halaman
- **JUSTIFY**, mengatur teks rata kiri dan rata kanan. Efeknya terlihat untuk teks yang sangat panjang.

Contoh : **palign.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Contoh tag P dengan ALIGN</TITLE>
</HEAD>
<BODY>
    PUISIKU:
    <P ALIGN = "CENTER">
    Mataharipun mulai terbangun dari mimpi<BR>
    menyemburatkan sinarnya yang hangat<BR>
    Burung-burung berkicau riang<BR>
    mengucapkan selamat pagi<BR>
    </P>
    Yogyakarta, 2002
</BODY>
</HTML>

```

#### 1.2.4 Tag <HR>

Untuk mempercantik halaman, seringkali pembuat dokumen Web menambahkan garis horisontal. Garis ini sebenarnya dapat dibuat dengan mudah, yakni dengan menyertakan tag<HR>.

Contoh : **horisontalruler.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Tag HR</TITLE>
</HEAD>
<BODY>
    <H2>Bahasa Pemrograman</H2>
    <HR>
    Visual Basic
    <HR>
    PHP
    <HR>
    Java
    <HR>
    dll
    <HR>
</BODY>
</HTML>

```

Table 1.1 Daftar atribut tag &lt;HR&gt;

Atribut	Value	Keterangan
SIZE	<i>pixels</i>	Menentukan ketebalan garis
WIDTH	<i>Pixels , %</i>	Menentukan lebar garis
ALIGN	Left, Center, Right	Menentukan peletakkan teks dalam baris
NOSHADA	noshade	Mengatur agar garis tidak disertai bayangan

Contoh penggunaan atribut SIZE : **sizehr.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Atribut SIZE pada Tag HR</TITLE>
</HEAD>
<BODY>
  <H1>KEBUN PESONA</H1>
  <HR SIZE = "10">
  Jl. Solo Km 14<BR>
  Yogyakarta<BR>
  Indonesia<BR>
</BODY>
</HTML>

```

Contoh penggunaan atribut NOSHADA : **noshadehr.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Atribut NOSHADA pada Tag HR</TITLE>
</HEAD>
<BODY>
  <H1>KEBUN PESONA</H1>
  <HR SIZE = "10" NOSHADA>
  Jl. Solo Km 14<BR>
  Yogyakarta<BR>
  Indonesia<BR>
</BODY>
</HTML>

```

Contoh penggunaan atribut WIDTH : **widhthr.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Atribut WIDTH dan ALIGN pada Tag HR</TITLE>
</HEAD>
<BODY>
  <HR ALIGN = "CENTER" WIDTH = "5%" SIZE = "10" NOSHADE>
  <HR ALIGN = "CENTER" WIDTH = "10%" SIZE = "10" NOSHADE>
  <HR ALIGN = "CENTER" WIDTH = "20%" SIZE = "10" NOSHADE>
  <HR ALIGN = "CENTER" WIDTH = "35%" SIZE = "10" NOSHADE>
  <HR ALIGN = "CENTER" WIDTH = "20%" SIZE = "10" NOSHADE>
  <HR ALIGN = "CENTER" WIDTH = "10%" SIZE = "10" NOSHADE>
  <HR ALIGN = "CENTER" WIDTH = "4%" SIZE = "10" NOSHADE>
</BODY>
</HTML>

```

### 1.2.5 Tag <CENTER>

Untuk menengahkan suatu teks, bisa juga digunakan tag <CENTER>. Tentu saja, untuk mengakhiri penengahan teks (yakni agar teks berikutnya tidak ditengahkan), perlu disertakan </CENTER>.

Contoh : **center.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Tag CENTER</TITLE>
</HEAD>
<BODY>
<CENTER>
  <H2>Berbagai Jenis Keladi yang mempesona:</H2>
  Red Flash  <BR>
  Red Fire   <BR>
</CENTER>
</BODY>
</HTML>

```

### 1.2.6 Tag <FONT>

Format teks pada dokumen HTML dapat menampilkan berbagai jenis font, ukuran font, dan warna font pada layar. Atribut-atribut yang digunakan pada <FONT> adalah SIZE, FACE, dan COLOR. Tag <FONT> ini merupakan tag berpasangan, oleh sebab itu harus diakhiri dengan tag slash-nya </FONT>.

**<FONT SIZE="x" FACE="y" COLOR="z"> teks </FONT>**

#### Dimana :

- x = ukuran font (1-7)
- y = jenis font (Comic Sans MS, Arial, Courier, dll)
- z = warna font (red, blue, green)

#### Beberapa Contoh Warna :

Nama Warna	Hexsa	Nama Warna	Hexsa
Antiquewhite	FAEBD7	Gold	FFD700
Aqua	00FFFF	Gray	808080
Black	000000	Green	008000
Blue	0000FF	Greenyellow	ADFF2F
Brown	A52A2A	Honeydew	FOFFFO
Coral	FF7F50	Magenta	FF00FF
Cyan	00FFFF	Orange	FFA500
Darkgreen	006400	Skyblue	87CEEB

Contoh penggunaan Tag Font : **font.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Tag Font</TITLE>
</HEAD>
<BODY>
  <FONT SIZE="6" FACE="Courir" COLOR="blue">STMIK Buddhi
    Masih dalam tag font

  </FONT>
  Di luar tag font
  <BR>

```

### 1.2.7 Format Text

Terdapat beberapa Tag yang dapat digunakan untuk Text Formatting, berikut contoh penggunaannya.

Tag	Keterangan
<b>&lt;b&gt;</b>	Untuk mencetak huruf tebal
<b>&lt;em&gt;</b>	Untuk mencetak huruf yang diberi penekanan
<b>&lt;i&gt;</b>	Untuk mencetak huruf miring
<b>&lt;small&gt;</b>	Untuk mencetak huruf tebal kecil
<b>&lt;strong&gt;</b>	Untuk mencetak huruf tebal
<b>&lt;sub&gt;</b>	Untuk mencetak huruf Subscript
<b>&lt;sup&gt;</b>	Untuk mencetak huruf Superscript
<b>&lt;ins&gt;</b>	Untuk mendefinisikan huruf yang input
<b>&lt;del&gt;</b>	Untuk mendefinisikan huruf yang dihapus
<b>&lt;mark&gt;</b>	Untuk menandai huruf

Contoh penggunaan Tag Font : **formattext.html**

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Format Pada Text</TITLE>
</HEAD>
<BODY>
  <p><b>This text is bold</b></p>
  <p><strong>This text is strong</strong></p>
  <p><i>This text is italic</i></p>
  <p><em>This text is emphasized</em></p>
  <p><code>This is computer output</code></p>
  <p>This is<sub> subscript</sub> and <sup>superscript</sup></p></BODY>
</HTML>
```

### 1.2.8 Pewarnaan Background <BODY>

Web site yang dibuat, kurang menarik jika belum disisipkan warna pada latar belakang dari sebuah dokumennya. Pewarnaan pada latar belakang dokumen dapat digunakan atribut pada <BODY>, yaitu atribut BGCOLOR. Adapaun warna yang dapat digunakan bisa dilihat pada tag Font di atas. Pembuatan warna background haruslah disesuaikan dengan warna teksnya. Jangan sampai Web site yang dibuat, nantinya akan menampilkan tulisan-tulisan yang tidak dapat dibaca.

Selain atribut BGCOLOR, ada juga atribut BACKGROUD untuk memberikan latar belakang berupa sebuah image/picture.

Contoh penggunaan Tag Font : **body.html**

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Pemberian Warna background</TITLE>
</HEAD>
<BODY Background="tes.jpg" >
    <FONT SIZE="3" FACE="Courier" Color="magenta">Teks dengan warna
    magenta dan menggunakan warna background honeydew </FONT><BR>
</BODY>
</HTML>
```

### 1.3 Latihan

Buatlah dokumen HTML dengan ketentuan sebagai berikut :

- Judul pada Title Bar **Latihan Pertama.**
- Judul pada dokumen **Daftar Riwayat Hidup.**
- Isi dokumen tuliskan Daftar Riwayat Hidup masing-masing.
- Berikan warna pada background
- Berikan tag2 lain yang sudah dipelajari di atas.

### 1.4 Link <A>

Untuk membuat link ke halaman lain, digunakan pasangan tag <A>...</A> atau yang biasa yang disebut tag jangkar atau anchor tag. Bentuk paling sederhana penggunaan tag jangkar adalah sebagai berikut :

```
<A HREF = "URL">Label</A>
```

Dalam hal ini, URL (Uniform Resource Locator) dapat berupa alamat suatu dokumen Web, berkas grafik, ataupun menyatakan suatu protokol lain (misalnya FTP).

Contoh : **link.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE> Penggunaan Link </TITLE>
</HEAD>
<BODY>
    <A HREF="http://www.yahoo.com"> klik disini </A>
    untuk menuju situs yahoo.com.
</BODY>
</HTML>

```

Untuk mempraktikan link, buatlah terlebih dulu dua buah berkas HTML bernama **halx.htm** dan **haly.htm**.

File **halx.htm** :

```

<HTML>
<HEAD>
<TITLE> Halaman X </TITLE>
</HEAD>
<BODY>
    <H1> Halaman X </H1>
</BODY>
</HTML>

```

File **haly.htm** :

```

<HTML>
<HEAD>
<TITLE> Halaman Y </TITLE>
</HEAD>
<BODY>
    <H1> Halaman Y </H1>
</BODY>
</HTML>

```

Sekarang marilah membuat kode HTML yang menyediakan link ke dua buah kode HTML di depan.

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE> Halaman Utama </TITLE>
</HEAD>
<BODY>
  <H1>Halaman Utama</H1>
  Silakan Klik pada link-link berikut : <BR>
  <A HREF="halx.html">Halaman X</A><BR>
  <A HREF="haly.html">Halaman Y</A><BR>
</BODY>
</HTML>

```

#### 1.4.1 Link ke Situs Web Lain

Hyperlink tak terbatas ke halaman-halaman Web yang berada pada sistem yang sama. Hyperlink seringkali ditujukan ke situs Web yang lain yang berada di sembarang Web server.

Contoh : **hyperlink.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE> Halaman Utama </TITLE>
</HEAD>
<BODY>
  <CENTER>
    <H1>Search Engine</H1>
    <BR>
    <A HREF="google.com">Google</A><BR>
    <A HREF="yahoo.com">Yahoo</A><BR>
  </CENTER>
</BODY>
</HTML>

```

Untuk membuat link tanpa garis bawah, tambahkan atribut

**style="text-decoration:none"**

### 1.4.2 Menentukan Warna Hyperlink

Label yang menjadi penghubung ke halaman Web lain secara bawaan diberi warna tertentu (umumnya biru) dan diberi garis bawah. Warna tersebut dapat diubah dengan menggunakan atribut LINK yang terdapat pada tag <BODY>.

Contoh :       **warnalink.html**

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE> Warna Hyperlink </TITLE>
</HEAD>
<BODY Link = "Black">
  <CENTER>
    <H1>Search Engine</H1>
    <BR>
      <A HREF="halx.html">Halaman X</A><BR>
      <A HREF="haly.html">Halaman Y</A><BR>
    </CENTER>
  </BODY>
</HTML>
```

Selain LINK, terdapat dua atribut lagi yang berkaitan dengan pengaturan warna hyperlink :

- **ALINK** berguna untuk menentukan link aktif, yaitu link yang telah diklik pemakai tetapi halaman belum selesai dibuka
- **VLINK** berguna untuk menentukan warna link yang halaman Web terkaitnya telah dikunjungi

Contoh :

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE> Halaman Utama </TITLE>
</HEAD>
<BODY Link = "Black" Alink = "Green" Vlink="Pink">
  <CENTER>
    <H1>Search Engine</H1>
    <BR>
      <A HREF="halx.html">Halaman X</A><BR>
      <A HREF="haly.html">Halaman Y</A><BR>
    </CENTER>
  </BODY>
</HTML>
```

### 1.4.3 Membuka Jendela Baru

Kadangkala dikehendaki agar kalau suatu link diklik maka jendela baru akan ditampilkan untuk menampung halaman yang di-link. Hal ini bisa dikerjakan dengan menambahkan kode :

**TARGET = “\_blank” pada tag <A>**

#### Contoh : target.html

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE> Halaman Utama </TITLE>
</HEAD>
<BODY Link = “Black” Alink = “Green” Vlink=“Pink”>
  <CENTER>
    <H1>Search Engine</H1>
    <BR>
    <A HREF=“halx.html” TARGET = “_BLANK”>Halaman X</A><BR>
    <A HREF=“haly.html” TARGET = “_BLANK”>Halaman Y</A><BR>
  </CENTER>
</BODY>
</HTML>
```

Selain **\_blank**, value dari target adalah : **\_parent**, **\_self**, **\_top**.

### 1.4.4 Membuat Link dalam satu Halaman

Link ke bagian lain dalam dokumen biasanya digunakan untuk dokumen yang sangat panjang, sehingga apabila ditampilkan dalam browser akan mengharuskan Anda melakukan scroll layar berulang-ulang. Navigasi untuk penelusuran dokumen dapat dimudahkan dengan membuat link antar bagian, dengan menandai setiap bagian tersebut dengan memberinya nama. Sehingga pada beberapa tempat di dalam dokumen akan ada bagian yang bernama, dan bagian lainnya dapat diletakkan link untuk menuju bagian-bagian tersebut.

Umumnya teknik ini diimplementasikan pada dokumen-dokumen yang menampilkan daftar isi dokumen tersebut, kemudian pada daftar tersebut dimungkinkan untuk diklik langsung menuju ke isinya secara langsung.

Cara memberi nama suatu bagian dalam dokumen :

1. Letakkan kursor pada baris atau teks yang menjadi awal dari bagian tersebut.
2. Sisipkan nama bagian tersebut dengan `<A NAME=“nama bagian”>`

Misalnya suatu halaman web mempunyai 3 paragraf. Jika ingin meloncat dari paragraf pertama ke paragraf ke tiga, maka hal pertama yang harus dilakukan adalah memberi nama untuk paragraf ke tiga tersebut, seperti di bawah ini:

**Contoh : linksatuh.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE> Link Dalam Satu Halaman </TITLE>
</HEAD>
<BODY>
  <P>Teks pada paragraf pertama <BR>
  <A HREF="#paragraf3">Langsung ke paragraf 3</A>
  <P>Teks pada paragraf kedua
  <P><A NAME="paragraf3">Paragraf ke 3</A>Teks pada paragraf ketiga</P>
</BODY>
</HTML>

```

**1.5 Gambar <IMG>**

Format gambar pada komputer sangat banyak dan beragam seperti BMP, GIF, JPEG, dll. Format gambar JPG/JPEG merupakan file yang sangat padat. File ini berukuran kecil meski gambarnya besar. Hal ini tentu saja akan menguntungkan bagi pengunjung web site, dimana kecepatan yang lebih baik sewaktu loading akan tercapai. Namun karena kompresi yang terlalu padat seperti itu akan membuat kualitas grafisnya kurang bagus.

**Contoh : gambar.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE> Contoh Gambar </TITLE>
</HEAD>
<body>
  Gambar matahari
  terbenam</img>
</body>
</HTML>

```

\*) Untuk memudahkan dalam pemanggilan gambar yang ingin ditampilkan pada halaman web, letakkan gambar dalam folder yang sama dengan halaman web. Jika gambar berada pada lokasi yang berbeda, maka harus disebutkan lengkap alamat (PATH) dari gambar tersebut.

**1.5.1 Atribut Gambar**

Tag <IMG> mempunyai beberapa atribut yang digunakan untuk memformat gambar, yaitu :

Table 1.2 Daftar atribut tag &lt;IMG&gt;

Atribut	Value	Keterangan
ALIGN	<i>Top, bottom, middle, left, right</i>	Menentukan peletakan posisis gambar
ALT	<i>Text</i>	Text alternatif jika gambar tidak tampl
BORDER	<i>pixels</i>	Memberikan garis disekitar gambar.
HEIGHT	<i>pixels</i>	Menentukan tinggi gambar
WIDTH	<i>pixels</i>	Menentukan lebar garis
HSPACE VSPACE	<i>pixels</i>	Meberikan spasi antara teks dan gambar
SRC	<i>url</i>	Menentukan lokasi dari gambar

**Contoh : gambarlengkap.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE> Contoh Gambar lengkap dengan atribut</TITLE>
</HEAD>
<body>
  Gambar
  matahari terbenam</img>
</body>
</HTML>

```

**1.6 List <OL> dan <UL>**

Daftar atau List digunakan untuk menguraikan daftar sesuatu, misalnya jenis-jenis list yang terdapat pada dokumen HTML tidak berbeda jauh dengan list-list yang berada pada dokumen MS. WORD, yaitu :

1. List dengan nomor (Ordered List / Numbering)
2. List tanpa nomor/dengan suatu lambang (Unordered List / Bullet)
3. List definisi

List dengan nomor adalah model daftar yang setiap itemnya diberi nomor. Sedangkan list tanpa nomor setiap itemnya menggunakan sebuah lambang/bullet. Selain kedua list tadi, ada juga list yang memberi uraian terhadap suatu item dalam daftar yang disebut list definisi.

### 1.6.1 List dengan Nomor / OrderedList / OL

Seperti yang telah diuraikan di atas, bahwa list dengan nomor selalu menggunakan angka-angka 1, 2, 3, 4, 5, .....dst sampai dengan namanya jumlah item dalam list tersebut. Dalam penomorannya, tidaklah harus menggunakan format nomor seperti di atas, tetapi dapat juga digunakan format penomoran lainnya, seperti huruf kecil, huruf kapital angka romawi kecil ataupun angka romawi besar.

