





# DAFTAR ISI

DAFTAR ISI.....	1
PENGANTAR UMUM .....	3
MODUL 1.....	5
WEB PAGES.....	5
1.1 Struktur Dasar Web Page .....	6
1.2 Tags Dasar dalam Web Page.....	6
MODUL 2.....	9
DEEPER WITHIN WEB PAGES.....	9
2.1. HTML Elemen .....	10
2.2. HTML Attribute .....	10
MODUL 3.....	14
How to display .....	14
3.1. HTML Text Formatting.....	15
3.2 HTML Styles .....	16
Latihan Modul 3 .....	19
MODUL 4.....	20
COnent focus .....	20
4.1. HTML List.....	21
4.2. HTML Links.....	22
4.3. HTML Frames.....	23
Latihan Modul 4 .....	24
MODUL 5.....	25
Interaction .....	25
5.1. HTML Forms .....	26
Latihan Modul 5 .....	30
MODUL 6.....	31
MORE INTERACTION.....	31
6.1. Scripting Variables .....	32
6.2. Scripting Arithmetic.....	33
6.3. Scripting Comparison .....	35
Latihan Modul 6 .....	47
MODUL 7.....	48
AND MORE INTERACTION.....	48
7.1. Scripting Loops.....	49
Latihan Modul 7 .....	58
MODUL 8.....	59
WEB STYLISH .....	59

8.1. CSS Syntax .....	60
8.2. CSS Id dan Class .....	61
8.3. CSS Formatting .....	62
Latihan Modul 8.....	67

# PENGANTAR UMUM

Pengembangan sistem informasi berbasis web dari hari ke hari semakin cepat dan kompleks. Agar perkembangan tersebut dapat berjalan dengan baik dan handal maka diperlukan pembelajaran yang baik dan teratur. Kemampuan untuk membuat sebuah aplikasi web tidaklah datang dengan sendirinya tanpa adanya proses pembelajaran yang tepat.

Pada modul ini Anda akan belajar bagaimana membuat sebuah aplikasi web mulai dari mendesain halaman web, menyisipkan gambar dan tabel pada halaman web hingga aplikasi web tersebut dapat digunakan sesuai dengan kebutuhan. Ada 8 (delapan) pertemuan yang dapat Anda gunakan untuk belajar membuat sebuah aplikasi web.

**Modul 1**, akan mempelajari tentang apa saja tag - tag yang ada di dalam pembuatan aplikasi web, serta membuat sebuah halaman web kecil dengan menggunakan notepad.

**Modul 2**, setelah kita mengenal apa saja tag - tag, pada modul kedua ini kita akan mencoba menganalisa fungsi dari sebuah elemen dan attribute.

**Modul 3**, pada modul ini kita akan belajar tentang cara membuat tabel, menyisipkan gambar, memformat text dan menambahkan style supaya halaman web yang kita buat tampak lebih indah dan teratur.

**Modul 4**, sekarang kita sudah bisa membuat sebuah halaman web, namun supaya halaman web kita bersifat dinamis, kita harus menghubungkan halaman utama dengan halaman - halaman yang lainnya.

**Modul 5, 6, dan 7**, pada 3 modul ini kita akan belajar tentang interaksi web. Yang dimaksud dengan interkasi adalah halaman web kita dapat menerima input yang berupa data dan dapat menampilkan output yang merupakan data setelah diolah.

**Modul 8**, bagian terakhir modul ini anda akan belajar tentang syntax dari Caschading Style Sheet (CSS) dan bagaimana mengelola halaman web yang telah kita buat dengan menggunakan CSS tersebut.

Tools yang digunakan untuk praktikum desain dan pemrograman web ini adalah segala editor teks yang support dengan bahasa HTML, CSS dan bahasa scripting serta browser yang support dengan HTML 4.0.1.

Beberapa editor teks yang disediakan di Laboratorium Komputer:

- ✓ Notepad
- ✓ Notepad++
- ✓ Visual Studio 2005

Sedangkan browser yang disediakan untuk uji coba halaman web:

- ✓ Internet Explorer
- ✓ Mozilla Firefox

Setelah selesai praktikum simpan konfigurasi dan desain halaman web anda ke komputer server yang digunakan untuk menyimpan data praktikum anda.

Selamat mencoba, semoga berhasil !

Setiap masalah pasti ada jalan keluarnya ☺

Surabaya, Oktober 2010

**Co - Ass Web Labkom**

e-mail : ry4nn4nd4@gmail.com

# MODUL 1

## WEB PAGES

*Percayalah pada kemampuan diri sendiri..  
Karena dengan percaya segalanya akan terasa mudah..  
--Ryan Nanda--  
Co – Ass Web Labkom*



### **Tujuan**

Praktikan dapat mengerti dan memahami konsep Web Pages serta mampu mengimplementasikan pada aplikasi web yang dibuat.

### **Materi**

Mengenal HTML, Struktur Dasar Web Pages,

### **Referensi**

<a href="http://www.w3schools.com">http://www.w3schools.com</a> ■
<a href="http://www.w3schools.com/html/html_intro.asp">http://www.w3schools.com/html/html_intro.asp</a> ■
<a href="http://www.hscript.com/html/html_1/index.html">http://www.hscript.com/html/html_1/index.html</a> ■

## 1.1 Struktur Dasar Web Page

Sebuah halaman web dibangun menggunakan HTML yang memiliki struktur dasar. Halaman yang selalu dimulai dengan tag awal **html elemen** dan selalu berakhir dengan tag akhir **html** elemen sebagai berikut:

```
<html>
<head>
  <title>.WELCOME</title>
</head>
<body>
  Tolong
</body>
</html>
```

Simpan nama file tersebut dengan nama WEBPAGE.html  
Ekstensi dari HTML adalah \*.html

## 1.2 Tags Dasar dalam Web Page

Halaman web dapat dianggap sebagai pohon dokumen yang terdapat beberapa cabang. Ada aturan seperti apa item setiap cabang dapat berisi (dan ini rinci dalam elemen's referensi masing-masing dalam "Berisi" dan "Terdapat oleh" bagian). Untuk memahami konsep pohon dokumen, ini berguna untuk mempertimbangkan sebuah halaman web sederhana dengan fitur-fitur konten khas bersama tampilan struktur pohon, sebagaimana ditunjukkan pada Gambar 1 .