Secara umum penggunaan OL, dapat dilihat seperti di bawah ini :

#### Contoh : OrderedList.html

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE> OrderedList </TITLE>
</HEAD>
<BODY>
  <H3>DAFTAR MAHASISWA Teknik Informatika</H3>
  <OL>
    <LI>Ardhi Sasongko
    <LI>Endah Mawar Indah
    <LI>Sandy Nayoan
    <LI>Dewi Aritonang
  </OL>
</BODY>
</HTML>

```

Selain format standar penomoran di atas, dapat juga dilakukan penomoran seperti yang ditunjukkan oleh tabel berikut ini :

Table 1.3 Format Penimoran List <IMG>

Jeni Penomoran	Keterangan
<b>A</b>	Ditampilkan menggunakan abjad huruf besar/kapital
<b>a</b>	Ditampilkan menggunakan abjad huruf kecil
<b>I</b>	Ditampilkan menggunakan angka romawi besar/kapital
<b>i</b>	Ditampilkan menggunakan angka romawi besar/kapital

Format penomoran di atas digunakan pada atribut TYPE pada tag <OL>. Jika tidak digunakan atribut TYPE, maka default format penomoran adalah menggunakan angka biasa (1, 2, 3, .....). Selain TYPE, atribut yang dimiliki oleh tag <OL> adalah atribut START. Atribut START digunakan untuk mendefinisikan nomor awal yang akan digunakan pada ORDERED LIST <OL>.

**Contoh : OrderedList2.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE> OrderedList </TITLE>
</HEAD>
<BODY>
  <H3>NAMA-NAMA BUNGA</H3>
  <OL TYPE = "A" START="2">
    <LI>Mawar
    <LI>Melati
    <LI>Anggrek
    <LI>Nusa Indah
    <LI>Kamboja
  </OL>
</BODY>
</HTML>

```

**1.6.2 List Tanpa Nomor / UnOrderedList / UL**

List tanpa nomor dapat ditampilkan dengan format/bullet standar berupa noktah/bulatan. Jenis ini dapat diubah menjadi disk (bentuk noktah tetapi tidak diarsir tengahnya dan bentuk kotak/box.

Untuk dapat mengubah bentuk bulletnya dapat digunakan atribut TYPE pada tag <UL>, yang penggunaannya sama dengan tag <OL>. Berbeda dengan tag <OL>, tag <UL> tidak mempunyai atribut start.

**Contoh : UnOrderedList.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE> UnOrderedList </TITLE>
</HEAD>
<BODY>
  <H3>NAMA-NAMA BUNGA</H3>
  <UL>
    <LI>Mawar
    <LI>Melati
    <LI>Anggrek
    <LI>Nusa Indah
    <LI>Kamboja
  </UL>
</BODY>
</HTML>

```



## 1.8 Latihan

1. Buatlah halaman web menggunakan tag Images (Gunakan gambar yang ada di computer). Lengkapi dengan atribut alt.



2. Buatlah halaman web seperti di bawah ini dengan menggunakan tag list.

### **The Best Features of the Internet**

- You can meet new people from countries around the world.
- You have access to new media as it becomes public:
  - New games
  - New applications
    1. For business
    2. For pleasure
  - Around the clock news
  - Search engines
  - Shopping
  - Programming
    1. XML
    2. Java
    3. XHTML
    4. Scripts
    5. New languages
- Links
- Keeping in touch with old friends
- It is the technology of the future!

3. Buat dokumen HTML menu yang di dalamnya terdiri atas teks dan gambar.

- 1. menu.htm**

Bila yang dipilih kata "hewan1" atau "gambar hewan1" (berupa file gambar) maka link akan beralih ke `hewan1.htm`.

Bila yang dipilih kata "hewan2" atau "gambar hewan2" (berupa file gambar) maka link akan beralih ke `hewan2.htm`

- 2. hewan1.htm** Berisi teks dan gambar dengan sebuah cerita pendek tentang hewan1.

Font yang digunakan adalah **verdana** dengan **ukuran = 4**. Pada akhir cerita, cantumkan teks : *kembali ke menu*. Link teks tersebut dengan **menu.htm**

- 3. hewan2.htm** Berisi teks dan gambar dengan sebuah cerita pendek tentang hewan2.

Font yang digunakan adalah **tahoma** dengan **ukuran = 2**. Pada akhir cerita, cantumkan teks : *cerita tentang hewan1*. Link teks tersebut dengan `hewan1.htm`.

## Pertemuan 2

### HTML Lanjutan

#### 2.1 Tabel <Table>

Tabel digunakan untuk menyajikan data dalam bentuk kolom dan baris. Umumnya setiap kolom menunjukkan data yang sejenis, dan setiap baris terdiri atas kolom-kolom yang menunjukkan kelompok data dalam satu kesatuan.

Untuk membuat sebuah tabel, diperlukan tiga buah tag yang berpasangan yaitu :

1. <TABLE>...</TABLE> : Digunakan untuk mendefinisikan sebuah tabel
2. <TR> ... </TR> : Disebut juga *Table Row* digunakan untuk membuat sebuah baris
3. <TH> ... </TH>: Disebut juga *Table Heading* digunakan untuk membuat judul kolom
4. <TD> ... </TD> : Disebut juga *Table Data* digunakan untuk membuat kolom dan data.

Pola dasar dalam pembentukan tabel adalah sebagai berikut :

```
<BODY>
<TABLE>
  <TR>
    <TD>Tabel 1 baris 1 kolom</TD>
  </TR>
</TABLE>
</BODY>
```

Contoh : **table1.html**

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE> Tabel satu baris dua kolom </TITLE>
</HEAD>
<BODY>
<TABLE>
  <TR>
    <TD>Ko11</TD>
    <TD>Ko12 </TD>
  </TR>
</TABLE>
</BODY>
</HTML>
```

### 2.1.1 Memberi Garis pada Tabel

Hasil dari file tabel1.htm menunjukkan bahwa tabel tidak menampilkan garis pembatas/garis tepi. Bila ingin memberikan garis tersebut maka perlu ditambahkan atribut BORDER pada tag <TABLE>. Atribut BORDER mempunyai nilai 1, 2, 3....dst yang memberikan efek ketebalan garis.

Perhatikan penggunaan BORDER pada file **tabel2.htm** berikut ini :

Contoh : **table2.html**

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE> Tabel satu baris dua kolom </TITLE>
</HEAD>
<BODY>
<TABLE BORDER="1">
  <TR>
    <TD>Kol1</TD>
    <TD>Kol2</TD>
  </TR>
</TABLE>
</BODY>
</HTML>
```

Jika diinginkan pembuatan tabel dengan beberapa baris dan kolom, maka ditambahkan beberapa pasang tag <TR> dan beberapa pasang tag <TD> sebagai berikut :

Contoh : **table3.html**

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE> Tabel satu baris dua kolom </TITLE>
</HEAD>
<BODY>
<TABLE BORDER="1">
  <TR>
    <TD>Baris1Kol1</TD>
    <TD>Baris1Kol2</TD>
  </TR>
  <TR>
    <TD>Baris2Kol1</TD>
    <TD>Baris2Kol2</TD>
  </TR>
</TABLE>
</BODY>
</HTML>
```

### 2.1.2 Meletakkan Judul Tabel

Judul tabel dapat dibuat dengan menambahkan tag `<CAPTION>` dan `</CAPTION>`. Tag ini diletakkan setelah `<TABLE>`. Letak judul ini adalah tepat di atas tabelnya. Untuk mengatur tata letak judul tabel, dapat digunakan atribut `ALIGN` dengan nilai "top" jika judul yang diinginkan berada di atas tabel. Dan menggunakan nilai "bottom" jika judul yang diinginkan berada di bawah tabel.

Contoh : **table4.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE> Tabel Dua Baris Dan Dua Kolom </TITLE>
</HEAD>
<BODY >
<TABLE BORDER="1">
<CAPTION ALIGN="top">TABEL DUA BARIS DAN DUA KOLOM</CAPTION>
  <TR>
    <TD>Baris1Kol1</TD>
    <TD>Baris1Kol2</TD>
  </TR>
  <TR>
    <TD>Baris2Kol1</TD>
    <TD>Baris2Kol2</TD>
  </TR>
</TABLE>
</BODY>
</HTML>

```

Selain judul tabel, biasanya baris dan kolom juga mempunyai judul. Untuk membuat judul kolom atau judul baris cukup mengganti tag `<TD>` .. `</TD>` dengan tag `<TH>` .. `</TH>` (*Table Heading*). Perhatikan contoh berikut :

Contoh : **table5.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Judul baris dan Judul Kolom </TITLE>
</HEAD>
<BODY>
<TABLE BORDER>
  <TR>
    <TH><BR></TH>
    <TH>JUDUL KOLOM1</TH>
    <TH>JUDUL KOLOM2</TH>
    <TH>JUDUL KOLOM3</TH>
  </TR>
  <TR>
    <TH>JUDUL BARIS1</TH>
    <TD>Isi Tabel</TD>
    <TD>Isi Tabel</TD>
    <TD>Isi Tabel</TD>
  </TR>
  <TR>
    <TH>JUDUL BARIS2</TH>
    <TD>Isi Tabel</TD>
    <TD>Isi Tabel</TD>
    <TD>Isi Tabel</TD>
  </TR>

```

Sel kosong pada baris kesatu dan kolom kesatu merupakan hasil dari tag <BR> yang diberikan di antara sepasang tag <TH> yang pertama.

### 2.1.3 Mengatur Posisi Tabel

Biasanya jika tabel ditampilkan di dalam sebuah browser, maka posisi tabel secara default akan berada di sebelah kiri dalam jendela browser. Selain posisi tabel berada disebelah kiri, tabel juga dapat diletakkan di tengah jendela browser. Untuk dapat meletakkan tabel di tengah-tengah jendela browser, cukup menambahkan sepasang tag <CENTER> sebelum tag <TABLE>. Perhatikan contoh berikut :

Contoh : **table6.html**

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE> Letak tabel di tengah browser </TITLE>
</HEAD>
<BODY>
<CENTER>
<TABLE BORDER="1">
  <TR>
    <TD>Kol1</TD>
    <TD>Kol2</TD>
  </TR>
</TABLE>
</CENTER>
</BODY>
</HTML>
```

#### 2.1.4 Cellspacing dan Cellpadding

Tag <TABLE> juga mempunyai atribut CELLSPACING dan CELLPADDING. Kedua atribut ini masing-masing mempunyai nilai dalam satuan pixel. Atribut CELLSPACING digunakan untuk mengatur spasi/jarak diantara kolom dalam tabel dan antara kolom dengan batas luar. Sedangkan atribut CELLPADDING berguna untuk mengatur teks dengan batas dalam. Perhatikan contoh berikut ini :

Contoh : **table7.html**

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>CellSpacing & CellPadding </TITLE>
</HEAD>
<BODY>
<TABLE BORDER="10" CellSpacing="10" CellPadding="15">
  <TR>
    <TD>Baris1Kol1</TD>
    <TD>Baris1Kol2</TD>
  </TR>
  <TR>
    <TD>Baris2Kol1</TD>
    <TD>Baris2Kol2</TD>
  </TR>
  <TR>
    <TD>Baris3Kol1</TD>
    <TD>Baris3Kol2</TD>
  </TR>
</TABLE>
</BODY>
</HTML>
```

### 2.1.5 RowSpan dan ColSpan

Anda seringkali melihat bahwa baris atau kolom tabel dapat digabungkan seperti baris atau kolom tabel yang dibuat pada MS. EXCEL. Untuk menggabungkan beberapa buah baris menjadi satu, cukup menambahkan atribut ROWSPAN pada tag <TD> atau tag <TH>. Sedangkan untuk menggabungkan beberapa buah kolom menjadi satu kolom, cukup menambahkan atribut COLSPAN pada tag <TD> atau tag <TH>. Atribut ROWSPAN dan COLSPAN masing-masing mempunyai nilai 1, 2, 3, .....dst. Artinya, jika ingin menggabungkan dua buah baris menjadi satu baris atau 3 buah kolom menjadi satu kolom maka nilai atribut ROWSPAN="2" dan nilai atribut COLSPAN="3". Perhatikan contoh berikut untuk penggabungan baris :

Contoh : **table8.html**

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Penggabungan 2 baris jadi 1 baris </TITLE>
</HEAD>
<BODY>
<CENTER>
<TABLE BORDER="1">
  <TR>
    <TD ROWSPAN="2">Baris1Kol1 dan Baris2kol1</TD>
    <TD>Baris1Kol2</TD>
  </TR>
  <TR>
    <TD>Baris2Kol1</TD>
  </TR>
  <TR>
    <TD>Baris3Kol1</TD>
    <TD>Baris3Kol2</TD>
  </TR>
</TABLE>
</CENTER>
</BODY>
</HTML>
```

Pada contoh tabel8.htm, terjadi penggabungan baris sebanyak dua baris, yaitu pada baris pertama kolom pertama dan baris kedua kolom pertama. Selanjutnya akan digabungkan beberapa kolom menjadi satu kolom. Perhatikan contoh berikut ini :

Contoh : **table9.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Penggabungan 2 kolom menjadi 1 kolom </TITLE>
</HEAD>
<BODY>
<CENTER>
<TABLE BORDER="1">
  <TR>
    <TD>Kol1</TD>
    <TD>Kol2</TD>
    <TD>Kol3</TD>
  </TR>
  <TR>
    <TD>Kol1</TD>
    <TD COLSPAN="2">Kol2 dan Kol3</TD>
  </TR>
</TABLE>
</CENTER>
</BODY>
</HTML>

```

### 2.1.6 Fungsi Atribut Width

Atribut WIDTH berguna untuk menentukan lebar dari tabel, proporsional dengan lebar window pada Browser. Width dapat menggunakan satuan piksel atau persentasi. Contohnya, WIDTH=75%, berarti lebar tabel adalah 75% dari lebar jendela browser yang tersedia. Atribut WIDTH dapat digunakan pada tag <TABLE>, <TH>, atau <TD>. Perhatikan contoh berikut ini :

Contoh : **table10.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Lebar Table dan Lebar Kolom </TITLE>
</HEAD>
<BODY>
<CENTER>
<TABLE BORDER="1">
  <TR>
    <TD WIDTH=30%>Lebar Kolom 30%</TD>
    <TD WIDTH=70%>Lebar Kolom 70%</TD>
  </TR>
  <TR>
    <TD>Lebar Kolom 30%</TD>
    <TD>Lebar Kolom 70%</TD>
  </TR>
</CENTER>
</BODY>
</HTML>

```

**2.1.7 Fungsi Atribut ALIGN dan VALIGN**

Elemen dalam tabel data dapat menggunakan atribut **ALIGN**, yang mengatur penempatan data. Atribut **ALIGN** mempunyai nilai "left" yang akan menempatkan data dari sebelah kiri (default), nilai "center" menempatkan data ditengah, nilai "right" menempatkan data pada sebelah kanan.

Selain atribut **ALIGN**, dapat juga menggunakan atribut **VALIGN** (*Vertical Alignment*) yang berguna untuk menentukan penempatan data dalam cell, dengan nilai "top" untuk di atas, nilai "middle" untuk di tengah, nilai "bottom" untuk di bawah.

**2.1.8 Pewarnaan Pada Tabel**

Warna berguna untuk memperjelas dan memperindah tampilan tabel. Warna biasanya digunakan dalam tabel tanpa menggunakan **BORDER**. Warna dinyatakan dalam atribut **BGCOLOR** (BackGround Color), yang dapat diisi dengan nilai heksadesimal atau langsung dengan nama warna yang telah didefinisikan oleh browser. Perhatikan contoh berikut :

Contoh : **table11.html**

```

<!DOCTYPE Html>
<HTML>
<HEAD> <TITLE>Pewarnaan Table </TITLE> </HEAD>
<BODY>
<CENTER>
<TABLE BORDER="1">
<TR>
  <TD BGCOLOR="Antiquewhite" WIDTH=40% ALIGN="Right">Baris1Kol1</TD>
  <TD BGCOLOR="Cyan"WIDTH=60% ALIGN="Center">Baris1Kol2</TD>
</TR>
<TR>
  <TD BGCOLOR="Skyblue">Baris2Kol1</TD>
  <TD BGCOLOR="Gray">Baris2Kol2</TD>
</TR>
<TR>
  <TD BGCOLOR="Green">Baris3Kol1</TD>
  <TD BGCOLOR="Pink">Baris3Kol2</TD>
</TR>
</TABLE>
</CENTER>
</BODY>
</HTML>

```

**2.1.9 Atribut NOWRAP**

Jika teks yang diisikan pada cell sangatlah panjang dan ingin ditampilkan dalam satu baris saja, maka gunakanlah atribut NOWRAP pada tag <TD> atau tag <TH>. Jika sudah diberikan atribut NOWRAP, maka penggunaan atribut WIDTH tidak akan berpengaruh. Jika panjang teks tersebut melebihi lebar tabel yang ditetapkan dengan atribut WIDTH maka tabel akan diperpanjang. Perhatikan contoh berikut ini :

Contoh : **table11.html**

```

<!DOCTYPE html>
<HTML>
<HEAD> <TITLE>Penggunaan Atribut Nowrap </TITLE> </HEAD>
<BODY>
<CENTER>
<TABLE BORDER="1">
  <TR>
    <TD WIDTH=20% NOWRAP>Teks teks Teks teks Teks teks Teks teks Teks
    teks</TD>
    <TD WIDTH=80%>Teks teks Teks teks Teks teks Teks teks</TD>
  </TR>
</TABLE>
</CENTER>
</BODY>
</HTML>

```

## 2.2 HTML Layout <Div>

<Div> adalah singkatan dari division yang berarti sebuah tag HTML untuk membuat suatu kelompok tertentu atau membuat suatu seksi yang bertujuan untuk pengelompokan sebuah file HTML menjadi beberapa bagian sesuai dengan tempatnya agar mempermudah para web developer memberi style pada setiap bagiannya.

Misalnya :

Sebuah halaman Web yang terdiri dari 4 bagian: Header, Menu, Content dan Footer. Untuk mengelompokannya kita dapat menggunakan tag div.

Contoh : **div.html**

```
<!DOCTYPE html>
<head>
<title>pengertian div tag</title>
</head>
<body>
<div id="container" style="width:500px">
  <div id="header" style="background-color:#FFA500;">
    <h1 style="margin-bottom:0;">Main Title of Web Page</h1>
  </div>

  <div id="menu" style="background- color:#FFD700;height:200px;width:100px;
float:left;">
    <b>Menu</b><br>
    HTML<br>
    CSS<br>
    JavaScript
  </div>

  <div id="content" style="background-color:#EEEEEE;height:200px;width:400px;
float:left;">
    Content goes here
  </div>

  <div id="footer" style="background-color:#FFA500;clear:both;text-align:center;">
    Copyright © W3Schools.com
  </div>
</div>
</body>
</html>
```

### 2.2.1 <Span>

HTML <span> elemen merupakan elemen inline yang dapat digunakan sebagai wadah untuk teks.<span> adalah elemen yang memiliki arti khusus. Ketika digunakan bersama-sama dengan CSS, <span> elemen dapat digunakan untuk mengatur atribut ke bagian teks yang berada di dalam span tersebut.

Penggunaan Tags <span> dalam HTML tidak akan memberikan efek pada elemen bila dilihat dalam browser, kecuali <span> diberikan style khusus seperti Inline CSS , ID atau CLASS selector, Contoh: (<span style =”... “>...</ span>). Tag <span> biasanya digunakan pada elemen-elemen yang lain di antara tag HTML.

Contoh :

```
<p>Buah Apel biasanya berwarna<span style=”color:a1ee55; font-size:1.6em; font-weight:bold;”>Hijau Muda</span></p>
```

Hasilnya :

Buah Apel biasanya berwarna **Hijau muda**

### 2.3 Latihan

1. Buatlah halaman Web yang berisikan table dengan menggunakan ColSpan dan RowSpan seperti tampilan di bawah ini :

Daftar Pramuniaga

Pramuniaga		KOTA		
		Jakarta	Tangerang	Bogor
Jenis Kelamin	Pria	30	20	30
	Wanita	20	8	18

2. Buatlah halaman Web yang berisikan table dengan menggunakan ColSpan, RowSpan dan Background Color seperti tampilan di bawah ini :

Picture of a camel	Camelid Comparison			
	# of Humps	Indigenous regic	Splits?	Produces Wool?
Camel	2	Africa/Asia	Llama	Llama
Llamas	1	Andes Mountains		

3. Buatlah desain halaman web dengan menggunakan div seperti di bawah ini :

<b>Website Logo</b>			< ~ Header ~ >
<u>Home</u>			< ~ Navigation bar ~ >
<u>Menu link</u> <u>Menu link</u> <u>Menu link</u> <u>Menu link</u> <u>Menu link</u>	<b>Page heading</b> <p>A three column layout with header, navigation bar and footer sections. The first, second and fourth table rows create the header, navigation bar and footer respectively and contain a single table cell each. All these table cells use colspan="3" attribute-value pair.</p> <p>The third table row contains three table cells which create the menu column (left), the content column (middle) and the extra column (right).</p>		
<b>Menu column</b>	<b>Content column</b>	<b>Extra column</b>	
			< ~ Footer ~ >

## Pertemuan 3

### HTML Formulir

#### 3.1 Dasar-Dasar Formulir

Pernakah Anda melihat tampilan sebuah halaman web berbentuk formulir yang digunakan untuk mengisi data-data ? Seperti misalnya dalam pembuatan email yang mewajibkan pengunjung web untuk mengisi data-datanya ? Bagaimana formulir tersebut nantinya akan diproses ? Anda pun dapat merancang sebuah formulir dengan menggunakan kode-kode HTML dan menentukan metode pengiriman informasi yang dimasukkan melalui formulir tersebut. Namun HTML mempunyai keterbatasan dalam pengolahan data-data yang dimasukkan ke dalam formulir tersebut.

Ada dua buah metode pengiriman yang digunakan dalam mengirim data yang telah diisikan ke dalam formulir :

##### 1. GET

Input yang didapat dari formulir akan dikirim ke server sebagai parameter bagi program yang menjalankannya.

##### 2. POST

Input yang didapat dari formulir akan dikirim ke server dan dianggap sebagai Standard Input (STDIN) untuk program yang memanggilnya.

Dengan adanya dua metode seperti di atas, metode POST mampu menyimpan data dalam jumlah banyak. Jika data yang diinput mempunyai banyak field pada suatu formulir, maka sangat cocok menggunakan metode POST karena metode ini dapat menangani data jauh lebih baik dari pada metode GET.

#### 3.2 Pembuatan Formulir

Sebuah form dibentuk dengan pasangan tag `<FORM>.....<FORM>`. Tag form ini memiliki dua buah atribut yaitu **METHOD** dan **ACTION**. Atribut **Method** digunakan untuk menunjukkan metode pengiriman informasi ke Web Server, apakah menggunakan metode GET atau POST. Atribut **Action** digunakan untuk menentukan kemana Informasi form akan diserahkan. Jika menggunakan CGI Script pada Web Server maka tentukanlah URL-script-nya.

Struktur dasar formulir adalah sebagai berikut :

```
<FORM METHOD="POST" ACTION="url-script">
<INPUT> → berfungsi untuk mendefinisikan kontrol-kontrol
</FORM>
```

### 3.2.1 Tipe Input

Tipe Input menyatakan suatu tipe kontrol yang akan digunakan. Macam-macam atribut tipe input yang dapat diolah oleh FORM antara lain :

1. Text Boxes
2. Radio Button
3. Check Boxes
4. Password Boxes
5. Submit Button dan Reset Button
6. Seleksi/Check List
7. Text Area
8. Hidden
9. Image

Pada umumnya Tag <INPUT> mempunyai atribut TYPE, NAME, VALUE, SIZE, dan MAXLENGTH. Misalnya :

```
<INPUT TYPE="text" NAME="nama" SIZE="25">
```

Dimana :

INPUT = Sebuah Tag

TYPE = Tipe Input

NAME = Definisi Variabel

nama = Nama Variabel

SIZE = Panjang Karakter

### 3.2.2 TextBox

Text Boxes digunakan untuk mengumpulkan data seperti nama orang, alamat, dan lain sebagainya. Perhatikan contoh berikut ini :

Contoh : **TextBox.html**

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>INPUT dengan Text Area </TITLE>
</HEAD>
<BODY>
<FORM METHOD="POST">
    Masukkan NIM :<INPUT TYPE="text" NAME="nim" SIZE="8">
    Masukkan NAMA :<INPUT TYPE="text" NAME="nama" SIZE="20">
    Masukkan ALAMAT :<INPUT TYPE="text" NAME="alamat" SIZE="30">
</FORM>
</BODY>
</HTML>
```

ALAMAT merupakan teks yang akan ditampilkan pada jendela browser. TYPE="text" merupakan tipe kontrol atau tipe input yang digunakan. Atribut NAME nim, nama, dan alamat adalah nama yang diberikan untuk kotak teks tersebut, nama ini akan dikembalikan

oleh browser sebagai identifier untuk informasi yang diminta oleh pemakai. Atribut SIZE merupakan ukuran panjang kotak teks dengan satuan karakter.

### 3.2.3 Radio Button

Radio Button digunakan bagi pengunjung web yang akan memilih satu pilihan dari banyak pilihan yang tersedia. Misalnya : Jenis Kelamin (Pria/Wanita).

Contoh : **Radio.html**

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>INPUT dengan Radio Button </TITLE>
</HEAD>
<BODY>
<FORM METHOD="POST">
  <H2>Jenis Kelamin :</H2><BR>
  <INPUT TYPE="radio" NAME="kelamin" VALUE="pria">Pria<BR>
  <INPUT TYPE="radio" NAME="kelamin" VALUE="wanita">Wanita
</FORM>
</BODY>
</HTML>
```

### 3.2.4 Check Box

Check Boxes digunakan bagi pengunjung web yang akan memilih satu atau lebih pilihan dari banyak pilihan yang tersedia. Misalnya : Hobby (Membaca, menari, memancing).

Contoh : **CheckBox.html**

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>INPUT dengan Check Boxes </TITLE>
</HEAD>
<BODY>
<FORM METHOD="POST">
<H3>Hobi :</H3><BR>
  <INPUT TYPE="Checkbox" NAME="membaca"
  VALUE="membaca">Membaca<BR> <INPUT TYPE="Checkbox"
  NAME="menyanyi" VALUE="menyanyi">Menyanyi<BR> <INPUT
  TYPE="Checkbox" NAME="menari" VALUE="menari">Menari<BR>
  <INPUTTYPE="Checkbox"NAME="memancing"VALUE="mamancing">Mema
  ncing<BR>
  <INPUT TYPE="Checkbox" NAME="olahraga" VALUE="olahraga">Olah
  Raga<BR>
</FORM>
</BODY>
</HTML>
```

### 3.2.5 Password Box

Password adalah teks yang dimasukkan sebagai input, namun teks tidak ditampilkan pada layar. Layar akan menampilkan tanda asteriks, atau bullet.

Contoh : **PasswordBox.html**

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>INPUT dengan Password Boxes </TITLE>
</HEAD>
<BODY>
Silakan Anda Login Terlebih Dahulu<BR>
<FORM METHOD="POST">
    LOGIN : <INPUT TYPE="Text" NAME="login" SIZE="20"><BR>
    PASSWORD : <INPUT TYPE="password" NAME="kode" SIZE="20"><BR>
</FORM>
</BODY>
</HTML>
```

### 3.2.6 Submit Button dan Reset Button

Submit Button berfungsi mengirimkan semua input yang sudah diberikan ke Server. Proses ini sesuai dengan atribut ACTION yang diberikan pada tag <FORM>. Sedangkan Reset Button berfungsi membersihkan semua input yang sudah diberikan.

Pada akhir sebuah formulir umumnya tombol SUBMIT dan RESET selalu ditampilkan sebagai berikut :

```
<INPUT TYPE="submit">
<INPUT TYPE="reset">
```

Contoh : **SubmitButton.html**

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE> SubmitButton </TITLE>
</HEAD>
<BODY>
<FORM METHOD="POST">
    Masukkan NIM :<INPUT TYPE="text" NAME="nim" SIZE="8">
    Masukkan NAMA :<INPUT TYPE="text" NAME="nama" SIZE="20">
    Masukkan ALAMAT :<INPUT TYPE="text" NAME="alamat"
    SIZE="30"><BR><BR> <INPUT TYPE="submit" VALUE=" OK "><INPUT
    TYPE="reset" VALUE="Cancel">
</FORM>
</BODY>
</HTML>
```

### 3.2.7 Seleksi / Check List

Seleksi adalah memilih nilai dari sebuah teks/string dari sebuah daftar singkat. Pilihan lebih dari satu akan digunakan atribut MULTIPLE dan jumlah pilihan dapat ditentukan dengan variabel SIZE.

```
<SELECT NAME="var">
  <OPTION>Pilihan1
  <OPTION>Pilihan2
  <OPTION>Pilihan3
</SELECT>
```

Dengan demikian, pemakai dapat memilih dan hasil pilihan akan dikirim ke server

Contoh : **CheckList.html**

```
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Pilihan – Pilihan </TITLE>
</HEAD>
<BODY>
<H2>Pekerjaan</H2>
<FORM METHOD="POST">
  <SELECT NAME="pekerjaan">
    <OPTION>Pelajar/Mahasiswa
    <OPTION>Pegawai Swasta
    <OPTION>Pegawai Negeri
    <OPTION>Wiraswasta
    <OPTION>Lain-lain
  </SELECT>
</FORM>
</BODY>
</HTML>
```

### 3.2.8 Text Area

Text Area adalah suatu area yang dapat diisi dengan teks berupa komentar, komplain, atau pesan apa saja. Biasanya teks yang ditampilkan berupa memo dan umumnya mempunyai karakter yang sangat panjang.

Struktur dasar TextArea adalah sebagai berikut :

```
<TEXTAREA COLS="x" ROWS="y">
  Isi Komentar
</TEXTAREA>
```

Contoh : **TextArea.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Komentar-Komentar</TITLE>
</HEAD>
<BODY>
<H2>Isikan pendapat anda di bawah ini</H2>
<FORM METHOD="POST">
  <TEXTAREA COLS="40" ROWS="5">
  </TEXTAREA>
</FORM>
</BODY>
</HTML>

```

### 3.2.9 Hidden Field

**Tag input** dengan **type hidden** adalah sebuah objek form khusus yang sesuai dengan nama tipenya, memang tersembunyi dan tidak dapat dilihat oleh user. Namun objek form ini menawarkan fleksibilitas yang dapat dimanfaatkan oleh programmer untuk menyisipkan suatu data untuk diproses.

**Type hidden** ini bisa digunakan untuk mengirim suatu data dari sebuah halaman ke halaman lain tanpa mengganggu bentuk form yang telah ada

Contoh : **HiddenField.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Hidden Field</TITLE>
</HEAD>
<BODY>
<FORM METHOD="POST">
  Nama: < INPUT TYPE ="text" NAME="nama"><br>
  <INPUT TYPE="hidden" NAME="negara" VALUE="Indonesia">
  < INPUT TYPE ="submit" VALUE="Submit"> </FORM>
</BODY>
</HTML>

```

### 3.2.10 Image

Mempunyai fungsi yang sama dengan submit button, tetapi mengambil tampilan/bentuk button sesuai dengan gambar yang diinginkan.

Contoh : **image.html**

```

<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Hidden Field</TITLE>
</HEAD>
<BODY>

```

### 3.3 Iframe

Iframe ini digunakan untuk menampilkan halaman web dalam halaman web  
 Syntax : `<iframe src=" URL "></iframe>`

#### Mengatur Tinggi dan Lebar

Tinggi dan atribut lebar digunakan untuk menentukan tinggi dan lebar dari iframe. Nilai atribut yang ditetapkan dalam pixel secara default, tetapi mereka juga dapat dalam persen (seperti "80%").

Contoh :

```
<iframe src="demo_iframe.htm" width="200" height="200"></iframe>
```

#### Hapus Perbatasan

Atribut frameborder menentukan apakah menggunakan border atau tidak untuk sekitar iframe.

Mengatur nilai atribut menjadi "0" untuk menghapus border:

Contoh :

```
<iframe src="demo_iframe.htm" frameborder="0"></iframe>
```

#### Menggunakan iframe sebagai Sasaran untuk sebuah Link

Iframe dapat digunakan sebagai kerangka target untuk link. Atribut target link harus mengacu pada nama atribut iframe :

Contoh :

```
<iframe src="demo_iframe.htm" name="iframe_a"></iframe>
<p><a href="http://www.w3schools.com"
target="iframe_a">W3Schools.com</a></p>
```

Contoh : **iframe.html**

```

<html>
<head>
<title>HTML Iframe</title>
</head>
<body>
<iframe src="url" name="iframe1"></iframe>
    <h2>iframe Target </h2>
<iframe src="formulir.html" name="iframe_a" frameborder="0"></iframe>
    <p><a href="url" target="iframe_a">W3Schools.com</a></p>
</body>
</html>

```

### 3.4 Latihan

1. Buatlah formulir seperti di bawah ini :

SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER BUDDHI	
Edit Formulir PMB	
No. PMB	: 201320036
No. Ujian	: <input type="text" value="201320036"/> <span style="color: red;">Cek Calon Mahasiswa</span>
Nama	: <input type="text" value="Christa Tri Wihardani"/>
No. Kwitansi	: <input type="text" value="20130180"/>
Status Awal Masuk	: <ul style="list-style-type: none"> <li><input type="radio"/> Pindahan</li> <li><input checked="" type="radio"/> Baru</li> <li><input type="radio"/> PSB</li> <li><input type="radio"/> Drop-in</li> <li><input type="radio"/> Asing</li> </ul>
	Jika PSSB, masukkan No. PSSB : <input type="text"/>
	Jika pindahan, masukkan No. Penyetaraan: <input type="text" value="0"/>
Jenis Kelamin	: <ul style="list-style-type: none"> <li><input checked="" type="radio"/> Pria</li> <li><input type="radio"/> Wanita</li> </ul>
Tempat Lahir	: <input type="text" value="Tangerang"/>
Tanggal Lahir	: <input type="text" value="06"/> <input type="text" value="Juni"/> <input type="text" value="1994"/>
Pilihan Program Studi	
Program	: <input type="text" value="REG-03 - MALAM"/>
Pilihan 1	: <input type="text" value="57401 - MANAJEMEN INFORMATIKA"/>
<input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>	

Data Pribadi (Sesuai KTP)	
Warga Negara	<input checked="" type="radio"/> Warga Negara Indonesia : <input type="radio"/> Warga Negara Asing Jika WNA, sebutkan: <input type="text"/>
Agama	: <input type="text" value="K - Katholik"/>
Status Sipil	: <input type="text" value="B - Belum Menikah"/>
Alamat Tinggal	: <input type="text" value="Jl M Toha km 3,1 No 117"/>
Kota	: <input type="text"/>
RT/RW	: <input type="text" value="002"/> / <input type="text" value="001"/>
Kode Pos	: <input type="text"/>
Propinsi	: <input type="text" value="Banten"/>
Negara	: <input type="text" value="Indonesia"/>
Telepon	: <input type="text" value="02155799247"/>
Telp. Bergerak	: <input type="text" value="089653377916"/>
Email	: <input type="text" value="sonyachrista@gmail.com"/>
<input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>	

2. Tampilkan Halaman Formulir di atas ke dalam **Iframe**.

# Pertemuan 4

## Dasar CSS dan HTML 5

### 4.1 CSS

CSS (**Cascading Style Sheet**) adalah standard pembuatan dan pemakaian style untuk dokumen terstruktur , CSS digunakan untuk mempersingkat penulisan tag HTML seperti font,color,text, dan table menjadi lebih ringkas sehingga tidak terjadi pengulangan tulisan.

#### 4.1.1 Keuntungan menggunakan CSS

- Memisahkan **presentastion** sebuah dokumen dari **content** document itu sendiri.
- Mempermudah dan Mempersingkat pembuatan dan pemeliharaan dokumen web
- Mempercepat proses rendering/pembacaan HTML.

#### 4.1.2 Cara penulisan CSS

##### 1. Inline style sheet

Penulisan didalam elemen HTML.