**Gambar 1.1 Contoh Struktur Halaman Web**

Jika kita melihat perbandingan ini, kita dapat melihat bahwa *html* elemen sebenarnya mengandung dua elemen : *Head* dan *Body*.

*Head* memiliki dua kantor cabang pembantu sebuah *meta* elemen dan *title*.

*Body* elemen berisi sejumlah judul, paragraph, dan Blockquote.

### **Latihan Modul 1**

Web Page dasar mengimplementasikan semua tags dasar yang sudah dipelajari di pertemuan ini.

# MODUL 2

## DEEPER WITHIN WEB PAGES

*Jika kita mampu mengalahkan rasa ingin menyerah atau putus asa,  
pasti kita telah membuka satu kemungkinan untuk sukses.*

*--Andrie Wongso --*

*Action & Wisdom Motivation Training*



### Tujuan

Praktikan mengerti dan memahami konsep Deeper Within Web Pages serta mampu mengimplementasikan dalam aplikasi web.

### Materi

HTML Elemen, HTML Attribute

### Referensi

[http://en.wikipedia.org/wiki/HTML\\_element](http://en.wikipedia.org/wiki/HTML_element) ■

[http://www.w3schools.com/html/html\\_attributes.asp](http://www.w3schools.com/html/html_attributes.asp) ■

## 2.1. HTML Elemen

Dalam Dokumen HTML, elemen dibagi menjadi dua kategori utama, yaitu elemen - elemen <HEAD> yang memberikan informasi tentang dokumen, seperti judul dokumen atau hubungannya dengan dokumen lain. Dan elemen - elemen <BODY> yang menentukan bagaimana isi suatu dokumen yang ditampilkan oleh browser, seperti paragraph, list, daftar, tabel dan lain - lain. Di dalam elemen HTML terdiri dari tag - tag beserta yang ada dalam tag - tag tersebut. Berikut ini adalah elemen dasar HTML:

- Paragraph (<p> ... </p>)
- Blockquote (<blockquote> ... </blockquote>).
- Preformatted (<pre> ... </pre>)
- Divider (<div> ... </div>).

## 2.2. HTML Attribute

Attribute digunakan untuk memperkuat tag. Yang dimaksud dengan memperkuat adalah ketika sebuah web browser menampilkan hasil tag yang anda buat, browser juga akan mencari attribute set dan kemudian menampilkan elemen (Tags + Attribute) secara keseluruhan. Pada titik tertentu anda mungkin ingin memberikan background atau mungkin mengubah lebar dan tinggi tabel, semua itu dapat dilakukan dengan menggunakan attribute.

### 2.2.1 Attribute Id:

Attribute Id tidak berperan langsung dalam format elemen anda, tetapi akan lebih berperan pada saat *Scripting* dan *Cascading Style Sheet* (CSS). Untuk mengklasifikasi dan mengidentifikasi unsur -unsurnya, akan lebih nyata apabila anda belajar CSS.

Kuncinya adalah anda klasifikasikan Id di dalam CSS. Hal ini penting ketika anda memiliki dua atau lebih dari unsur yang sama pada 1 halaman, seperti contoh tag <p> di satu paragraph anda ingin menulis teks dengan type italic, tetapi di paragraph yang lain anda ingin menulis teks dengan type Bold. Maka agar keinginan anda itu dapat terjadi, anda harus menggunakan attribute.

**Syntax:**

```
<p id="italicsparagraph"> Teks </p>
```

- ✓ Membuat huruf miring.

### 2.2.2 Attribute Nama

Attribute nama jauh berbeda dari id dan kelas. Attribute nama berfungsi sebagai penanda layaknya setiap manusia mempunyai nama yang digunakan untuk membedakan antara satu dengan yang lainnya. Attribute nama biasa digunakan sebagai variable scripting untuk bahasa script seperti javascript, ASP, dan PHP pada textbox, button, checkbox, radiobutton dan control – control yang lain.

**Syntax:**

```
<input type="teks" name="TeksField1" />
```

- ✓ Membuat control textbox dengan nama TextField1

### 2.2.3 Attribut Judul

Attribute judul ini berfungsi untuk membuat tek dengan ukuran –ukuran tertentu. Attribute judul mempunyai 6 tags. Masing – masing tags mempunyai fungsi yang berbeda. Attribute - atribut ini biasa digunakan untuk penulisan teks judul dan sub judul.

**Syntax:**

```
<h1 title="Judul1"> Teks </h1>
```

- ✓ Membuat teks dengan ukuran besar

```
<h2 title="Judul2"> Teks </h2>
```

- ✓ Membuat teks dengan ukuran lebih kecil dari h1

```
<h3 title="Judul3"> Teks </h3>
```

- ✓ Membuat teks dengan ukuran lebih kecil dari h2

```
<h4 title="Judul4"> Teks </h4>
```

- ✓ Membuat teks dengan ukuran lebih kecil dari h3

```
<h5 title="Judul5"> Teks </h5>
```

- ✓ Membuat teks dengan ukuran lebih kecil dari h4

```
<h6 title="Judul6"> Teks </h6>
```

- ✓ Membuat teks dengan ukuran lebih kecil dari h5

### 2.2.4 Attribute rata

Attribute rata ini berfungsi untuk merapikan teks paragraph yang anda buat. Anda dapat membuat teks paragraph menjadi rata kiri (Left), rata tengah (Center), maupun rata kanan (Right) dengan menggunakan attribute rata ini sesuai dengan kebutuhan anda.