Contoh script 1

```
<html>
<head>
<title>Contoh : Inline style sheet </title>
</head>
<body>
    <p style='color: red ; font-size:200%' > Hello </p>
</body>
</html>
```

Hasil :

Hello

##### 2. Embedded Style Sheet

Penulisan CSS didalam dokumen HTML dan menggunakan tag <style></style>

```
<html>
<head>
<title>Contoh : Embedded style sheet</title>
<style>
    H1 {color:red}
</style>
</head>
<body>
    <H1> Hello </H1>
</body>
</html>
```

## 3. L

---

```

<html>
<head>
<title> Contoh : Linked style sheet</title>
<link href="style1.css" rel="stylesheet" type="text/css">
</head>
<body>
    <H1 >coba Linked style sheet </H1>
</body>
</html>

```

inked Style Sheet

Penulisan skrip CSS dihalaman berbeda atau terpisah dari html.

Contoh script3 :

Contoh script style1.css

---

```

H1
{
    color : RED;
}

```

#### 4.1.3 Aturan penulisan CSS

##### 1. Selector

Terdiri dari tag,class,ID

##### 2. Declaration

Mendeskripsikan property dan value

Contoh

```

H1
{
    Color : #0000FF
}

```

Keterangan :

Selector : H1

Properti : Color

Value : #0000FF

#### 4.1.4 Macam-macam Selector

##### 1. Tag/Elemen

Setiap tag yang ada pada HTML bisa dijadikan selector

##### 2. Class

Diawal penulisan menggunakan tanda titik, pada HTML ditambahkan class

style1.css

```
H1
{
  color : RED;
}

.isiteks {
  Font-family : Arial,Helvetica;
  Font-size : 12 px;
  Color : #6699FF;
}
```

Contoh script 4

```
<html>
<head>
<link href='style1.css' rel="stylesheet" type='text/css'>
</head>
<body>
  <p class='isiteks' > Contoh class </p>
</body>
</html>
```

### 3. ID

Diawali dengan tanda #, dapat digunakan untuk mendefinisikan header, content, dan footer dalam desain web krn didefinisikan dengan ID berbeda.

Style1.css

```
H1
{
  color : RED;
}

.isiteks
{
  Font-family : Arial,Helvetica;
  Font-size : 24 px;
  Color : #6699FF;
}

#judul1
{
  Font-family : Arial,Helvetica;
  Font-size : 12 px;
  Color : #6699FF;
}
```

## Contoh script5

```

<html>
<head>
<link href='style1.css' rel="stylesheet" type='text/css'>
</head>
<body>
    <p id='judul1'> Contoh ID </p>
</body>
</html>

```

**4.1.5 CSS Background**

## Property background

## 1. Background-color

```

Body {
    Background-color : green;
}

```

## 2. Background-Image

Contoh:

```

Body {
    Background-image : url(gambar1.jpg);
}

```

## 3. Background-attachment

```

Body {
    Background-attachment : scroll;
}

```

## 4. Background-Repeat

```

Body {
    Background-repeat : repeat;
    Background-image : url(image/gambar1.jpg);
}

```

## 5. Background-position

```

Body {
    Background-repeat : no-repeat;
    Background-image : url(image/gambar1.jpg);
    Background-position : bottom center
}

```

style1.css

```
H1
{
  color : RED;
}

.isiteks
{
  Font-family : Arial,Helvetica;
  Font-size : 24 px;
  Color : #6699FF;
}

#judul1
{
  Font-family : Arial,Helvetica;
  Font-size : 12 px;
  Color : #6699FF;
}

body
{
  background-color:black;
  color:gold;
}
```

Contoh script6

```
<html>
<head>
<link href='style1.css' rel="stylesheet" type='text/css'>
</head>
<body>
  Hello, Contoh Body
</body>
</html>
```



Hello, Contoh Body

#### 4.1.6 CSS Font

Property Font :

1. Font-family
 

```
P {
  Font-family : Arial,Helvetica;
}
```
2. Font-size (satuan em,pt,px,mm,cm,%)
 

```
P {
  Font-size : 24em;
}
```

3. Font-style (nilai : oblique,italic,normal)
 

```
P {
  Font-style : oblique;
}
```
4. Font-variant (nilai : normal,small-caps)
 

```
P {
  Font-family : Arial,Helvetica;
}
```
5. Font-weight (nilai :normal,bold,bolder,lighter,100-900)
 

```
P {
  Font-weight : bolder;
}
```

#### 4.1.7 CSS Teks

Property teks :

1. color
 

```
P {
  color :red;
}
```
2. teks-align (nilai left,right,center,justify)
 

```
P {
  Text-align :justify;
}
```
3. text-decoration (nilai : none,underline,overline,line-through,blink)
 

```
P {
  Text-decoration :overline;
}
```
4. text-transform (nilai : none,capitalize,uppercase,lowercase)
 

```
P {
  Text-transform : capitalize;
}
```
5. Letter-spacing
 

```
P {
  letter-spacing : 20px;
}
```
6. Teks-indent
 

```
P {
  Text-indent : 50px;
}
```

#### 4.1.8 CSS List

Property list :

1. List-style-type (nilai : disc,circle,square,lower-roman,upper-roman,none,lower-alpha,upper-alpha)
 

```
UI {
  List-style-type : disc ;
}
```
2. List-style-image (nilai : url,none)
 

```
UI {
  List-style-image : url(gambar1.jpg) ;
}
```
3. List-style-position (nilai : inside,outside)
 

```
UI {
  List-style-position : inside;
}
```

#### 4.1.9 Margin dan Padding

Padding hampir sama dengan margin :

- a. Margin hanya digunakan untuk membuat batasan-batasan sisi halaman.
- b. Padding dapat membuat batasan-batasan pada komponen web lain seperti tabel, disamping pengaturan batas halaman.

1. Margin : pengaturan batas atas,bawah,kanan,kiri pada halaman web
 

```
body {
  Margin-top : 4cm;
  Margin-right : 3cm;
  Margin-bottom : 3cm;
  Margin-left : 4cm;
}
```
2. Margin : pengaturan batas atas,bawah,kanan,kiri antara border dengan content.
 

```
.kotak {
  Padding-top : 25px;
  Padding-right : 25px;
  Padding-bottom : 25px;
  Padding-left : 25px;
  border-top-style: solid;
  border-right-style: solid;
  border-bottom-style: solid;
  border-left-style: solid;
}
```

**Pada file HTML :**

```

<html>
<head>
  <link rel="stylesheet" type="text/css" href="cssku.css">
  </style>
</head>
<body>
  <p class=kotak > teks1 <p>
</body>
</html>

```

**4.2 Contoh Penggunaan CSS****4.2.1 Padding dan Margin****paddingvsmargin.css**

```

div{
  background-color:#CC9;
  margin: 15px 0 15px 0;
  padding: 15px 15px 15px 15px;
}

```

**Paddingvsmargin.html**

```

<DOCTYPE html>
<html>
<head>
<title>CSS Id dan Class</title>

<link rel="stylesheet"
  href="paddingvsmargin.css" type="text/css">
</head>
<body>
  <div>
    <h1>Pindang Patin</h1>
    <p >Masakan khas dari Palembang,
    paling mantap disatap waktu hangat</p>
    <p>Maknyuss!!</p>
  </div>
</body>
</html>

```

## 4.2.2 Pembuatan Menu

### Menu.css

```
body{
    padding:50px;
}
#menu{
    font-family: Arial, san-serif;
    font-weight:bold;
    text-transform:uppercase;
    margin:50px 0;
    padding: 0;
    list-style-type:none;
    background-color:#eee;
    font-size: 13px;
    height:40px;
    border-top:2px solid #eee;
    border-bottom:2px solid #ccc;
}
#menu li{
    float:left;
    margin:0;
}
#menu li a{
    text-decoration:none;
    display:block;
    padding :0 20px;
    line-height:40px;
    color:#666;
}
#menu_wrapper{
    padding:0 16px 0 0;
}
#menu_wrapper li a{
    color: #000;
}
#menu_wrapper li a:hover{
    color: #ccc;
    background:#555;
    border-bottom: 2px solid #444;
}
```

**Menu.html**

```

<DOCTYPE html>
<html>
<head>
<title>CSS Menu</title>
<link rel="stylesheet"
      href="menu.css" type="text/css">

</head>
<body>
<div id="menu_wrapper">
  <ul id="menu">
    <li><a href="#">Home </a> </li>
    <li><a href="#">About </a> </li>
    <li><a href="#">Service </a> </li>
    <li><a href="#">Product </a> </li>
    <li><a href="#">Contact </a> </li>
  </ul>
</div>
</body>
</html>

```

**4.2.3 Membuat Layout****Layout.css**

```

#halaman{ background-color:#660099;
          min-width: 700px;
          height:200px;
        }
#atas {
  background-color:#CF0;
  height:100px;}
#kiri{
  background-color:#0C0;
  max-width:200px; float:left;
  height:400px;}
#kanan {
  background-color:#063;
  max-width:200px; float:right;
  height:400px;}
#bawah {
  background-color:#F00;
  height:100px;}
#tengah {
  background-color:#000000;
  height:400px;}

```

## Layout.html

```

<DOCTYPE html>
<html>
<head>
<title>CSS Layout</title>
<link rel="stylesheet"
      href="layout.css" type="text/css">

</head>
<body>
<div id="halaman">
  <div id="atas">bagian atas halaman </div>
  <div id="kiri"> bagian kiri halaman </div>
  <div id="tengah">
    <div id="kanan"> bagian kanan halaman </div>
    <div id="isi"> bagian isi halaman </div>
  </div>
  <div id="bawah">Bagian bawah halaman</div>
</div>
</body>
</html>

```

## 4.3 HTML 5

Berikut adalah tag-tag baru yang terdapat di HTML 5 :

Tag	Deksripsi
<article>	Mendefinisikan artikel
<aside>	Mendefinisikan isi selain dari konten utama
<audio>	Mendefinisikan suara/music
<bdi>	Membatasi bagian teks yang mungkin memiliki format baca berbeda dengan teks diluarnya
<canvas>	Digunakan untuk menggambar grafis melalui script
<command>	Mendefinisikan sebuah tombol komando (command button)
<datalist>	Menentukan daftar pilihan standar untuk kontrol input
<details>	Mendefinisikan detail tambahan yang dapat ditampilkan-sembunyikan oleh user
<embed>	Mendefinisikan sebuah aplikasi eksternal / plugins
<figcaption>	Mendefinisikan caption untuk elemen <figure>
<figure>	Menentukan sebuah konten mandiri
<footer>	Mendefinisikan sebuah footer pada dokumen atau section
<header>	Menentukan pengenalan awal halaman web atau kelompok dari elemen navigasi untuk dokumen
<hgroup>	Mengelompokkan elemen <h1> sampai <h6> ketika heading memiliki multiple level
<keygen>	Mendefinisikan sebuah key generator untuk form
<mark>	Mendefinisikan sebuah teks yang disorot/ditandai
<meter>	Mendefinisikan pengukur untuk ukuran skalar dalam kisaran yang diketahui
<nav>	Mendefinisikan link navigasi

<output>	Mendefinisikan sebuah hasil dari kalkulasi/perhitungan
<progress>	Mendefinisikan sebuah perkembangan jalannya proses
<q>	Mendefinisikan sebuah kalimat kutipan (quote)
<rp>	Mendefinisikan apa yang harus ditampilkan apabila browser tidak mensupport tag <ruby>
<ruby>	Menentukan sebuah penulisan huruf Asian (kanji, mandarin, dll)
<section>	Mendefinisikan section dalam dokumen
<source>	Mendefinisikan sumber elemen media yang beruntun ( <video> dan <audio> )
<summary>	Mendefinisikan heading untuk bagian <detail>
<time>	Mendefinisikan waktu (24 jam) atau tanggal dalam format Gregorian
<track>	Mendefinisikan subtitle dari elemen multimedia ( <audio> dan <video> )
<video>	Mendefinisikan video atau film
<wbr>	Mendefinisikan ganti baris ketika teks dirasa terlalu panjang

Beberapa Tag Input di HTML 5 :

#### **Input Type : Color**

Select your favorite color: <input type="color" name="favcolor">

#### **Input Type : Date**

Birthday: <input type="date" name="bday">

#### **Input Type : email**

E-mail: <input type="email" name="email">

#### **Input Type : number**

Quantity (between 1 and 5): <input type="number" name="quantity" min="1" max="5">

#### **Input Type : range**

<input type="range" name="points" min="1" max="10">

Tag-tag yang hilang di HTML 5 adalah :

- <acronym>
- <applet>
- <basefont>
- <big>
- <center>
- <dir>
- <font>
- <frame>
- <frameset>
- <noframes>
- <strike>
- <tt>

**Contohhtml5.html**

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>HTML5</title>
</head>
<body>

<header>
  <hgroup>
    <h1>Channel9</h1>
    <h2>HTML5 and CSS3 Fundamentals</h2>
  </hgroup>

  <nav>
    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">About</a></li>
      <li><a href="#">Contact</a></li>
    </ul>
  </nav>
</header>

<section>
<article>
<h1>Background Every Web Developer Needs to Know</h1>
<h2>Introduction</h2>
  Content.....
</article>
</section>

<footer>
<small>Copyright © 2012-2014 Bob Tabor</small>
  <nav>
    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">About</a></li>
      <li><a href="#">Contact</a></li>
    </ul>
  </nav>
</footer>

</body>
</html>

```

### 4.3.1 Tag Video

Pada versi HTML sebelumnya, tidak disertai oleh tag `<video>`. Apabila ingin memasukkan video ke dalam sebuah web masih harus menggunakan plugins tambahan yang biasa disebut **flash player**. Kelebihan di dalam tag `<video>` adalah dalam browser tidak perlu menggunakan flash player, dan buffering menjadi lebih ringan karena fasilitasnya sudah di backing oleh kode HTML sendiri.

#### Atribut Opsional :

Attribute	Value	Description
<b>autoplay</b>	autoplay	Menentukan bahwa video akan mulai berputar otomatis
<b>controls</b>	controls	Menentukan kontrol video yang akan ditampilkan (Contoh: tombol play/pause)
<b>height</b>	<i>pixels</i>	Mengeset tinggi dari video player
<b>loop</b>	loop	Menentukan bahwa video akan berputar kembali meski video telah habis
<b>muted</b>	muted	Menentukan bahwa output suara akan di mute / dihilangkan
<b>poster</b>	URL	Menentukan gambar yang akan tampil ketika video masih belum berputar
<b>preload</b>	auto metadata none	Menentukan jika dan bagaimana kapan video ini harus menjalani proses loading ketika halaman terbuka
<b>src</b>	URL	Menentukan alamat URL dari mana sumber video
<b>width</b>	<i>pixels</i>	Mengeset lebar dari video player

#### Contoh kode tag `<video>`

```

<!DOCTYPE html>
<html>
<body>
<video width="320" height="240" controls="controls" >
  <source src="bbb.mp4" type="video/mp4" />
  <source src="bbb.ogg" type="video/ogg" />
  <source src="bbb.webm" type="video/webm" />
  Your browser does not support the video tag.
</video>
</body>
</html>

```

### 4.3.2 Tag Audio

Digunakan untk memasukkan audio, misalnya lagu kedalam halaman web. Pada HTML5, hal itu dipermudah dengan tag `<audio>`, ketika di generasi sebelumnya (HTML 4.01) memasukkan audio ke dalam web masih harus menggunakan flash.

Attribute	Value	Description
<b>autoplay</b>	autoplay	Menentukan bahwa audio akan mulai berputar otomatis
<b>controls</b>	controls	Menentukan kontrol audio yang akan ditampilkan (Contoh: tombol play/pause)
<b>loop</b>	loop	Menentukan bahwa audio akan berputar kembali meski audiotelah habis
<b>preload</b>	auto metadata none	Menentukan jika dan bagaimana kapan audio ini harus menjalani proses loading ketika halaman terbuka
<b>src</b>	URL	Menentukan alamat URL dari mana sumber audio

```

<!DOCTYPE html>
<html>
<body>
<audio controls="controls">
  <source src="multimedia/song.ogg" type="audio/ogg" />
  <source src="multimedia/song.mp3" type="audio/mp3" />
  Your browser does not support the audio element.
</audio>
</body>
</html>

```



### 5.1.1 Syntax PHP

- Kode PHP disimpan sebagai plain text dalam format ASCII, sehingga kode PHP dapat ditulis hampir di semua editor text seperti windows notepad, windows wordpad, dll.
- Kode PHP adalah kode yang disertakan di sebuah halaman HTML dan kode tersebut dijalankan oleh server sebelum dikirim ke browser.
- Blok scripting PHP selalu diawali dengan `<?php` dan diakhiri dengan `?>`. Blok scripting PHP dapat ditempatkan dimana saja di dalam dokumen. Pada beberapa server yang mendukung, blok scripting PHP dapat diawali dengan `<?` dan diakhiri dengan `?>`. Namun, untuk kompatibilitas maksimum, sebaiknya menggunakan bentuk yang standar (`<?php ?>`).
- Setiap baris kode PHP harus diakhiri dengan semikolon (;). Semikolon ini merupakan separator yang digunakan untuk membedakan satu instruksi dengan instruksi lainnya.
- PHP menggunakan `//` untuk membuat komentar baris tunggal atau `/*` dan `*/` untuk membuat suatu blok komentar.

```
<?php
    echo "Hello World";
    //Ini adalah komentar satu baris
    /* Ini adalah komentar
    dua baris */
?>
```

### 5.1.2 Variable

Variabel digunakan untuk menyimpan suatu nilai, seperti text, angka atau array. Ketika sebuah variabel dibuat, variabel tersebut dapat dipakai berulang-ulang.

Pada PHP semua variabel harus dimulai dengan karakter '\$'. Variabel PHP tidak perlu dideklarasikan dan ditetapkan jenis datanya sebelum kita menggunakan variabel tersebut. Hal itu berarti pula bahwa tipe data dari variabel dapat berubah sesuai dengan perubahan konteks yang dilakukan oleh user. Secara tipikal, variabel PHP cukup diinisialisasikan dengan memberikan nilai kepada variabel tersebut.

```
<?php
    $text ="PHP"
    echo $text;
?>
```

#### Aturan Penamaan Variabel

1. Nama variabel harus diawali dengan sebuah huruf atau garis bawah (underscore) “\_”
2. Nama variabel hanya boleh mengandung karakter alpha-numeric dan underscore (a-Z, 0-9, dan \_)
3. Nama variabel tidak boleh mengandung spasi.

## 5.2 Operator pada PHP

Operator digunakan untuk menentukan sebuah nilai dengan melakukan suatu prosedur, atau dengan suatu operasi dari beberapa nilai. Nilai yang digunakan dalam sebuah operasi disebut operand. Penjumlahan adalah sebuah contoh sederhana dari suatu operasi.

Sebagai contoh :  $2 + 4$

**2 dan 4** adalah sebuah operand. Ekspresi ini akan menghasilkan **6**.

### 5.2.1 Operator Aritmatika

PHP menggunakan operator arithmetic dasar :

- + (Tambah)
- (Kurang)
- % (Modulus)
- \* (Perkalian)
- / (Pembagian)

PHP mengabaikan karakter spasi dalam sebuah operasi. Ekspresi  $\$x = 5 + 9$  dengan dengan  $\$x=5+9$  akan menghasilkan nilai yang sama. Penggunaan spasi disesuaikan dengan keinginan dari para user.

### 5.2.2 Unary Operator

Tanda minus ( - ) digunakan dengan sebuah nilai numerik tunggal untuk menegatifkan sebuah bilangan (untuk membuat negatif bilangan positif atau membuat positif bilangan negatif).

Contoh :

```
$x = 2;
$y = -$x;
$i = -234;
$j = -$i;
```

### 5.2.3 Operator Perbandingan

Operator perbandingan digunakan untuk menguji suatu kondisi. Ekspresi yang menggunakan operator perbandingan akan selalu menghasilkan nilai boolean, yaitu antara true (benar) atau false (salah).

Operator	Name	Description	Example
$x == y$	Equal	True if x is equal to y	$5==8$ returns false
$x === y$	Identical	True if x is equal to y, and they are of same type	$5===\text{"5"}$ returns false
$x != y$	Not equal	True if x is not equal to y	$5!=8$ returns true
$x <> y$	Not equal	True if x is not equal to y	$5<>8$ returns true

<b>x !== y</b>	Not identical	True if x is not equal to y, or they are not of same type	5!== "5" returns true
<b>x &gt; y</b>	Greater than	True if x is greater than y	5>8 returns false
<b>x &lt; y</b>	Less than	True if x is less than y	5<8 returns true
<b>x &gt;= y</b>	Greater than or equal to	True if x is greater than or equal to y	5>=8 returns false
<b>x &lt;= y</b>	Less than or equal to	True if x is less than or equal to y	5<=8 returns true

#### 5.2.4 String Concatenation Operator

Tanda titik ( . ) sebagai operator concatenate (penggabung) digunakan untuk menggabungkan dua atau lebih nilai string menjadi sebuah string tunggal.

Contoh :

```
$subjek = "saya";
$predikat = "sedang belajar";
$objek = "PHP";
$kalimat = $subjek." ".$predikat." ".$objek;
```

```
echo $kalimat;
echo "$kalimatversi 3";
echo " $kalimat versi 3";
echo "${kalimat} versi 3";
```

```
$bilangan = "<B>1</B>&nbsp;";
$bilangan .= "<I>2</I>&nbsp;";
$bilangan .= "<U>3</U>";
echo $bilangan;
```

#### 5.2.5 Variable Assignment Shortcut

Pada PHP dimungkinkan untuk melakukan penggunaan jalan pintas untuk operator pada pernyataan penugasan dimana operand pertama adalah sebuah variabel dan hasilnya disimpan pada variabel yang sama.

Assignment	Same as...	Description
<b>x = y</b>	x = y	The left operand gets set to the value of the expression on the right
<b>x += y</b>	x = x + y	Addition
<b>x -= y</b>	x = x - y	Subtraction
<b>x *= y</b>	x = x * y	Multiplication
<b>x /= y</b>	x = x / y	Division
<b>x %= y</b>	x = x % y	Modulus
<b>a . = b</b>	a = a . b	Concatenate two strings

### 5.2.6 Incrementing/Decrementing Operators

Operator	Name	Description
<code>++ x</code>	Pre-increment	Increments x by one, then returns x
<code>x ++</code>	Post-increment	Returns x, then increments x by one
<code>-- x</code>	Pre-decrement	Decrements x by one, then returns x
<code>x --</code>	Post-decrement	Returns x, then decrements x by one

## 5.3 Condition Statement

Di dalam PHP terdapat beberapa Condition Statement yaitu :

- **if statement** : mengeksekusi beberapa kode hanya jika kondisi yang ditentukan benar
- **if...else statement** : mengeksekusi beberapa kode jika kondisi benar dan kode lain jika kondisi salah
- **if...else if...else statement** : pernyataan lain - memilih salah satu dari beberapa blok kode yang akan dieksekusi
- **switch statement** : memilih salah satu dari banyak blok kode yang akan dieksekusi

### 5.3.1 If statement

Digunakan untuk mengeksekusi beberapa kode hanya jika kondisi tertentu adalah benar.

#### Sintax :

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

#### Contoh :

```
<?php
    $t=date("H");
    if ($t<"20")
    {
        echo "Have a good day!";
    }
?>
```

### 5.3.2 If...else Statement

Digunakan untuk mengeksekusi beberapa kode jika kondisi benar dan kode lain jika kondisi salah.

**Syntax :**

```

if (condition)
{
    code to be executed if condition is true;
}
else
{
    code to be executed if condition is false;
}

```

**Contoh :**

```

<?php
$t=date("H");
if ($t<"20")
{
    echo "Have a good day!";
}
else
{
    echo "Have a good night!";
}
?>

```

**5.3.3 If...else if...else Statement**

Digunakan untuk memilih salah satu dari beberapa blok kode yang akan dieksekusi.

**Syntax :**

```

if (condition)
{
    code to be executed if condition is true;
}
else if (condition)
{
    code to be executed if condition is true;
}
else
{
    code to be executed if condition is false;
}

```

**Contoh :**

```

<?php
$t=date("H");
if ($t<"10")
{
    echo "Have a good morning!";
}
else if ($t<"20")

```

```

{
  echo "Have a good day!";
}
else
{
  echo "Have a good night!";
}
?>

```

#### 5.3.4 Switch Statement

Pernyataan switch digunakan untuk melakukan tindakan yang berbeda berdasarkan kondisi yang berbeda. Pernyataan switch digunakan untuk memilih salah satu dari banyak blok kode yang akan dieksekusi.

##### Syntax :

```

switch (n)
{
  case label1:
    code to be executed if n=label1;
    break;
  case label2:
    code to be executed if n=label2;
    break;
  default:
    code to be executed if n is different from both label1 and label2;
}

```

##### Contoh :

```

<?php
$favcolor="red";
switch ($favcolor)
{
  case "red":
    echo "Your favorite color is red!";
    break;
  case "blue":
    echo "Your favorite color is blue!";
    break;
  case "green":
    echo "Your favorite color is green!";
    break;
  default:
    echo "Your favorite color is neither red, blue, or green!";
}

```

```
}
?>
```

## 5.4 Perulangan (Looping)

Digunakan untuk mengeksekusi blok code yang sama berulang-ulang, terdapat beberapa Looping di PHP :

- **while** - loop seluruh blok kode sementara kondisi yang ditentukan benar
- **do ... while** - loop seluruh blok kode satu kali, dan kemudian mengulangi loop selama kondisi yang ditentukan benar
- **for** - loop seluruh blok kode jumlah tertentu kali
- **foreach** - loop seluruh blok kode untuk setiap elemen dalam array

### 5.4.1 While Loop

**Syntax :**

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

**Contoh :**

```
<html>
<body>

<?php
$i=1;
while($i<=5)
{
    echo "The number is " . $i . "<br>";
    $i++;
}
?>

</body>
</html>
```

### 5.4.2 Do...while Statement

**Syntax :**

```
do
{
    code to be executed;
```

```

}
while (condition);

```

**Contoh :**

```

<html>
<body>

<?php
$i=1;
do
{
    $i++;
    echo "The number is " . $i . "<br>";
}
while ($i<=5);
?>

</body>
</html>

```

**5.4.3 For Loop****Syntax :**

```

for (init; condition; increment)
{
    code to be executed;
}

```

**Contoh :**

```

<html>
<body>

<?php
for ($i=1; $i<=5; $i++)
{
    echo "The number is " . $i . "<br>";
}
?>

</body>
</html>

```

**5.4.4 Foreach Loop****Syntax :**

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

**Contoh :**

```
<html>
<body>

<?php
$x=array("one","two","three");
foreach ($x as $value)
{
    echo $value . "<br>";
}
?>

</body>
</html>
```

**Contoh-2 :**

```
<?php
$myList = array(1,2,3,4,5,6,7,8,9);

foreach ($myList as $value)
{
    echo " $value ";
}
echo "<br>";

$myList = array('a'=>1,'b'=>2,'c'=>3,'d'=>4,'e'=>5, 'f'=>6,'g'=>7,'h'=>8,'i'=>9);

foreach ($myList as $key => $value)
{
    echo $key." ".$value."<br>";
}
?>
```

## 5.5 Array

Array adalah sebuah tempat untuk sekumpulan nilai. Sebuah array terdiri dari sejumlah element, yang masing-masing memiliki sebuah nilai - data yang tersimpan pada elemen array tersebut - dan sebuah key atau index, dimana elemen tersebut dapat dirujuk.

Normalnya, sebuah index berupa integer. Secara default, array adalah basis nol, artinya elemen pertama dari array memiliki index nol. akan tetapi index dapat juga berupa string.

Bentuk sederhana array terdiri dari serangkaian elemen yang bertanda dimulai dari nol dan bertambah secara sekuensial. Sebagai contoh sebuah array bernama \$branch, setiap elemen berisi nama kota cabang sebuah perusahaan.

\$branch[0]	\$branch[1]	\$branch[2]	\$branch[3]	\$branch[4]
"Semarang"	"Surabaya"	"Medan"	"Bandung"	"Yogyakarta"

Ada tiga jenis array di PHP:

- Numeric array – Array dengan dengan kunci ID numerik
- Associative array – Array dimana setiap kunci ID berasosiasi dengan sebuah nilai
- Multidimensional array - Array yang menyimpan satu atau lebih array

### 5.5.1 Inisialisasi array

Jika tanda kurung siku pada variabel array tidak diberikan nilai index, maka secara default maka element sebenarnya bernilai index 0,1,2,... contoh dibawah akan menghasilkan array yang sama dengan contoh diatas.

```
$branch[0] = "Semarang";
$branch[1] = "Surabaya";
$branch[2] = "Medan";
```

Ada banyak cara untuk melakukan inisialisasi sebuah array. Cara pertama yang sederhana adalah cukup dengan memberikan nilai kepada variabel array.

```
<?php
$branch[] = "Semarang";
$branch[] = "Surabaya";
$branch[] = "Medan";
echo $branch[0]."<Br>";
echo $branch[1]."<Br>";
echo $branch[2]."<Br>";
?>
```

Dalam prakteknya, pemberian index dilakukan secara sekuensial atau berurutan. Tetapi dilain hal dapat dilakukan peng-indekan secara acak sesuai keinginan user.

```
<?php
$branch[20] = "Semarang";
$branch[22] = "Surabaya";
$branch[23] = "Medan";
echo $branch[23]."<BR>";
echo $branch[21];
?>
```

Jika menginginkan jumlah dari elemen array yang terdapat pada sebuah variabel array, dapat digunakan fungsi `count()`. Fungsi tersebut mengembalikan nilai fungsi berupa integer yang menyatakan jumlah elemen array.

```
<?php
$branch[20] = "Semarang";
$branch[23] = "Surabaya";

echo count ($branch)."<br>";
echo $branch[24];
?>
```

## 5.6 Function

Function atau dalam bahasa Indonesia di sebut fungsi adalah sekumpulan basis program yang terdiri atas serangkaian perintah program di susun sedemikian rupa sehingga bisa menjadi satu modul saja.

Ketika programmer membuat program yang cukup rumit/kompleks, perlu digunakan fungsi karena fungsi akan sangat membantu menyederhanakan penulisan program.

Dalam segi pembuatan segi pembuatan, fungsi terbagi dua:

### ► Fungsi built-in yang di sediakan PHP

Pemrogram tinggal memanggil nama fungsinya saja dan siap digunakan. Seluruh isi modul fungsi sudah dibuatkan oleh PHP secara built-in.

### ► Fungsi buatan sendiri

Pemrogram juga diberikan kebebasan untuk membuat fungsi sendiri. Hal ini dilakukan jika fungsi yang dibutuhkan tidak disediakan oleh PHP secara built-in.

Pada pembuatan fungsi sendiri tata cara penamaan sebuah fungsi mengikuti aturan yang sama dengan tata cara penamaan variabel. Selama masih sesuai dengan aturan PHP, programmer diberi kebebasan memberikan nama fungsi. Nama fungsi yang benar harus diawali dengan huruf atau garis bawah (underscores), kemudian diikuti dengan angka huruf, angka, huruf atau garis bawah.

**Syntax :**

**Sruktur Fungsi:**

```
function nama_fungsi(argumen1[,argumen2])
{
    statement;
    [return value]
}
```

**Contoh :**

```

<html>
<body>

<?php
function writeName()
{
echo "Kai Jim Refsnes";
}

echo "My name is ";
writeName();
?>

</body>
</html>

```

**5.6.1 Parameter Function**

Ada dua jenis argumen yang bisa dilewatkan dalam fungsi:

**1. Passing by value**

Argumen passing by value merupakan perlakuan standar dalam fungsi PHP. Setelah keluar dari fungsi, maka nilai variabel akan selalu tetap.

**Contoh :**

```

<html>
<body>
<?php
function writeName($fname)
{
echo $fname . " Refsnes.<br>";
}
echo "My name is ";
writeName("Kai Jim");
echo "My sister's name is ";
writeName("Hege");
echo "My brother's name is ";
writeName("Stale");
?>

</body>
</html>

```

**Contoh-2 :**

```

<html>
<body>

<?php
function writeName($fname,$punctuation)
{
echo $fname . " Refsnes" . $punctuation . "<br>";
}

echo "My name is ";
writeName("Kai Jim",".");
echo "My sister's name is ";
writeName("Hege","!");
echo "My brother's name is ";
writeName("Ståle","?");
?>

</body>
</html>

```

**2. Passing by references**

Argumen akan ikut berubah setelah keluar dari fungsi. penggunaan passing by reference harus diawali dengan tanda ampersan (“&”)

**Contoh :**

```

<?
function tambah_string(&$string)
{
$string.=' kata tambahan';
}

$str='Kata awal, ';
tambah_string($str);
echo $str;
?>

```

**5.6.2 Return Value**

Fungsi dapat memberikan nilai balik dengan memberikan statement *return*.

**Contoh :**

```
<html>
<body>

<?php
function add($x,$y)
{
$total=$x+$y;
return $total;
}

echo "1 + 16 = " . add(1,16);
?>

</body>
</html>
```

**Contoh-2 :**

```
<?php
function luas_kotak($num)
{
    return $num * $num;
}
echo luas_kotak(2)."<br>";
//Jika hasil yang di return lebih dari satu
function angka_kecil()
{
    return array(0,1,2);
}
list($nol,$satu,$dua) = angka_kecil();
echo $nol.", ".$satu.", ".$dua;
?>
```

# Pertemuan 6

## Form dan Input User

### 6.1 Form

Form adalah sebuah tag HTML yang di fungsikan untuk melakukan pengiriman data dari client side menuju server side untuk di olah dan akan di kembalikan ke sisi client side jika di perlukan, tergantung dari fungsi website tersebut. Pengiriman data pada FORM dapat melalui dua buah cara yaitu **GET** dan **POST**.

**GET** adalah metode pengiriman data menggunakan query string, jadi seluruh nilai pada form anda akan di kirim ke sisi server/file dan nilai dari form anda akan tampil pada barus URL/ Address bar contoh,

<http://www.namadomain.com/file.php?nama=ian+lubis&alamat=denpasar>

**file.php** adalah file yang menerima dari nilai query string, dan disana terdapat dua buah variable “**nama**” dan “**alamat**”, setiap varibale **GET** selalu di akhiri dengan tanda =, hal ini bertujuan untuk mengisikan nilai dari variable tersebut.

Contoh penulisan untuk mengambil nilai dari query string adalah :

```
<?php echo $_GET[nama_variable]; ?>
```

**POST**, post sendiri mengirimkan data ke sisi lain/ file namun tidak menampilkan nilai dari form pada address bar seperti GET/ Query String.

Contoh penulisan untuk mengambil nilai dari form **POST** adalah:

```
<?php echo $_POST[nama_variable]; ?>
```

Perbedaannya :

- Method POST tidak menampilkan nilai variabel pada URL dan Method GET menampilkan nilai variabel yang dikirimkan.
- Method POST lebih aman dan Method GET kurang aman (pada contoh diatas berati password ditampilkan pada URL).
- Jika Method GET dibatasi panjang string hingga 2047 karakter, Method POST tidak.
- Perbedaan pengambilan data jika Method POST menggunakan \$\_POST sedangkan Method GET menggunakan \$\_GET.
- Method POST biasanya digunakan untuk input dari FORM, Method GET menggunakan input dari LINK atau akses menggunakan link.
- Method POST digunakan untuk mengirimkan data rahasia seperti password, Method GET digunakan untuk mengirimkan/mengambil data pablik seperti id\_user atau id\_halaman.

## 6.2 Penggunaan GET

Contoh Penggunaan metode **GET**, menggunakan elemen FORM (Form Input Text, Checkbox, Radio Button, Textarea, Dropdown) pada HTML.

### latihan1.php

```
<html>
  <head>
    <title>Latihan 1</title>
  </head>
  <body>
    <form method="GET" action="latihan2.php">
      Nama: <input type="text" name="nama"><br />
      Hoby: <input type="checkbox" name="hoby" value="renang"> Renang<br />
      Status: <input type="radio" name="status" value="single"> Single
             <input type="radio" name="status" value="menikah"> Menikah<br />
      Jeniskelamin: <select name="jeniskelamin">
                    <option value="pria">pria</option>
                    <option value="wanita">wanita</option>
                  </select>
      <input type="submit" name="send" value="kirim">
    </form>
  </body>
</head>
```

### latihan2.php

```
<?php
  echo $_GET["nama"];
  echo "<br />";
  echo $_GET ["hoby"];
  echo "<br />";
  echo $_GET ["status"];
  echo "<br />";
  echo $_GET ["jeniskelamin"];
  echo "<br />";
?>
```

### 6.3 Penggunaan POST

Contoh Penggunaan metode POST, menggunakan elemen FORM (Form Input Text, Checkbox, Radio Button, Textarea, Dropdown) pada HTML.

#### latihan3.php

```
<html>
  <head>
    <title>Latihan 1</title>
  </head>
  <body>
    <form method="POST" action="latihan4.php">
      Nama: <input type="text" name="nama"><br />
      Hoby: <input type="checkbox" name="hoby" value="renang"> Renang<br />
      Status: <input type="radio" name="status" value="single"> Single
             <input type="radio" name="status" value="menikah"> Menikah<br />
      Jeniskelamin: <select name="jeniskelamin">
                    <option value="pria">pria</option>
                    <option value="wanita">wanita</option>
                  </select>
      <input type="submit" name="send" value="kirim">
    </form>
  </body>
</head>
```

#### latihan4.php

```
<?php
  echo $_POST["nama"];
  echo "<br />";
  echo $_POST["hoby"];
  echo "<br />";
  echo $_POST["status"];
  echo "<br />";
  echo $_POST["jeniskelamin"];
  echo "<br />";
?>
```

## 6.4 Isset dan Empty

Dalam pembuatan web, kadang kala kita hanya membutuhkan proses di satu halaman saja, menerima inputan dan menampilkan output di halaman yang sama. Misalkan seperti berikut :

### Latihan5.php

```
<?php
    echo $_POST["nama"];
?>
<html>
    <head>
        <title>Latihan 1</title>
    </head>
    <body>
        <form method="POST">
            Nama: <input type="text" name="nama"><br />
            <input type="submit" name="send" value=" kirim">
        </form>
    </body>
</html>
```

Jika halaman di atas dijalankan maka akan terjadi error, karena `$_POST["nama"]` belum didefinisikan, untuk mengatasi hal tersebut maka kita gunakan validasi dengan menggunakan fungsi **ISSET** atau **EMPTY**.

**isset** digunakan untuk menentukan apakah suatu variabel telah didefinisikan dan variabel tersebut isinya bukan NULL.