#### Syntax:

`<p align="Left">` Teks posisi rata kiri `</p>`

`<p align="Center">` Teks posisi rata tengah `</p>`

`<p align="Right">` Teks posisi rata kanan `</p>`

## Latihan Modul 2

# MODUL 3

## HOW TO DISPLAY

*Bersyukurlah atas apa yang telah kita dapatkan..  
Agar segalanya terasa bermakna..  
--Co - Ass Web--*



### Tujuan

Praktikan dapat memahami dan membuat ulang halaman web serta memanfaatkannya dalam aplikasi web

### Materi

Html Text Formatting, HTML Styles, HTML Images, HTML Tables

### Referensi

<a href="http://www.w3schools.com/html/html_formatting.asp">http://www.w3schools.com/html/html_formatting.asp</a>	■
<a href="http://www.w3schools.com/html/html_styles.asp">http://www.w3schools.com/html/html_styles.asp</a>	■
<a href="http://www.w3schools.com/html/html_images.asp">http://www.w3schools.com/html/html_images.asp</a>	■
<a href="http://www.w3schools.com/html/html_tables.asp">http://www.w3schools.com/html/html_tables.asp</a>	■

### 3.1. HTML Text Formatting

Html text formatting adalah pengaturan format teks pada halaman web yang anda buat.

**This text is bold**

**This text is big**

*This text is italic*

`This is computer output`

This is <sub>subscript</sub> and <sup>superscript</sup>

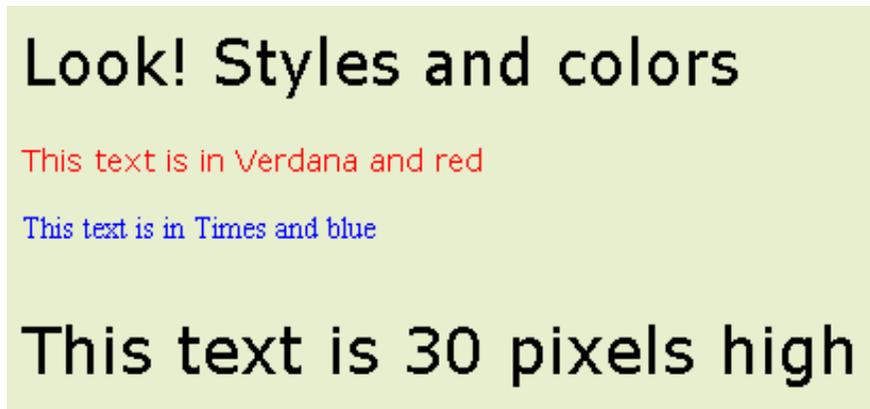
**Gambar 2.1 Contoh Format text pada Halaman web.**

Beberapa format text yang dapat digunakan adalah sebagai berikut:

Tag	Description
<u>&lt;b&gt;</u>	Defines bold text
<u>&lt;big&gt;</u>	Defines big text
<u>&lt;em&gt;</u>	Defines emphasized text
<u>&lt;i&gt;</u>	Defines italic text
<u>&lt;small&gt;</u>	Defines small text
<u>&lt;strong&gt;</u>	Defines strong text
<u>&lt;sub&gt;</u>	Defines subscripted text
<u>&lt;sup&gt;</u>	Defines superscripted text
<u>&lt;ins&gt;</u>	Defines inserted text
<u>&lt;del&gt;</u>	Defines deleted text

### 3.2 HTML Styles

Hampir sama dengan format teks, style digunakan untuk menata gaya penulisan yang ada pada sebuah web, hanya saja dalam penulisan attributenya yang berbeda.



Gambar 2.2 Contoh hasil penggunaan Styles

Beberapa tag untuk styles yang dapat digunakan adalah seperti pada gambar di bawah ini:

Tags	Description
<center>	Defines centered content
<font> and <basefont>	Defines HTML fonts
<s> and <strike>	Defines strikethrough text
<u>	Defines underlined text
Attributes	Description
align	Defines the alignment of text
bgcolor	Defines the background color
color	Defines the text color

Untuk contoh penggunaan style dalam sebuah halaman web adalah sebagai berikut:

**Syntax:**

```
<html>
<Body style="Background-color : yellow;">
<h2 Style=" Font-family : Arial;">
This is Heading
</h2>
<p Style="Font-family : Arial; color : red; Font-size : 20px;">
This is Paragraph
</p>
</Body>
</html>
```

### 3.3 HTML Images

Html Images berfungsi untuk menambahkan sebuah gambar pada halaman web yang anda buat. Pada dasarnya letak direktori file gambar yang akan anda tambahkan mempengaruhi penulisan url direktori.

**Syntax:**

```

    ✓ Untuk menambahkan gambar yang letak file gambarnya
      terdapat di dalam satu direktori dengan file html web anda.

    ✓ Untuk menambahkan gambar yang letak file gambarnya tidak
      terdapat di dalam satu direktori dengan file html web anda.
```

### 3.4 HTML Tables

HTML Tables berfungsi untuk menambahkan sebuah tabel yang baris dan kolomnya dapat anda sesuaikan dengan kebutuhan tabel yang ingin anda buat. Elemen table terdiri dari Tr, Th dan Td. Dimana fungsi dari masing - masing elemen berbeda. Tr digunakan untuk membuat baris, Th digunakan untuk membuat header, sedangkan Td digunakan untuk membuat kolom. Untuk lebih jelasnya dapat anda coba langsung syntax di bawah ini:

**Syntax:**

```
<Table border="1">  
<Th colspan="5" align="Center"> Header </Th>  
<Tr>  
<Td> Kolom 1 </Td>  
<Td> Kolom 2 </Td>  
<Td> Kolom 3 </Td>  
<Td> Kolom 4 </Td>  
<Td> Kolom 5 </Td>  
</Tr>  
<Tr>  
<Td Rowspan ="2"> A </Td>  
<Td Colspan ="4"> B </Td>  
</Tr>  
<Td Colspan ="2"> C </Td>  
<Td Colspan ="2"> D </Td>  
</Table>
```

**Keterangan Attribute:**

- Rowspan
  - ✓ Menggabungkan (Merge) Baris
- Colspan
  - ✓ Menggabungkan (Merge) Kolom

**Hasil :**

Header				
<u>Kolom 1</u>	Kolom2	<u>Kolom 3</u>	<u>Kolom 4</u>	Kolom5
A	B			
	C		D	

## **Latihan Modul 3**

# MODUL 4

## CONTENT FOCUS

*Jangan pernah takut pada kegagalan..  
Karena kegagalan adalah proses untuk mencapai kesuksesan..  
--Ryan Nanda--  
Co - Ass Web Labkom*