```
<?php
$var = 0;

if (isset($var)) {
    echo $var;
}
else {
    echo 'false, variable is not set!';
}
?>
```

Hasil: 0 (true, variable is set)

**empty** juga dapat digunakan untuk keperluan yang sama dengan **isset**, namun perbedaannya **empty** sekaligus mengecek jika nilai variabel tersebut kosong maka hasilnya **true**.

```
<?php
$var = 0;

if (empty($var)) {
    echo 'true, variable is empty!';
}
else {
    echo $var;
}
?>
```

Hasil: true, variable is empty!

`$var = 0` bagi **isset** tetap **true** karena sekali lagi saya katakan, **isset** hanya mengecek apakah suatu variabel sudah didefinisikan atau belum. **isset** hanya bernilai **false** jika variabel belum didefinisikan atau jika variabel **null** (`$var = null`).

Berikut ini adalah nilai-nilai yang dianggap **empty**:

1. `// $var = "`; (ilustrasi variabel yang belum didefinisikan atau tidak ada)
2. `$var = "`;
3. `$var = 0`;
4. `$var = false`;
5. `$var = null`;

Semuanya **isset** kecuali yang belum didefinisikan dan **null**.

Sehingga halaman di atas menjadi :

```
<?php
if(isset($_POST["nama"]))
    echo $_POST["nama"];
?>
```

Atau :

```
<?php
if(empty($_POST["nama"]))
    echo $_POST["nama"];
?>
```

## 6.5 Halaman Login

Berikut adalah contoh penerapan penggunaan Form dan Input User pada Halaman Login. Buatlah Halaman index.php seperti berikut :

### Index.php

```
<html>
<head>
<title>Halama Login</title>
</head>
<body>
<form id="form1" name="form1" method="post" action="dologin.php">
  <table width="200" border="1">
    <tr>
      <td>username</td>
      <td><label>
        <input type="text" name="user" id="user" />
      </label></td>
    </tr>
    <tr>
      <td>password</td>
      <td><label>
        <input type="text" name="pass" id="pass" />
      </label></td>
    </tr>
    <tr>
      <td>
        <?
        if(isset($_GET["err"]))
        {
            $err=$_GET["err"];
            echo $err;
        }
        ?>
      </td>
      <td><label>
        <input type="submit" name="submit" id="submit" value="Submit" />
      </label></td>
    </tr>
  </table>
</form>
</body>
</html>
```

**DoLogin.php**

```
<?php
    $user= $_POST[user];
    $pass= $_POST[pass];

    if($user == "admin" && $pass == "admin")
    {
        header("Location:home.php");
    }
    else
    {
        header("Location:index.php?err=invalidlogin");
    }
?>
```

**Home.php**

```
<html>
<head>
<title>Halaman Home</title>
</head>
<body>
    selamat datang di halama Home
</body>
</html>
```

**6.6 Latihan**

1. Silahkan membuat form menggunakan teknik GET dengan isian form nama\_anda, pekerjaan, tanggal\_lahir, bulan\_lahir, tahun\_lahir, alamat, status, cita\_cita.
2. Buat sistem manajemen restoran berbasis text dengan ketentuan berikut :
  - Harga semua masakan adalah Rp. 15.000
  - Sistem akan melakukan pengulangan tampilan Pilih Menu, sampai dipilih menu 5 (Hitung Total) atau menu 6 (Keluar Aplikasi)
  - Sistem akan melakukan pengulangan tampilan Menu Restoran, sampai dipilih menu 6 (Keluar Aplikasi)

Menu Restoran:

1. Nasi Goreng
2. Nasi Rawon
3. Nasi Gudæg
4. Jus Jeruk
5. Hitung Total
6. Keluar Aplikasi

Pilih Menu? 1

Pilih Menu? 4

(mengulang ke tampilan Pilih Menu, sampai dipilih menu 5 (Hitung Total) atau menu 6 (Keluar Aplikasi))

Pilih Menu? 5

Pilihan menu adalah Nasi Goreng dan Jus Jeruk.

Total yang harus dibayar adalah Rp. 30.000

(mengulang ke tampilan Menu Restoran, sampai dipilih menu 6 (Keluar Aplikasi))

NB : Gunakan Fungsi Perulangan dan Fungsi Kondisi.

## Pertemuan 7

### Session dan Include

#### 7.1 Session

Session Adalah sebuah variabel sementara yang diletakkan di server. Di mana PHP bisa mengambil nilai yang tersimpan di server walaupun kita membuka halaman baru. Penyimpanan isi variabel session berada di server, jadi tidak bisa diakses secara langsung oleh client. Biasanya session akan hilang jika anda menutup browser.

Dalam aplikasi berbasis web, session banyak digunakan sebagai autentifikasi login. Dengan session memungkinkan programmer mengatur siapa saja yang bisa mengakses suatu halaman.



**Gambar 7.1 Proses Session**

- **SESSION\_START()**

Deklarasi awal kalau kita menggunakan session, yaitu menggunakan fungsi `session_start()`

Set suatu nilai ke variabel `$_SESSION`

#### **Contoh pembuatan Session :**

```

<?php
    session_start(); //perintah ini harus ada pada setiap halaman yang
                    berhubungan dengan session
    $_SESSION['sesLogin']=$user;
?>
  
```

- **SESSION\_DESTROY()**

**session destroy()** berguna untuk menghapus dan mengakhiri session, sekaligus menghapus semua elemen yang ada. Jika browser telah ditutup, secara otomatis session akan diakhiri walaupun fungsi `session_destroy()` tidak dituliskan secara eksplisit.

Contoh :

**useSession01.php**

```
<?php
    session_start();

    if(!isset($_SESSION['count']))
    {
        $_SESSION['count']=0;
    }
    else
    {
        $_SESSION['count']++;
    }
    $count=$_SESSION['count'];
    echo "<h1> Demo Session </h1>";
    echo "anda telah mengunjungi halaman ini sebanyak : $count kali";
?>
```

Contoh Menggunakan `session_id()` :

**useSession02.php**

```
<?php
    session_start();
    $idSession = session_id();

    if(!isset($_SESSION['count']))
    {
        $_SESSION['count']=0;
    }
    else
    {
        $_SESSION['count']++;
    }
    $count=$_SESSION['count'];
    echo "<h1> Demo Session </h1>";
    echo "anda telah mengunjungi halaman ini sebanyak : $count kali <br>";
    echo "$idSession";
?>
```

Contoh Menggunakan Sesion\_Destroy() :

**useSession03.php**

```
<?php
    session_start();
    $idSession = session_id();
    session_destroy();

    if(!isset($_SESSION['count']))
    {
        $_SESSION['count']=0;
    }
    else
    {
        $_SESSION['count']++;
    }
    $count=$_SESSION['count'];
    echo "<h1> Demo Session </h1>";
    echo "anda telah mengunjungi halaman ini sebanyak : $count kali <br>";
    echo "$idSession";
?>
```

## 7.2 Penggunaan Session Pada Login

### Login.php

```
<html>
<head>
<title>Halama Login</title>
</head>
<body>
<form id="form1" name="form1" method="post" action="dologin.php">
  <table width="200" border="1">
    <tr>
      <td>username</td>
      <td><label>
        <input type="text" name="user" id="user" />
      </label></td>
    </tr>
    <tr>
      <td>password</td>
      <td><label>
        <input type="password" name="pass" id="pass" />
      </label></td>
    </tr>
    <tr>
      <td>
        <?
        if(isset($_GET["err"]))
          {
            $err=$_GET["err"];
            echo $err;
          }
        ?>
      </td>
      <td><label>
        <input type="submit" name="submit" id="submit" value="Submit" />
      </label></td>
    </tr>
  </table>
</form>
</body>
</html>
```

**DoLogin.php**

```

<?php
    session_start();
    include "koneksi.php";
    $user= $_POST[user];
    $pass= $_POST[pass];

    if($user == "" && $pass == "")
    {
        header("Location:login.php?err=data masih kosong");
    }
    else
    {
        $query = mysql_query("select * from user where username =

```

**Home.php**

```

<?php
    session_start();
    if(!isset($_SESSION['username'])){
        header("Location:login.php?err=Login Dahulu");
    }
    echo $_SESSION['username'];
?>
<html>
<head>
<title>Halaman Home</title>
</head>
<body>
<?php
    echo "selamat datang, $_SESSION[username] <br>";
    echo "<a href='Logout.php'>Logout</a>";
?>
</body>
</html>

```

**Logout.php**

```

<?php
    session_start();
    session_destroy();
    header("Location:login.php?err=anda sudah Logout");
?>

```

### 7.3 Cookies

Cookies merupakan fasilitas seperti halnya session, cookies merupakan teknik pertama kali yang dikenalkan oleh Netscape untuk membuat adanya suatu state koneksi antara klien dan server web (session). Hanya saja mekanisme pencatatan dilakukan pada setiap klien (dalam hal ini browser web). Cookies merupakan data yang disimpan dalam browser web, yang menjadi identitas yang selalu akan dikirimkan kepada server setiap kali terjadi kontak antara klien dan server, apabila diminta. Data ini pertama kali dikirimkan oleh server, kemudian disimpan dalam browser web.

Sintax lengkap cookies :

Setcookies("namacookie","data-nilai",masaberlaku,"path","domain",secure)

Namacookie	Nama cookie yang diinginkan
Data-nilai	Data untuk cookie
Masaberlaku	Waktu berlakunya cookie dalam hitungan detik, dari jam sekarang ditambah dengan beberapa detik. Data berupa integer tanggal
Path	Nama path yang valid untuk berlakunya suatu cookie. Jika script berbeda path dengan script yang melakukan pendefinisian cookie, maka bisa jadi cookie tidak jadi berlaku
Domain	Nama domain yang digunakan untuk membuat proses kirim mengirim cookies dilakukan apabila domain yang benar (membatasi domain)
Secure	Digunakan agar proses pengiriman cookies dilakukan pada mode koneksi yang secure (aman). Data berupa integer.

**Contoh Cookies :**

#### Cookies1.php

```
<?php
    $value="tes";
    setcookie("mycookies",$value,time()+5);
    $x=$_COOKIE['mycookies'];
?>
<html>
<head>
    <title>Demo Cookies</title>
</head>
<body>
    <h1>Demo Cookies</h1>
    <?php
        echo "Cookies anda : $x";
    ?>
</body>
</html>
```

**Contoh menghapus Cookies :****Cookies2.php**

```
<?php
    setcookie("mycookies",,time()-5);
?>
```

Selain menggunakan script PHP seperti di atas, untuk menghilangkan cookie juga dapat langsung dari browser yang sedang digunakan. Jika menggunakan firefox pilih menu **Tool > Clear Private Data** atau dengan menggunakan **Ctrl+Shift+Del**.

**7.4 SESSION VS COOKIE**

Penyimpanan informasi dengan sistem cookies ini paling sering digunakan untuk :

- Menyimpan username dan password login agar pengguna tidak selalu harus mengisikannya pada saat membuka halaman. Ini biasanya dengan memberitahunya dengan checkbox "Remember me" atau "Ingat saya selalu" yang diletakkan pada layar tampilan login.
- Untuk mencatat konfigurasi yang dilakukan oleh pengguna, seperti warna tema, jenis huruf, pilihan bahasa dll
- Untuk mengetahui apakah pengunjung pernah datang atau belum ke halaman yang sedang dibuka (seperti digunakan dalam hit counter/penghitung pengunjung)

Session ini sering digunakan untuk keperluan :

- Menyimpan informasi login yang berlaku hanya dalam satu sesi
- Menyimpan catatan order barang dalam sistem e-commerce / transaksi online

	menyimpan informasi	tempat menyimpan	bisa dipanggil	bersifat global	dipanggil lain waktu	dibatasi browser	dapat diedit user
cookies	Ya	Klien	Ya	Ya	Ya	Ya	Ya
session	Ya	Server	Ya	Ya	Tidak	Tidak	Tidak

**7.5 Include(), Include\_once(), require(), require\_once()**

**PHP** memiliki kemampuan untuk **menyertakan** file php lainnya ke dalam suatu program PHP yaitu dengan menggunakan fungsi include, include\_once, require dan require\_once. masing-masing mempunyai fungsi yang sama, yaitu memasukkan fungsi atau content dari satu file php ke dalam file php lain. Berikut contoh penggunaan dari masing-masing fungsi tersebut.

## 1. include()

Fungsi **include()** akan menyertakan dan mengevaluasi seluruh program yang ada di file yang disertakan. Jika terdapat error pada program yang disertakan, maka error akan ditampilkan di layar. Dan jika file yang disertakan ternyata tidak ditemukan (mungkin karena lokasi yang salah atau memang file tidak ada), maka program selanjutnya (setelah include) akan tetap dijalankan walaupun ditampilkan error.

### Contoh include :

#### var.php

```
<?php
    $var = "nilai var ke-1";
?>
```

#### include.php

```
<?php
    include("var.php");
    echo $var;//output:nilai var ke-1

    //ubah variabel $var
    $var = "nilai var ke-2";
    echo $var;//output:nilai var ke-2

    //sisipkan lagi var.php
    include("var.php");
    //output variabel $var kembali pada nilai semula
    echo $var;//output:nilai var ke-1
?>
```

## 2. include\_once()

Fungsi **include\_once()** mempunyai definisi yang sama dengan **include()**. Perbedaannya fungsi ini akan memastikan bahwa file yang disisipkan hanya akan disisipkan sekali saja. Nilai variabel mengikuti nilai variabel terbaru.

### Contoh include\_once:

#### var.php

```
<?php
    $var = "nilai var ke-1";
?>
```

**include\_once.php**

```

<?php
    include("var.php");
    echo $var;//output:nilai var ke-1

    //ubah variabel $var
    $var = "nilai var ke-2";
    echo $var;//output:nilai var ke-2

    //sisipkan lagi var.php dengan include_once
    include_once("var.php");
    //output variabel $var mengikuti nilai variabel yang baru
    echo $var;//output:nilai var ke-1
?>

```

**3. require()**

Fungsi `require()` akan menyisipkan dan mengevaluasi file yang sudah dispesifikasikan sebelumnya dalam parameter fungsi. `require ()` menghasilkan pesan kesalahan fatal, dan script akan berhenti.

**require.php**

```

<?php
    require("var.php");
    echo $var;//output:nilai var ke-1

    //ubah variabel $var
    $var = "nilai var ke-2";
    echo $var;//output:nilai var ke-2

    //sisipkan lagi var.php
    require("var.php");
    //output variabel $var kembali pada nilai semula
    echo $var;//output:nilai var ke-1
?>

```

**4. require\_once()**

Fungsi `require_once()` mempunyai definisi yang sama dengan `require()`. Perbedaannya fungsi ini akan memastikan bahwa file yang disisipkan hanya akan disisipkan sekali saja. Nilai variabel mengikuti nilai variabel terbaru.

**Require\_once.php**

```
<?php
    require("var.php");
    echo $var;//output:nilai var ke-1

    //ubah variabel $var
    $var = "nilai var ke-2";
    echo $var;//output:nilai var ke-2

    //sisipkan lagi var.php
    require_once("var.php");
    // output variabel $var mengikuti nilai variabel yang baru
    echo $var;//output:nilai var ke-1
?>
```

## Pertemuan 8

### PHP dan MySQL

#### 8.1 MySQL

#### 8.2 Struktur dan Syntax MySQL

#### 8.3 Membuat Database, membuat table

Buatlah database dengan nama : **mahasiswa** dan satu buah tabel dengan nama : **tbl\_mahasiswa** mengikuti spesifikasi berikut:

Field	Tipe data	Size
id_mhs	int	5
nama_mhs	varchar	50
jenis_kelamin	varchar	10
tgl_lahir	date	-
alamat	varchar	200

Atau dengan menggunakan Query :