### **Tujuan**

Praktikan dapat memahami dan membuat sebuah halaman web dan TOC content dari halaman web tersebut

### **Materi**

HTML List, HTML Links, HTML Frames

### **Referensi**

<a href="http://www.w3schools.com/html/html_List.asp">http://www.w3schools.com/html/html_List.asp</a> ■
<a href="http://www.w3schools.com/html/html_Links.asp">http://www.w3schools.com/html/html_Links.asp</a> ■
<a href="http://www.w3schools.com/html/html_frames.asp">http://www.w3schools.com/html/html_frames.asp</a> ■

## 4.1. HTML List

HTML List berfungsi untuk membuat daftar urutan dengan menggunakan penomoran maupun symbol. Di dalam HTML List terdapat 3 macam jenis list (daftar), yaitu:

- **Ordered List**

Ordered List berfungsi untuk membuat daftar list dimana tiap bagian akan diberi penomoran, tag diawali dengan **<ol>** dan diakhiri dengan **</ol>**. Untuk tiap bagian diawali dengan **<li>** dan diakhiri dengan **</li>**. Penomoran list secara default akan diawali dengan angka 1 / dengan attribute **Type = 1** .

Macam - macam attribute untuk penomoran yang lain:

- ✓ Jika anda ingin format penomoran dengan huruf A, B, C maka attribute yang digunakan adalah **Type = A**
- ✓ Jika anda ingin format penomoran dengan huruf i, ii, iii maka attribute yang digunakan adalah **Type = i**
- ✓ Jika anda ingin format penomoran dengan huruf a, b, c maka attribute yang digunakan adalah **Type = a**
- ✓ Jika anda ingin format penomoran dengan huruf I,II,III maka attribute yang digunakan adalah **Type = I**

**Syntax:**

```
<ol type = A>  
<li>Yamaha</li>  
<li>Honda</li>  
<li>Suzuki</li>  
</ol>
```

- **Unordered List**

Unordered List berfungsi untuk membuat daftar list dimana tiap bagian tidak akan diberi penomoran, tag diawali dengan **<ul>** dan diakhiri dengan **</ul>**. Untuk tiap bagian diawali dengan **<li>** dan diakhiri dengan **</li>**. Tag **<ul>** secara default akan menggunakan lingkaran / dengan attribute **Type = Circle**.

Macam - macam attribute untuk type yang lain:

- ✓ Untuk type Cakram maka attribute yang digunakan adalah **Type = Disk**
- ✓ Untuk type Kotak maka attribute yang digunakan adalah **Type = Square**

Tag `<ul>` juga dapat digunakan tanpa menggunakan `<li>` jika kumpulan teks hanya membentuk satu blok.

**Syntax:**

```
<ul type = square>
<li>Yamaha</li>
<li>Honda</li>
<li>Suzuki</li>
</ul>
```

- **Definitions List**

Definitions List berfungsi untuk membuat daftar list item sekaligus dengan deskripsi dari item tersebut. Tag diawali dengan `<dl>` dan diakhiri dengan `</dl>`. Untuk tiap bagian diawali dengan `<dt>` dan diakhiri dengan `</dt>`. Untuk deskripsi tiap bagian tag diawali dengan `<dd>` dan diakhiri dengan `</dd>`.

**Syntax:**

```
<dl>
<dt>Yamaha</dt>
<dd>Merk Kendaraan</dd>
<dt>Honda</dt>
<dd>Juga Merk Kendaraan</dd>
</dl>
```

## 4.2. HTML Links

HTML Links berfungsi untuk menghubungkan halaman lain pada web yang anda buat, link juga dapat menghubungkan web yang anda buat dengan halaman web yang lain. Tag perintah menggunakan anchor `<a>` dan dikombinasikan dengan attribute `href="Url"` yang berfungsi untuk mengarahkan kemana web anda akan berpindah halaman (halaman yang dituju). Pada tag anchor ini juga dapat ditambahkan attribute untuk pengaturan warna font dan `style / type font`.

**Syntax:**

```
<a href=" www.facebook.com" style="color : red; text-decoration : none"
> Facebook </a>
```

### 4.3. HTML Frames

Html Frames berfungsi untuk menampilkan lebih dari satu dokumen HTML di jendela browser yang sama. Atau lebih tepatnya terdapat suatu bagian di dalam halaman web anda yang dapat berubah - ubah (dinamis).

Tag	Description
<code>&lt;frameset&gt;</code>	Defines a set of frames
<code>&lt;frame /&gt;</code>	Defines a sub window (a frame)
<code>&lt;noframes&gt;</code>	Defines a noframe section for browsers that do not handle frames
<code>&lt;iframe&gt;</code>	Defines an inline sub window (frame)

#### 4.3.1 FrameSet

Frameset adalah gabungan dari beberapa frame yang dijadikan dalam satu frame / tempat.

##### Syntax:

```
<frameset cols="25%,75%">  
<frame src="Image.html">  
<frame src="Detail.html">  
</Frameset>
```

##### Penjelasan Syntax:

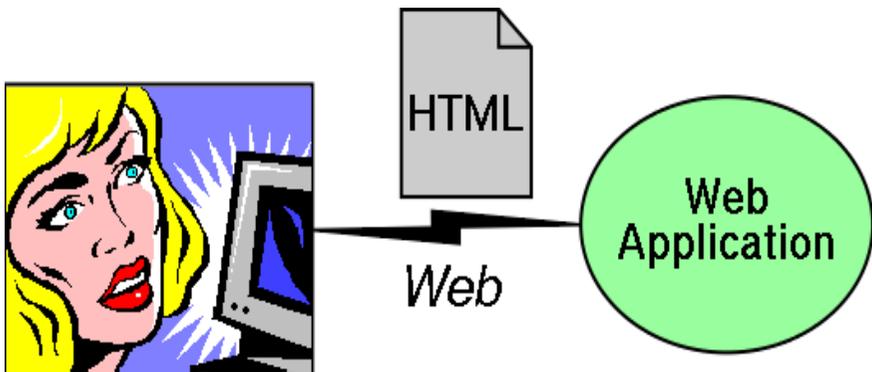
Frameset dibagi menjadi 2 kolom. Kolom pertama diberi ukuran 25% dari ukuran frameset, Kolom kedua diberi ukuran 75% dari ukuran frameset. Dokumen "Image.html" dimasukkan ke dalam kolom pertama, dokumen "detail.html" dimasukkan ke dalam kolom kedua

## **Latihan Modul 4**

# MODUL 5

## INTERACTION

*The only real mistake is the one from which we learn nothing.  
Kesalahan satu-satunya adalah kita tidak mau belajar  
--John Powell--*



### Tujuan

Praktikan mengerti dan dapat membuat sebuah halaman web dengan interaksi input dan script dasar

### Materi

HTML Forms, HTML Inputs, HTML Scripting

### Referensi

<a href="http://id.wikipedia.org/wiki/VBScript">http://id.wikipedia.org/wiki/VBScript</a> ■
<a href="http://id.wikipedia.org/wiki/JavaScript">http://id.wikipedia.org/wiki/JavaScript</a> ■
<a href="http://www.w3schools.com/HTML/Html_forms.asp">http://www.w3schools.com/HTML/Html_forms.asp</a> . ■

## 5.1. HTML Forms

Pada dasarnya tag `<Form>` dan `<Input>` digunakan bersama – sama untuk meminta masukan dari user kemudian dikirimkan ke server.

Tag `<Form>` menyediakan kerangkanya.

Tag `<Input>` menyediakan elemennya.

Standar penulisan form adalah sebagai berikut:

```
<Form Method = "POST / GET" Action = "URL">
```

```
...
```

```
</Form>
```

Attribute *Method* memiliki dua nilai, yaitu *POST* dan *GET*.

Method *GET* berfungsi untuk mengirimkan data pada server dengan cara meletakkan data pada bagian akhir **URL** (*Uniform Resource Locator*) yang ditunjuk.

Method *POST* berfungsi untuk mengirimkan datanya secara terpisah. Jika data masukkan terlalu banyak, disarankan menggunakan method *POST*.

Attribut *ACTION* berisi URL dari program / dokumen yang berisi scripting yang dipanggil oleh form tersebut.

## 5.2 HTML Input

Macam – macam control input:

- **TextBox**  
`<Input Type = "text"/>`
- **CheckBox**  
`<Input Type="ChekBox" />`
- **RadioButton**  
`<Input Type="Radio"/>`
- **Submit Button**  
`<Input Type="Submit"/>`
- **Reset Button**  
`<Input Type="Reset">`

**Syntax:**

```
<Form Action = "CekMain.php" Method = "POST">
<Table>
<Tr>
<Td>
<Input type = "text" name = "Username" />
<Input type = "Password" name = "Password" />
</Td>
</Tr>
<Tr>
<Td>
<Input type = "Submit" name = "Simpan" value = "Simpan" />
<Input type = "Reset" name = "Reset" value = "Reset" />
</Td>
</Tr>
</Table>
```

### 5.3 HTML Scripting

HTML Script diawali dengan tag <Script> yang berfungsi untuk mendefinisikan script dengan bahasa client - side. Misal javascript ataupun vbscript.

- **Javascript**

Javascript adalah bahasa script yang populer di internet dan dapat bekerja di sebagian besar penjelajah web browser seperti Internet Explorer (IE), Mozilla Firefox, Netscape, opera dan web browser lainnya.

Kode javascript biasa dituliskan dalam bentuk fungsi (*Function*) yang ditaruh di bagian dalam tag <Head> yang dibuka dengan tag <Script type = "text/javascript">

**Penulisan Javascript:**

```
<Script type = "text/javascript">
Alert ("Halo Dunia .. !")
</script>
```

Kode javascript juga bisa diletakkan di file tersendiri yang berekstensi .Js(Singkatan dari JavaScript). Untuk memanggil kode javascript yang terdapat di file tersendiri, di bagian awal

<Head> harus ditentukan dahulu nama file .Js yang dimaksud. Untuk lebih jelasnya lihat contoh kode di bawah ini:

```
<Head>
<Script type ="text/javascript" src="namafile.Js">
</script>
</Head>
```

### **Penggunaan Script pada bagian head:**

Pada dasarnya script akan dieksekusi ketika dipanggil, biasanya berbentuk fungsi (*Function*), atau dipanggil berdasarkan *trigger* pada *event* tertentu. Jika anda meletakkan script pada bagian head, maka script akan dijalankan terlebih dahulu ketika halaman web anda dijalankan.

```
<html>
<head>
<script type="text/javascript">
...
</script>
</head>
<body>
</body>
</html>
```

### **Penggunaan Script pada bagian body:**

Jika anda meletakkan script pada bagian body, maka script akan dijalankan secara bersamaan dengan syntax yang ada pada bagian body, atau dapat disimpulkan bahwa script dan isi body dijadikan satu bagian.

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
...
</script>
</body>
</html>
```

- **Vbscript**

*Vbscript* atau lebih lengkapnya **Visual Basic Script** Edition adalah sebuah bahasa script yang diinterpretasikan saat dijalankan / dieksekusi yang dikembangkan oleh **Microsoft Corporation** pada tahun 1996 yang seringnya digunakan dalam penjelajah web **Internet Explorer**(mulai Internet Explorer 3.0). *Vbscript* merupakan sebuah bahasa script turunan dari bahasa pemrograman *Visual Basic for Application* (VBA) yang digunakan di **Microsoft Office** dan beberapa platform pengembangan buatan Microsoft lainnya. *Vbscript* menghilangkan beberapa fungsi dari VBA, seperti halnya fungsi Input / Output dan akses langsung terhadap system operasi untuk menyediakan sebuah platform yang aman untuk mengembangkan aplikasi berbasis web dengan menggunakan platform *Active Server Pages* (ASP).

*Vbscript* tidak dapat digunakan untuk membuat program yang berdiri sendiri. *Vbscript* harus dimasukkan ke dalam sebuah berkas HTML. Ketika Internet Explorer membuka dokumen berkas HTML tersebut, *vbscript* dapat melakukan fungsi yang sama seperti *javascript* dan dapat dieksekusi. *VBScript* juga dapat digunakan untuk membuat sebuah alikasi HTML.

Skrip *vbscript* yang dimasukkan di dalam berkas HTML harus dimasukkan di antara tag <SCRIPT>...</SCRIPT>.

Berikut ini adalah contoh kode dari *vbscript*:

```
<Input type="Button" value = "Simpan" name = "Btn_Simpan">  
<script language = "vbscript">  
    Sub Btn_Simpan_onclik  
        MsgBox("Data tersimpan !")  
    End Sub  
</script>
```

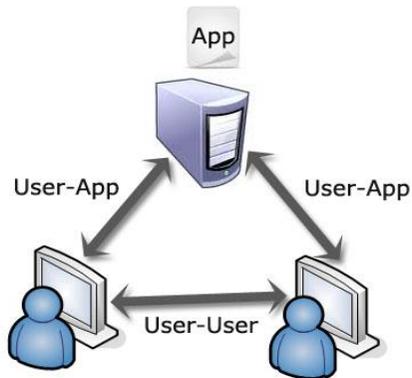
Penjelajah web form lainnya seperti **Mozilla Firefox**, **Opera** tidak memiliki dukungan terhadap *vbscript* secara langsung. Karena hal ini para pengembang website lebih memilih untuk menggunakan bahasa *javascript*.

## **Latihan Modul 5**

# MODUL 6

## MORE INTERACTION

*Whether you think you can or whether you think you can't, you're right!*  
--Henry Ford--



### Tujuan

Praktikan dapat memahami dan membuat halaman web yang mampu menangani interaksi yang telah ditentukan

### Materi

**Scripting Variables, Arithmetic, Comparison, Branching, PopUp, Function**

### Referensi

<a href="http://www.tizag.com/javascriptT/javascriptvariables.php">http://www.tizag.com/javascriptT/javascriptvariables.php</a>	■
<a href="http://www.w3schools.com/js">http://www.w3schools.com/js</a>	■
<a href="http://www.pageresource.com/jscript/jvar.htm">http://www.pageresource.com/jscript/jvar.htm</a>	■

## 6.1. Scripting Variables

Scripting Variables berfungsi untuk menyimpan suatu nilai / informasi yang dapat digunakan kembali. Variabel adalah nama simbolis yang mewakili beberapa data yang anda tetapkan.

Apakah anda ingat tentang pelajaran aljabar waktu di banku sekolah?

$X = 5$

$Y = 6$

$Z = X + Y$

Dalam penulisan diatas dapat diartikan bahwa X dapat digunakan untuk menyimpan nilai 5, sedangkan Y dapat digunakan untuk menyimpan nilai 6. Dan anda juga dapat menghasilkan nilai dari Z yaitu 11 yang didapat dari penjumlahan X dan Y.

Fungsi dari variabel sama seperti contoh aljabar diatas. Sebuah variabel dapat memiliki nama pendek seperti Z, atau nama yang lebih deskriptif seperti Hasil.

Aturan nama untuk variabel Javascript:

- Nama-nama variabel adalah case sensitif (y dan Y adalah dua variabel yang berbeda)
- Nama-nama variabel harus dimulai dengan huruf atau karakter garis bawah

### Pendeklarasian variable:

Untuk mendeklarasikan sebuah variable diawali dengan kata kunci **var**, agar lebih jelas penulisan syntaxnya anda dapat melihat contoh penulisan variable dibawah ini:

```
Var angka;
```

```
Var nama;
```

```
Var angka = 5;
```

```
Var nama = "Ryan";
```

**Catatan:** Apabila anda ingin menyimpan nilai dalam bentuk text / huruf, maka anda harus memberikan tanda kutip di awal dan di akhir text / huruf tersebut.

Setelah anda menjalankan script diatas maka variabel angka akan menyimpan nilai **5**, sedangkan variabel nama akan menyimpan nilai **"Ryan"**.

**Variabel boolean :**

Variabel boolean hanya mempunyai dua nilai yaitu **true** dan **false**. Dalam penulisannya tidak menggunakan tanda kutip.

```
Var Ryan_Cakep = True;
Var Ryan_Jelek = False;
```

**Null Value :**

Jika anda menyimpan nilai null pada sebuah variabel, maka dapat diartikan bahwa nilai dari variabel tersebut bukan nol ataupun kosong. dibawah ini cara penulisannya:

```
Var Uangku = null;
```

**6.2. Scripting Arithmetic**

Scripting Arithmetic berfungsi untuk melakukan fungsi aritmatika antara nilai - nilai yang disimpan oleh variabel.

**Arithmetic Operators**

Operator	Description	Example	Result
+	Addition	$x=y+2$	$x=7$
-	Subtraction	$x=y-2$	$x=3$
*	Multiplication	$x=y*2$	$x=10$
/	Division	$x=y/2$	$x=2.5$
%	Modulus (division remainder)	$x=y\%2$	$x=1$
++	Increment	$x=++y$	$x=6$
--	Decrement	$x=--y$	$x=4$

## Assignment Operators

Operator	Example	Same As	Result
=	$x=y$		$x=5$
+=	$x+=y$	$x=x+y$	$x=15$
-=	$x-=y$	$x=x-y$	$x=5$
*=	$x*=y$	$x=x*y$	$x=50$
/=	$x/=y$	$x=x/y$	$x=2$
%=	$x\%=y$	$x=x\%y$	$x=0$

### Penjumlahan variabel string( Huruf/Text )

Operators + tidak hanya digunakan untuk penjumlahan angka, tetapi juga dapat difungsikan untuk menggabungkan dua atau lebih variabel yang bernilai string / text.