```
create table tbl_mhsiswa (
  id_mhs int(5) primary key auto_increment,
  nama_mhs varchar(50),
  jenis_kelamin varchar(10),
  tgl_lahir date,
  alamat varchar(200)
);
```

Kemudian isi dengan 5 buah data.

## 8.4 Koneksi PHP dan MySQL

Untuk melakukan koneksi antara PHP dan MySQL yang perlu dilakukan adalah memuat fungsi untuk menangani koneksi antara PHP dan MySQL. Buatlah file koneksi seperti di bawah ini :

### koneksi.php

```
<?php
    $host = "localhost";
    $username = "root";
    $password = "";
    $database = "mahasiswa";

    mysql_connect($host, $username, $password) or die ("Koneksi Gagal");
    mysql_select_db($database) or die ("Database tidak ditemukan");

?>
```

## 8.5 Database Query

Setelah berhasil melakukan koneksi, selanjutnya kita akan coba untuk menampilkan isi database menggunakan php. Di sini kita akan coba bagaimana menampilkan isi database menggunakan pengulangan jenis while.

Untuk dapat menampilkan isi database, pertama adalah lakukan dulu koneksi, baru kemudian jalankan querynya untuk menampilkan isi databasenya.

### selectquery1.php

```
<?php
    include "koneksi.php";

    $tampilkan = "select * from tbl_mahasiswa";
    $query_tampilkan = mysql_query($tampilkan);

    while($hasil = mysql_fetch_array($query_tampilkan))
    {
        echo $hasil['nama_mhs']."<br />";
        echo $hasil['jenis_kelamin']."<br />";
    }

?>
```

kita juga dapat melakukan query insert dengan menggunakan php, buat sebuah file dengan nama **insertquery.php**

### **insertquery.php**

```
<?php
    include "koneksi.php";
    $insert = "insert into tbl_mahasiswa(nama_mhs, jenis_kelamin, tgl_lahir,
    alamat)
    values('Deny Sarwono', 'Pria', '1986-12-09','Jawa Barat');";
    $insert_query = mysql_query($insert);
    if($insert_query) {
        echo "Berhasil di insert";
    }
    else {
        echo "Gagal Insert Record";
    }
?>
```

selain itu kita juga bisa melakukan query delete dengan menggunakan php, buat sebuah file dengan nama **deletequery.php**

### **deletequery.php**

```
<?php
    include "koneksi.php";
    $delete = "delete from tbl_mahasiswa where id_mhs='0'";
    $delete_query = mysql_query($delete);
    if($delete_query)
        echo "Record Telah berhasil di hapuss...";
    else
        echo "Record Gagal untuk dihapus..";
?>
```

## **8.6 Menampilkan data menggunakan Table**

Agar tampilan data terlihat lebih rapih, maka kita gunakan table untuk menampilkan datanya. Proses penampilan data sebenarnya sama saja dengan sebelumnya, hanya ditambahkan tag Table untuk tampilan data.

**selectquerytabel.php**

```

<html>
<head><title>Tampil Data dengan Table</title></head>
<body>
<table border='1' cellpadding='2' cellspacing='2' align='center'
width='70%'>
<tr>
    <th>No</th>
    <th>Nama</th>
    <th>Jenis Kelamin</th>
    <th>Tanggal Lahir</th>
    <th>Alamat</th>
</tr>
<?php
include "koneksi.php";
$stampilkan = "select * from tbl_mahasiswa";
$query_tampilkan = mysql_query($stampilkan);

while($sisi = mysql_fetch_array($query_tampilkan))
{
    $no = $sisi['id_mhs'];
    $nama = $sisi['nama_mhs'];
    $jeniskelamin = $sisi['jenis_kelamin'];
    $stanggallahir = $sisi['tgl_lahir'];
    $alamat = $sisi['alamat'];
?>
    <tr align='center'>
        <td><?php echo $no?></td>
        <td><?php echo $nama ?></td>
        <td><?php echo $jeniskelamin ?></td>
        <td><?php echo $stanggallahir ?></td>
        <td><?php echo $alamat ?></td>
    </tr>
<?php
    }
?>
</table>
</body>
</html>

```

## 8.7 Input ke database

### forminsert.php

```

<html>
<head><title>Form Insert</title></head>
<body>
<form action="proces_insert.php" method="POST">
<table border='1' cellpadding='3' cellspacing='3' align='center'>
<caption><strong>Formulir Biodata
Mahasiswa</strong><br/><br/></caption><br/>
<tr><td>Nama</td><td><input type="text" name="nama" /></td></tr>
<tr><td>Jenis Kelamin</td><td><label for="jeniskelamin"><input
type="radio" name="jeniskelamin" value="pria"/>Pria</label>
<label for="jeniskelamin"><input type="radio" name="jeniskelamin"
value="wanita"/>Wanita</label>
</td></tr>
<tr><td>Tanggal lahir</td><td>
<select name="tanggal">
<option value="01">1</option>
<option value="01">1</option>
<option value="02">2</option>
<option value="03">3</option>
</select>

<select name="bulan">
<option value="01">Januari</option>
<option value="02">Februari</option>
<option value="03">Maret</option>
</select>

<select name="tahun">
<option value="1990">1990</option>
<option value="1991">1991</option>
</select>
</td></tr>
<tr><td>Alamat</td><td><textarea cols="20" rows="5"
name="alamat"></textarea></td></tr>
<tr><td></td><td><input type="submit" name=" kirim" value=" kirim"
/></td></tr>
</table>
</form>
</body>
</html>

```

**proces\_insert.php**

```

<?php
include "koneksi.php";

$nama = $_POST['nama'];
$jeniskelamin = $_POST['jeniskelamin'];

$tanggallahir = $_POST['tahun'].'-'. $_POST['bulan'].'-'. $_POST['tanggal'];
$alamat = $_POST['alamat'];

$insert = "insert into `tbl_mahasiswa` (`nama_mhs`, `jenis_kelamin`, `tgl_lahir`,
`alamat`)values ('$nama', '$jeniskelamin', '$tanggallahir', '$alamat')";

$insert_query = mysql_query($insert);

if($insert_query) {
    echo "Insert Record Berhasil<br />";
    echo "Anda Telah berhasil Menginput data:";
    echo $nama,"<br />",$jeniskelamin,"<br />",$tanggallahir,"<br />",$alamat ;
}
else
    echo "Gagal Insert Record";
?>

```

**8.8 Menampilkan Detail Data****displaydata.php**

```

<?php

include "koneksi.php";

$stampil_nama = "select id_mhs, nama_mhs from tbl_mahasiswa";

$stampil_nama_query = mysql_query($stampil_nama);

while ($hasil = mysql_fetch_array($stampil_nama_query))
{
    echo "<a href=detaildata.php?id=", $hasil['id_mhs'], ">",
    $hasil['nama_mhs'], "</a><br />";
}
?>

```

**detaildata.php**

```
<?php
include "koneksi.php";

$Sid = $_REQUEST['id'];

$detail = "select * from tbl_mahasiswa where id_mhs='$Sid'";

$detail_query = mysql_query($detail);
while ($hasil = mysql_fetch_array($detail_query))
{

$nama = $hasil['nama_mhs'] ;
$jenis_kelamin = $hasil['jenis_kelamin'];
$tanggal_lahir = $hasil['tgl_lahir'];
$alamat = $hasil['alamat'];

$data_lengkap = "Nama : ".$nama."<br />";
$data_lengkap .= "Jenis Kelamin : ".$jenis_kelamin."<br />";

$data_lengkap .= "Tanggal Lahir : ".$tanggal_lahir."<br />";
$data_lengkap .= "Alamat : ".$alamat."<br />";
}

echo "Informasi Detil mengenai <strong>$nama</strong> adalah :
<br/>".$data_lengkap;
?>
```

## Pertemuan 9

### PHP dan MySQL 2

#### 9.1 Delete Data

Untuk menghapus data, kita gunakan halaman tampil data, kemudian tambahkan beberapa source code berikut :

##### selectquerytabel.php

```

<html>
<head><title>Tampil Data dengan Table</title></head>
<body>
<table border='1' cellpadding='2' cellspacing='2' align='center'
width='70%'>
<tr>
    <th>No</th>
    <th>Nama</th>
    <th>Jenis Kelamin</th>
    <th>Tanggal Lahir</th>
    <th>Alamat</th>
    <th>Action</th>
</tr>
<?php
include "koneksi.php";
$stampilkan = "select * from tbl_mahasiswa";
$query_tampilkan = mysql_query($stampilkan);

while($sisi = mysql_fetch_array($query_tampilkan))
{
    $no = $sisi['id_mhs'];
    $nama = $sisi['nama_mhs'];
    $jeniskelamin = $sisi['jenis_kelamin'];
    $stanggalahir = $sisi['tgl_lahir'];
    $alamat = $sisi['alamat'];
?>
    <tr align='center'>
        <td><?php echo $no?></td>
        <td><?php echo $nama ?></td>
        <td><?php echo $jeniskelamin ?></td>
        <td><?php echo $stanggalahir ?></td>
        <td><?php echo $alamat ?></td>
        <td><?php echo "<a
href=deletedata.php?id=".$no.">". "Delete"."</a>";?></td>
    </tr>
<?php
}
?>
</table>
</body>
</html>

```

**deletedata.php**

```

<?php
include "koneksi.php";
$id = $_REQUEST['id'];

$delete = "delete from tbl_mahasiswa where id_mhs='$id'";
$delete_query = mysql_query($delete);

header("Location:selectquerytabel.php");

?>

```

**9.2 Update Data**

Untuk menghapus data, kita gunakan halaman tampil data, kemudian tambahkan beberapa source code berikut :

**selectquerytabel.php**

```

<td><?php echo "<a href=deletedata.php?id=".$no.">". "Delete"."</a>". "
"."<a href=deletedata.php?id=".$no.">". "Update"."</a>";?></td>

```

**Updatedata.php**

```

<?php
include "koneksi.php";
$id = $_REQUEST['id'];
$stampilkan = "select * from tbl_mahasiswa where id_mhs='$id'";
$query_tampilkan = mysql_query($stampilkan);

if($sisi = mysql_fetch_array($query_tampilkan))
{
    $no = $sisi['id_mhs'];
    $nama = $sisi['nama_mhs'];
    $jeniskelamin = $sisi['jenis_kelamin'];
    $tanggallahir = $sisi['tgl_lahir'];
    $alamat = $sisi['alamat'];
}

?>

```

```

<html>
<head><title>Form Insert</title></head>
<body>
<form action="proces_update.php" method="POST">
<table border='1' cellpadding='3' cellspacing='3' align='center'>
<caption><strong>Formulir Biodata
Mahasiswa</strong><br/><br/></caption><br/>
<input type="hidden" name="id" size="30" value="<?php echo $no ; ?>"
maxlength="50"/>
<tr>
<td>Nama</td>
<td><input type="text" name="nama" value="<?php echo $nama ?>" /></td>
</tr>
<tr>
<td>Jenis Kelamin</td>
<td><label for="jeniskelamin">
<input type="radio" name="jeniskelamin"
value="pria"/>Pria</label>
<label for="jeniskelamin">
<input type="radio" name="jeniskelamin"
value="wanita"/>Wanita</label>
</td>
</tr>
<tr>
<td>Tanggal lahir</td>
<td>
<select name="tanggal">
<option value="01">1</option>
<option value="01">1</option>
<option value="02">2</option>
<option value="03">3</option>
</select>

<select name="bulan">
<option value="01">Januari</option>
<option value="02">Februari</option>
<option value="03">Maret</option>
</select>
<select name="tahun">
<option value="1990">1990</option>
<option value="1991">1991</option>
</select>
</td>
</tr>

```

```

<tr>
  <td>Alamat</td>
  <td><textarea cols="20" rows="5" name="alamat"><?php echo $alamat
?></textarea>
  </td>
</tr>
<tr><td></td><td><input type="submit" name=" kirim" value=" kirim"
/></td>
</tr>
  <?php
      }
  ?>
</table>
</form>
</body>
</html>

```

### 9.3 Login dengan Database

Untuk membuat Login dengan database, kita gunakan halaman login yang sebelumnya kita sudah buat. Kemudian buat tabel di database mahasiswa dengan nama **user**.

Field	Tipe data	Size
username	int	5
password	varchar	50

**Index.php**

```
<html>
<head>
<title>Halama Login</title>
</head>
<body>
<form id="form1" name="form1" method="post" action="dologin.php">
  <table width="200" border="1">
    <tr>
      <td>username</td>
      <td><label>
        <input type="text" name="user" id="user" />
      </label></td>
    </tr>
    <tr>
      <td>password</td>
      <td><label>
        <input type="password" name="pass" id="pass" />
      </label></td>
    </tr>
    <tr>
      <td>
        <?
        if(isset($_GET["err"]))
        {
            $err=$_GET["err"];
            echo $err;
        }
        ?>
      </td>
      <td><label>
        <input type="submit" name="submit" id="submit" value="Submit" />
      </label></td>
    </tr>
  </table>
</form>
</body>
</html>
```

**DoLogin.php**

```
<?php
    session_start();
    include "koneksi.php";
    $user= $_POST[user];
    $pass= $_POST[pass];

    if($user == "" && $pass == "")
    {
        header("Location:login.php?err=data masih kosong");
    }
    else
    {
        $query = mysql_query("select * from user where username =
'$user'");

        if($hasil = mysql_fetch_array($query))
        {
            if(md5($pass) == $hasil['password'])
            {
                $_SESSION['username']= $user;
                header("Location:home.php");
            }
            else
                header("Location:login.php?err=invalidlogin");
        }
        else{
            header("Location:login.php?err=invalidlogin");
        }
    }
?>
```

**Home.php**

```
<?php
    session_start();
    if(!isset($_SESSION['username'])){
        header("Location:login.php?err=Login Dahulu");
    }
    echo $_SESSION['username'];
?>
<html>
<head>
<title>Halaman Home</title>
</head>
<body>
<?php
    echo "selamat datang, $_SESSION[username] <br>";
    echo "<a href='Logout.php'>Logout</a>";
?>
</body>
</html>
```

# Pertemuan 10

## 10.1 Pengertian JavaScript

JavaScript adalah bahasa pemrograman berbasis prototipe yang berjalan di sisi klien. Jika kita berbicara dalam konteks web, sederhananya kita dapat memahami JavaScript sebagai bahasa pemrograman yang berjalan khusus untuk di browser atau halaman web agar

halaman web menjadi lebih hidup. Jika dilihat dari suku katanya, JavaScript terdiri dari dua

kata, yaitu Java dan Script. Java adalah bahasa pemrograman yang berorientasi objek, sedangkan Script adalah serangkaian instruksi program.

JavaScript adalah bahasa script (bahasa pemrograman yang dapat memegang kontrol aplikasi) yang berbasis pada bahasa pemrograman Java. Namun JavaScript bukanlah bagian

dari teknologi Java dari Sun. Java merupakan bahasa pemrograman berorientasi objek murni,

sedangkan JavaScript digunakan secara prosedural. Cara kerja JavaScript adalah mengakses

elemen pada HTML dan membuat aksi jika elemen-elemen HTML itu mengalami perubahan,

misalnya berubahnya warna halaman web begitu sebuah tombol diklik. JavaScript membuat

sebuah halaman web menjadi lebih dinamis. JavaScript yang digunakan pada halaman web

merupakan client side scripting yang berarti bahwa web browser mengidentifikasi dan menjalankan skrip program yang disisipkan dalam dokumen web (yang diterima dari server),

dan mungkin memperbarui tampilan halaman di komputer pemakai(user) tanpa mengirimkan

permintaan(request) baru kepada web server. Karena JavaScript pada halaman web adalah

client side scripting, maka kode JavaScript yang ditulis satu file dengan dokumen HTML akan ditampilkan ketika sebuah halaman web dilihat source code-nya.

Secara fungsional, JavaScript digunakan untuk menyediakan akses skrip pada objek yang

dibenamkan(embedded). Contoh sederhana dari penggunaan JavaScript adalah membuka

halaman pop up, fungsi validasi pada form sebelum data dikirimkan ke server, merubah gambar kursor ketika melewati objek tertentu, dan lain-lain.

Yang harus diperhatikan dalam pengelolaan pemrograman JavaScript diantaranya JavaScript adalah bahasa pemrograman yang case sensitive, yang artinya JavaScript membedakan huruf kecil dan huruf besar. Hal ini sama seperti bahasa pemrograman Turbo

C atau C++ dimana huruf, "A" tidak sama dengan huruf "a".

JavaScript bekerja pada sisi browser, artinya untuk menampilkan halaman web, user menuliskan alamat web di address bar url. Setelah itu, browser mengambil file HTML (dengan file JavaScript yang melekat padanya jika memang ada) ke server yang beralamat di

URL yang diketikkan oleh user. Selesai file diambil, file ditambihkan pada browser. Nah, setelah file JavaScript berada pada browser, barulah skrip JavaScript tersebut bekerja. Efek dari JavaScript yang bekerja pada sisi browser ini, JavaScript dapat merespon perintah

user dengan cepat, dan membuat halaman web menjadi lebih responsif. JavaScript melakukan apa yang tidak bisa dilakukan oleh HTML, PHP, dan CSS dalam menangani hal-hal yang membutuhkan respon cepat terhadap aksi dari user. Misalnya, fungsi validasi pada form. Ketika user mengisi sebuah form yang divalidasi menggunakan JavaScript, user mengetikkan data lalu mengklik submit. Sebelum data dikirimkan ke server, data akan dicek terlebih dahulu pada browser menggunakan fungsi JavaScript yang ada pada halaman web. Sehingga jika memang data yang diisikan tidak valid, daripada membuang-buang waktu dengan mengirim data ke server baru divalidasi di server, lalu server mengirimkan respon balik mengenai ketidakvalidan input data user, lebih baik pengecekan validasi data form dilakukan secara lokal di browser menggunakan fungsi JavaScript.

Untuk lebih mempercepat dan menghemat waktu, implementasi terpopuler saat ini dari pemrograman JavaScript adalah teknik AJAX (Asynchronous JavaScript and XML). Teknik ini sering digunakan oleh aplikasi berbasis web seperti Gmail, Google Reader, dan lain-lain. Teknik AJAX ini membuat proses pertukaran data antara server dan browser terjadi di belakang layar sehingga interaksi antara user dan aplikasi web semakin responsif.

## 2. Kelebihan dan Kekurangan JavaScript

Kelebihan dari bahasa pemrograman JavaScript adalah sebagai berikut :

- Ukuran file kecil

Skrip dari JavaScript memiliki ukuran yang kecil sehingga ketika web yang memiliki JavaScript ditampilkan di browser, maka akses tampilannya akan lebih cepat dibandingkan

ketika browser membuka suatu web yang memiliki skrip Java. Hal ini juga sangat berkaitan

dengan daya kerja server. Semakin kecilspace suatu web yang disimpan dalam suatu server,

maka daya kerja server ketika di-browsing oleh user di internet akan tidak terlalu berat.

Selain

itu sifat JavaScript Client Side yang tidak perlu lagi ditangani oleh server ketika browser memanggil web dari sebuah server.

- Mudah untuk dipelajari

JavaScript merupakan bahasa semi pemrograman yang merupakan gabungan antara bahasa pemrograman Java dengan bahasa kode HTML sehingga disebut juga bahasa hybrid. Walaupun JavaScript merupakan turunan dari Java, namun JavaScript tidak memiliki aturan serumit Java.

- Terbuka

JavaScript tidak terikat oleh hardware maupun software tertentu, bahkan system operasi seperti

windows maupun unix. Karena ia bersifat terbuka, maka ia dapat dibuat maupun dibaca di

semua jenis komputer.

Sedangkan kekurangan JavaScript diantaranya adalah :

- Skrip tidak terenkripsi

Karena JavaScript bersifat client side, maka skrip yang kita buat di text editor dan telah dijadikan web di server, ketika user me-request web dari server tersebut, maka sintaks JavaScript akan langsung ditampilkan di browser. User bisa melihatnya dan menirunya dari source-nya.

- Kemampuan terbatas

Walaupun JavaScript mampu membuat bentuk web menjadi interaktif dan dinamis, namun JavaScript tidak mampu membuat program aplikasi sendiri seperti Java.

- Keterbatasan objek

JavaScript tidak mampu membuat kelas-kelas yang bisa menampung objek-objek tambahan

seperti Java, karena JavaScript telah memiliki objek yang built in pada struktur bahasanya.

### 3. Pola Penulisan JavaScript

Seperti yang telah dijelaskan pada pengertian di atas bahwa JavaScript merupakan bahasa

yang case sensitive seperti bahasa pemrograman Java, yaitu membedakan penulisan dengan

huruf kecil dan huruf besar memiliki arti yang berbeda. Dalam bahasa pemrograman JavaScript juga, sebagai contoh fungsi perintah penulisan variabel tidak boleh ditulis "Var" atau juga "VAR" (huruf besar semua), yang benar adalah "var" (huruf kecil semua). Perintah

lain adalah new Date tidak boleh ditulis new date (huruf kecil semua), dan sebagainya.

JavaScript dapat ditulis secara inline atau satu file dengan dokumen HTML atau juga dapat

ditulis pada file terpisah dengan ekstensi .js. JavaScript yang ditulis satu file dengan dokumen

HTML dapat ditulis sebagai berikut :

```
<html>
<head>.....</head>
<script type="text/javascript" language="JavaScript">
/* kode JavaScript dapat ditulis disini */
</script>
<body>.....</body>
</html>
```

atau dapat juga ditulis di tengah tag body sebagai berikut :

```
<html>
<head></head>
<body>.....
<script type="text/javascript" language="JavaScript">
/* kode JavaScript dapat ditulis disini */
</script>
</body>
</html>
```

Sedangkan jika JavaScript ditulis pada file terpisah dengan dokumen HTML maka kode dapat ditulis sebagai berikut :

```
<html>
<head></head>
<script type="text/javascript" language="JavaScript"
src="file_JavaScript.js"></script>
<body></body>
</html>
```

Untuk lebih memudahkan pemeliharaan halaman web maka sebaiknya kode JavaScript ditulis secara terpisah dari dokumen HTML sehingga jika diperlukan perubahan pada JavaScript, dokumen HTML tidak perlu diubah.

### 4. Sintak JavaScript

Beberapa sintaks pada JavaScript mirip dengan sintaks pada bahasa pemrograman Java seperti pada sintaks perulangan dan percabangan kondisi, namun JavaScript bukanlah

Java. Cara kerja JavaScript biasanya lebih mengarah pada membuat aksi jika sebuah elemen HTML mengalami perubahan, misalnya ketika mouse diklik.

### 5.1. Variabel dan Tipe Data

Variabel pada JavaScript ditulis sebagai berikut :

```
var nama_variabel = nilai_variabel;
```

misalnya sebagai berikut :

```
var text;
```

```
var count = 1;
```

```
var ErrorMessage = "Koneksi Gagal.";
```

```
var daysLeft = 5.7;
```

```
var boolean = true;
```

Variabel pada JavaScript berlaku untuk semua nilai. Nilai-nilai yang diizinkan pada variabel JavaScript sebagai berikut :

- Bilangan bulat atau integer

- Bilangan desimal atau float

- String, dengan karakter escape sebagai berikut :

- \n => untuk membuat garis baru

- \t => untuk membuat tab

- \" => untuk membuat tanda petik dua

- ' => untuk membuat tanda petik satu

- \\ => untuk membuat garis miring

### 5.2. Operator

q Operator adalah perintah yang memanipulasi nilai atau variable dan memberikan suatu hasil tertentu.

q Operand adalah Data-data yang dimanipulasi oleh operator untuk memberikan suatu hasil yang diinginkan

Sebagai contoh :

$$X + 3 = 6$$

Keterangan :

§ X dan 3 adalah operan

§ + adalah operator

§ 6 adalah hasil

Ada beberapa Jenis-jenis operator

- Jenis-Jenis Operator

### 1. Operator Unary

q Adalah operator yang hanya mengolah satu nilai atau satu

q Operator unary dalam javascript terbagi atas 3 macam, yaitu :

### 2. Operator Binary

q Adalah operator yang mengolah dua nilai atau dua operan.

q Operator Binary dalam javascript antara lain :

### 3. Operator Penugasan

q Adalah operator yang memberi nilai kepada variable.

q Jenis operator yang umum digunakan adalah tanda sama dengan (=).

Operator Penugasan dalam javascript antara lain

### 4. Operator Perbandingan

q Adalah operator yang digunakan untuk membandingkan satu nilai dengan nilai yang lain, untuk mendapatkan hasil benar atau salah.

q Operator Perbandingan dalam javascript antara lain :

### 5. Operator Logika

q Pada dasarnya Operator logika hampir sama dengan operator perbandingan yang berfungsi untuk membandingkan dua buah nilai untuk memberikan hasil benar atau salah.

q Perbedaannya adalah operator logika berfungsi membandingkan perbandingan .

q Perhatikan contoh di bawah ini :

( total > 10) && (biaya < 100)

- Pada contoh di atas digunakan operator && yang artinya adalah mengharuskan kedua nilai memenuhi syarat.

- Jika kedua nilai benar maka ekspresi di atas akan bernilai benar.

- Beberapa jenis operator logika yang terdapat di javascript adalah sebagai berikut :

### 5. Operator String

- Adalah operator yang mengolah nilai string, yaitu nilai yang berupa karakter (bukan angka atau ekspresi aritmatik).

Jenis operator string pada dasarnya sama dengan operator-operator yang telah dijelaskan di atas. Hanya saja, operator-operator tersebut digunakan untuk memanipulasi string. Perhatikan contoh di bawah ini :

```
var nama = "haidar" + "ali"
```

- Dari contoh di atas maka dapat diambil kesimpulan bahwa variable nama adalah string "haidar ali".

Ekspresi

### 5.3. Komentar

Penulisan komentar pada JavaScript sama dengan penulisan komentar pada bahasa pemrograman Java, yaitu sebagai berikut :

```
// isi komentar (untuk satu baris komentar)
```

-

```
/* isi komentar */ (untuk lebih dari satu baris komentar)
```

### 5.4. Array

Array pada pemrograman berfungsi seperti halnya sebuah tabel yang diakses elemennya menggunakan indeks. Penulisan kodearray pada JavaScript adalah sebagai berikut :

```
var x = new Array();
```

```
var x = new Array("merah", "kuning", "hijau", 1,5,8);
```

```
var x = new Array(6);
```

```
var x = ["merah", "kuning", "hijau", 1,5,8];
```

dengan cara pengaksesan elemenarray sebagai berikut :

```
var str = x[indeks];
```

### 5.5. Kondisi Percabangan

Kondisi percabangan pada JavaScript juga sama dengan pada bahasapemrograman Java, yaitu

sebagai berikut :

- Statemen if

```
if (kondisi) {
aksi_1
} else {
aksi_2
}
```

- Statemen depend on

```
Switch (nama_variabel) {
case nilai_var_1 : aksi_1
break;
case nilai_var_2 : aksi_2
break;
case nilai_var_n : aksi_n
break;
default : aksi_default
break;
}
```

### 5.6. Perulangan

Perulangan pada JavaScript juga sama dengan perulangan pada bahasapemrograman Java,

yaitu sebagai berikut :

- Perulangan for

```
For (nama_var = nilai_awal; nama_var opr_relasi; nama_var++) {
// proses perulangan disini
}
```

- Perulangan while

```
while (kondisi_perulangan) {
// proses perulangan disini
// iterasi
// kondisi terminasi
}
```

- Perulangan do-while

```
do {
// proses perulangan disini
// iterasi
} while (kondisi_perulangan);
```

### 5.7 Fungsi

Ada beberapa fungsi yang sering digunakan dalam JavaScript, yaitu sebagai berikut :

#### 1. Alert

Alert pada JavaScript adalah jendela kecil yang muncul untuk menyampaikan sebuah pesan

saat halaman web di-load atau pada saat pengunjung mengklik link yang berisi tombol ok dan

teks keterangan.

Contoh jendela alert

kode untuk memanggil jendela alert tersebut adalah sebagai berikut :

```
<script type="text/javascript" language="JavaScript">
alert('Selamat Datang');
</script>
```

#### 2. Confirm

Mirip alert, tetapi fungsi ini memiliki dua pilihan, yaitu "OK" dan "CANCEL".

Contoh jendela confirm :

Kode untuk memanggil jendela alert tersebut adalah sebagai berikut :

```
<script type="text/javascript" language="JavaScript">
confirm('Selamat Datang');
</script>
```

### 3. Prompt

Digunakan untuk memunculkan dialog atau kotak pesan saat halaman web di-load atau saat

pengunjung mengklik sebuah link.

Contoh jendela prompt :

Jika kita tidak mengetikkan sesuatu pada jendela prompt tersebut, maka akan muncul alert :

Kode untuk memanggil jendela prompt tersebut adalah sebagai berikut :

```
<script type="text/javascript" language="JavaScript">
while(prompt('Masukkan Nama Anda','')=="") {
alert('Anda belum memasukkan nama');
}</script>
```

## Statement

q merupakan kata kunci yang digunakan untuk mengendalikan alur program javascript yang kita buat.

q Pernyataan bisa dibuat sebagai kalimat lengkap yang menyatukan ekspresi, variable, dan nilai-nilai

yang telah kita pelajari pada modul-modul sebelumnya.

Pernyataan (statement) terbagi menjadi :

### Pernyataan if...else

ü Pernyataan ini digunakan untuk membandingkan suatu kondisi, jika kondisi itu benar, maka akan dijalankan ekspresi tertentu, tetapi jika kondisi itu salah akan dijalankan ekspresi lain.

ü Sintaks dari pernyataan if.....else :

**If (kondisi)**

**{ekspresi 1}**

**else (kondisi)**

**{ekspresi 2}**

Keterangan :

§ **Kondisi** : kondisi yang dibandingkan

§ **Ekspresi-1** : ekspresi-ekspresi yang akan dijalankan jika kondisi memenuhi syarat.

§ **Ekspresi-2** : ekspresi-ekspresi yang akan dijalankan jika kondisi tidak memenuhi syarat.

### Pernyataan switch..case

ü Instruksi ini memungkinkan kita untuk melakukan test berbagai nilai dari variabel yang sama.

ü Dengan cara ini kita bisa melakukan testing terhadap berbagai nilai variabel lebih sederhana daripada memakai instruksi if.

ü Sintaks dari pernyataan switch.....case :

**switch** (Variabel) {

**case** Nilai1:

blok instruksi;

**break;**

**case** Nilai2:

blok instruksi;

**break;**

**case** NilaiX:

blok instruksi;

**break;**

**default:**

blok instruksi;

**break;**

}

ü Kata **break** di dalam tanda kurung sesudah kata **switch** menunjukkan nama variabel yang akan di test pada kasus nilai yang berbeda.

ü Pada saat nilai variabel yang akan di test sama dengan nilai kasusnya (**case**) maka blok instruksi di bawahnya akan dieksekusi.

ü Kata **break** berarti berhentinya atau keluar dari struktur kondisi **switch**.

ü Kata **default** berarti blok instruksi dibawahnya akan dieksekusi bila nilai variabel yang masuk tidak sama dengan semua kasus (**case**) yang ada.

## Pernyataan while

ü Pernyataan **while** digunakan untuk melakukan suatu loop atau menjalankan ekspresi berulang

selama (**while**) kondisi yang ditentukan bernilai benar.

ü Sintak dari pernyataan **while** adalah sebagai berikut :

**While (kondisi)**

**(ekspresi)**

Keterangan :

§ **Kondisi** : kondisi yang dibandingkan

§ **Ekspresi-1** : ekspresi-ekspresi yang akan dijalankan jika kondisi memenuhi syarat.

Contoh :

Jika seorang nasabah bank ingin menabung di suatu bank dengan dana awal Rp.100.000,-. Dia tidak

akan berhenti menabung sampai kas saldonya mencapai satu juta.

Jika kita terjemahkan ke dalam bahasa javascript maka kalimatnya adalah :

```
var saldo = 100000
```

```
while (saldo <=1000000)  
(saldo+=100000)
```

Dari sintak di atas maka dapat di uraikan sebagai berikut :

§ **var saldo = 100000** ; artinya adalah variable saldo di beri nilai awal 100000

§ **while (saldo <= 1000000)**; artinya adalah memeriksa kondisi apakah isi variable saldo kurang dari atau sama dengan 1000000. Jika benar, maka pernyataan berikutnya akan dijalankan.

§ **(saldo +=100000)**; artinya sama dengan **saldo= saldo+100000**. Variable saldo yang baru diisi dengan variable uang 100000 maka akan ditambah dengan 100000 sehingga jumlahnya kni menjadi 200000, dan begitu seterusnya.

## Pernyataan for

ü Pernyataan **for** berfungsi melakukan suatu loop atau menjalankan ekspresi berulang-ulang untuk kondisi yang telah di tentukan sebelumnya.

ü Pernyataan **for** jika dilihat hampir sama dengan pernyataan **while**, yang berbeda adalah pernyataan **for** menyertakan kondisi awal dan kondisi akhir serta besarnya kenaikan.

ü Sintak dari pernyataan **for** adalah sebagai berikut :

**for (kondisi\_awal; kondisi \_akhir; kenaikan)**

**(ekspresi)**

Keterangan :

§ **kondisi\_awal** : kondisi awal yang memenuhi untuk loop dimulai

§ **kondisi\_akhir** : kondisi akhir yang memenuhi untuk loop diakhiri

§ **kenaikan** : besarnya kenaikan nilai variable untuk setiap iterasi loop

## Pernyataan break

ü Pernyataan (*statement*) **break** berfungsi sebagai menghentikan ekspresi loop yang sedang dijalankan, dan melanjutkan ke baris skrip di bawah loop tersebut.

ü Jika pengulangannya belum selesai sampai pada *statement* yang di syaratkan maka ia akan tetap

me-loop, dan berhenti pada kondisi yang telah ditetapkan di dalam sintak javascript.

ü Perhatikan contoh di bawah ini :

### Pernyataan Continue

ü Pernyataan **continue** berfungsi untuk menghentikan eksekusi loop. Tetapi hal ini tidak menyebabkan keluar dari loop tetapi hanya melompati satu buah iterasi dan melanjutkan iterasi loop yang selanjutnya.

### Pernyataan For...in

ü Pernyataan **for...in** adalah pernyataan yang melakukan loop(pengulangan) di dalam sebuah objek.

ü Sintak dari pernyataan for...in adalah :

**for (counter in nama\_object)**

**{ekspresi}**

Keterangan :

- **Counter** : jumlah counter di dalam object (jumlah property)
- **Nama\_object** : objek yang ingin dilakukan loop
- **Ekspresi** : ekspresi yang dijalankan selama loop

### Pernyataan var

ü Pernyataan var berfungsi untuk menugaskan suatu nilai ke dalam variable.

ü Sintak dari pernyataan var adalah :

**var nama\_variable = value**

contoh :

var **nama** = "firdaus"

var **a** = 5

- Pada pertemuan sebelumnya, sebenarnya kita telah mempelajarinya yaitu pada modul variable mengenai nilai dari variable.
- Sebenarnya kita bisa memberikan nilai untuk variable tanpa pernyataan var tersebut, seperti :

**nama="firdaus"**

### Pernyataan komentar

ü Pernyataan komentar berfungsi untuk menerangkan atau memberi keterangan pada setiap barisnya yang ada di dalam sintak HTML.

ü Pernyataan ini memudahkan bagi user untuk mempelajari script javascript yang kita buat, selain itu komentar yang kita buat dalam script tersebut tidak akan terlihat di *browser* dan hanya bisa terlihat di *source*-nya saja, karena *browser* tidak ikut memproses sintak komentar tersebut.

ü Sintaknya ada dua yaitu :

1. //komentar

2. /\* komentar

komentar \*/ (untuk lebih dari satu baris komentar)

# **Pertemuan 11**

## **11.1 PROJECT**

Kerjakan Lah Project Pembuatan WEBSITE Company Profile