```
Text1 = "Ryan";
Text2 = "Ganteng";
Text3 = Text1 + Text2;
```

Maka setelah syntax diatas dijalankan hasilnya adalah "RyanGanteng". Tetapi akan berbeda dengan syntax dibawah ini:

```
Text1 = "Ryan";
Text2 = "Ganteng";
Text3 = Text1 + " " + Text2;
```

Setelah dijalankan maka hasilnya adalah "Ryan Ganteng". Perbedaannya terlihat pada spasi antara text "Ryan" dan "Ganteng".

### Penjumlahan variabel number(Angka) dan string( Huruf/Text )

Pada kenyataannya nilai number ternyata juga dapat ditambahkan dengan string. Hasil dari penjumlahannya akan bernilai **String**. Ketika *number* dijumlahkan dengan *number*, maka hasilnya adalah penjumlahan yang sebenarnya. Akan tetapi akan berbeda dengan hasil penjumlahan antara *number* dengan *string* atau *string* dengan *string*, karena hasil penjumlahannya bernilai *string*, maka hasilnya adalah bukan penjumlahan yang sebenarnya, tetapi penggabungan antara kedua nilai tersebut.

Contoh :

`X = 5 + 5;`  
`Document.write(X)` } **Number + Number**

`X = "5" + "5";`  
`Document.write(X)` } **String + String**

`X = "5" + 5;`  
`Document.write(X)` } **String + Number**

`X = 5 + "5";`  
`Document.write(X)` } **Number + String**

Hasil :

**Number + Number** = 10  
**String + String** = 55  
**String + Number** = 55  
**Number + String** = 55

### 6.3. Scripting Comparison

Scripting Comparison berfungsi untuk membandingkan antar nilai dari suatu variabel.

Dibawah ini adalah operator yang ada di Scripting comparison:

Operator	Description	Example
<code>==</code>	is equal to	<code>x==8</code> is false
<code>===</code>	is exactly equal to (value and type)	<code>x===5</code> is true <code>x==="5"</code> is false
<code>!=</code>	is not equal	<code>x!=8</code> is true
<code>&gt;</code>	is greater than	<code>x&gt;8</code> is false
<code>&lt;</code>	is less than	<code>x&lt;8</code> is true
<code>&gt;=</code>	is greater than or equal to	<code>x&gt;=8</code> is false
<code>&lt;=</code>	is less than or equal to	<code>x&lt;=8</code> is true

**If (Umur < 18)**

**Document.write("Masih Muda");**

Dari contoh diatas, variabel umur dibandingkan dengan angka 18, yang artinya jika nilai dari variabel umur adalah kurang dari 18, maka hasilnya akan tampil " **Masih Muda**"

### Logical Operator

Logical operator digunakan untuk menentukan logika antara nilai dari suatu variable.

Operator	Description	Example
&&	and	(x < 10 && y > 1) is true
	or	(x==5    y==5) is false
!	not	!(x==y) is true

### Conditional operator

Dalam Scripting comparison juga berisi sebuah operator kondisional yang memberikan nilai ke variabel yang didasarkan pada beberapa kondisi.

**Syntax dasar:**

**NamaVariabel = (Kondisi)?nilai1:nilai2;**

**Contoh :**

**Status = (Sex = "Laki-laki")?"Tuan":"Nyonya";**

Jika variabel **Sex** berisi "**Laki-laki**" maka variabel **Status** akan bernilai "**Tuan**", jika variabel **Sex** berisi bukan "**Laki-laki**" maka variabel status akan bernilai "**Nyonya**".

### 6.4 Scripting Branching

Scripting Branching atau bisa juga disebut sebagai script percabangan ini digunakan untuk menguji sebuah kondisi dan kemudian mengeksekusi pernyataan tertentu bila kondisi tersebut terpenuhi, dan mengeksekusi pernyataan lain bila kondisi tersebut tidak terpenuhi.

Pada dasarnya dalam JavaScript terdapat dua macam pernyataan percabangan yaitu **if..else** dan **switch**.

### If ... Else

#### If (kondisi)

```
{
    //Pernyataan 1 dieksekusi
    //Bila kondisi terpenuhi
}
Else
{
    //Pernyataan 2 dieksekusi
    //Bila kondisi tidak terpenuhi
}
```

**Kondisi** adalah ekspresi JavaScript yang mana hasil evaluasinya memiliki nilai Boolean **true** atau **false**.

Untuk kasus yang melibatkan lebih banyak kondisi, maka kita dapat meletakkan pernyataan **if** lain setelah **else**.

#### if (kondisi1)

```
{
    //pernyataan1 dieksekusi
    //bila kondisi1 terpenuhi
}
else if (kondisi2)
{
    //pernyataan2 dieksekusi
    //bila kondisi1 tidak terpenuhi
}
else
{
    //pernyataan3 dieksekusi
    //bila kondisi2 tidak terpenuhi
}
```

### Contoh penulisan di dalam Javascript:

```
<SCRIPT language="JavaScript">
function tanyabilangan()
{
  var bil = parseFloat(document.fform.bilangan.value);
  var jenis = " ";
  if(isNaN(bil))
  {
    alert("Anda Belum memasukkan Bilangan");
  }
  else
  {
    if (bil > 0)
    {
      jenis = " Adalah bilangan Positif";
    }
    else if (bil < 0)
    {
      jenis = " Adalah bilangan Negatif";
    }
    else
    {
      jenis = " Adalah Nol";
    }
    alert (bil+" "+jenis);
  }
}
</SCRIPT>
```

### Hasil dari contoh diatas adalah :

- ✓ jika variabel bil tidak diisi, maka akan muncul alert “**Anda belum memasukkan bilangan**”
- ✓ Jika variabel bil bernilai lebih dari 0, maka variabel jenis akan bernilai “**Adalah bilangan positif**”
- ✓ Jika variabel bil bernilai kurang dari 0, maka variabel jenis akan bernilai “**Adalah bilangan negatif**”
- ✓ Dan jika kondisi bernilai selain kondisi – kondisi diatas, maka variabel jenis akan bernilai “**Adalah nol**”

## Switch

Selain menggunakan if..else, percabangan juga dapat ditangani dengan perintah switch. Dengan kata lain pernyataan switch digunakan untuk menyederhanakan pernyataan if..else yang terlalu banyak.

### Contoh penulisan di dalam Javascript:

```
<SCRIPT language="JavaScript">
function tanyabulan()
{
var bulan = parseFloat(document.fform.ibulan.value);
var namabulan=" ";
switch (bulan)
{
case 1 : namabulan = "Bulan ke 1 adalah = Januari"; break;
case 2 : namabulan = "Bulan ke 2 adalah = Februari"; break;
case 3 : namabulan = "Bulan ke 3 adalah = Maret"; break;
case 4 : namabulan = "Bulan ke 4 adalah = April"; break;
case 5 : namabulan = "Bulan ke 5 adalah = Mei"; break;
case 6 : namabulan = "Bulan ke 6 adalah = Juni"; break;
case 7 : namabulan = "Bulan ke 7 adalah = Juli"; break;
case 8 : namabulan = "Bulan ke 8 adalah = Agustus"; break;
case 9 : namabulan = "Bulan ke 9 adalah = September"; break;
case 10 : namabulan = "Bulan ke 10 adalah = Oktober"; break;
case 11 : namabulan = "Bulan ke 11 adalah = November"; break;
case 12 : namabulan = "Bulan ke 12 adalah = Desember"; break;
default : namabulan = "Anda salah mengisi";
}
alert(namabulan);
}
</SCRIPT>
```

## 6.5 Scripting PopUp

Javascript memiliki tiga macam popup, yaitu:

- ✓ **Alert box**
- ✓ **Confirm Box**
- ✓ **Prompt Box**

- **Alert Box**

**Alert box** adalah sebuah kotak peringatan / pesan (*Messagebox*). Dan ketika sebuah kotak peringatan / pesan itu muncul, maka pengguna harus klik "Ok" untuk melanjutkan proses.

**Syntax dasar:**

**Alert("Peringatan / Pesan yang ingin ditampilkan")**

**Contoh :**

```
<html>
<head>

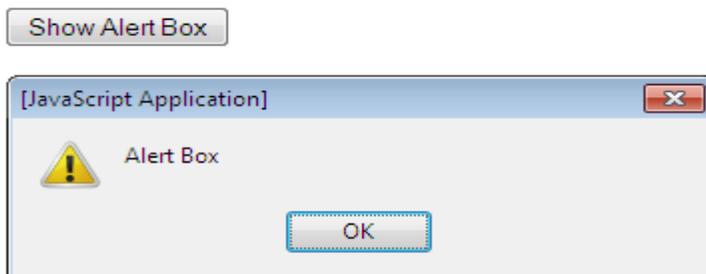
<script type="text/javascript">
function tampil_alert()
{
alert("Alert Box")
}
</script>

</head>
<body>

<input type="button" value="Show Alert Box" onclick="tampil_alert()" />

</body>
</html>
```

**Hasil:**



- **Confirm Box**

**Confirm Box** adalah sebuah kotak konfirmasi yang sering digunakan jika anda ingin pengguna memverifikasi atau menerima sesuatu. Ketika muncul kotak konfirmasi, user harus klik "**OK**" atau "**Batal**" untuk melanjutkan.

- ✓ Jika user mengklik "**OK**", maka Confirm Box mengembalikan nilai **true**.
- ✓ Jika user mengklik "**Batal**", maka Confirm Box mengembalikan nilai **false**.

Syntax dasar:

Confirm("Peringatan / Pesan yang ingin ditampilkan")

Contoh :

```
<html>
<head>

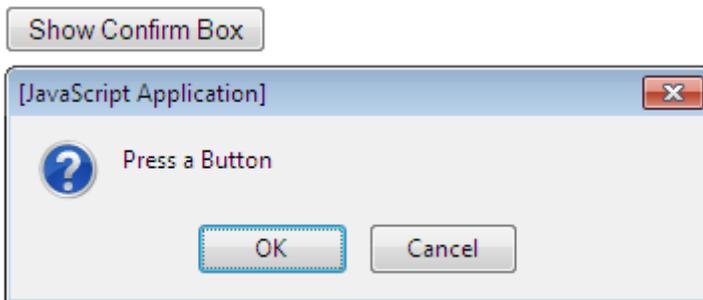
<script type="text/javascript">
function tampil_confirm()
{
var r = confirm("Press aButton");
if (r == true)
    {
    alert ("You pressed Ok !");
    }
else
    {
    alert("You pressed Cancel !");
    }
}
</script>

</head>
<body>

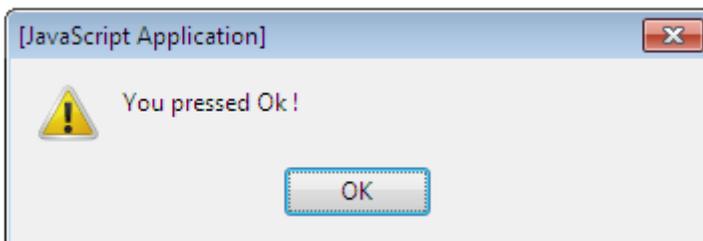
<input type="button" onclick="tampil_confirm()"
value="Show Confirm Box" />

</body>
</html>
```

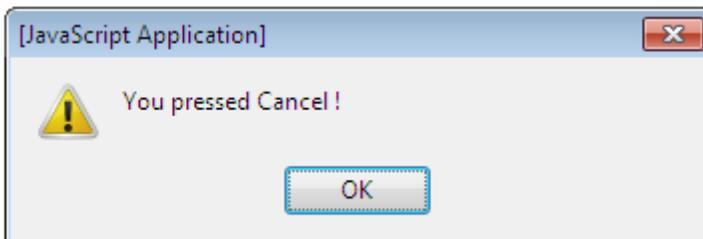
**Hasil:**



Jika button **ok** yang ditekan, maka hasilnya:



Jika button cancel yang ditekan, maka hasilnya:



- **Prompt Box**

**Prompt Box** adalah sebuah kotak prompt yang sering digunakan jika Anda ingin user untuk memasukkan nilai sebelum memasuki sebuah halaman. Ketika prompt box muncul, user harus klik "OK" atau "Cancel" setelah memasukkan nilai input.

- ✓ Jika user mengklik "OK", maka Prompt Box mengembalikan nilai **input**.
- ✓ Jika user mengklik "Batal", maka Prompt Box mengembalikan nilai **null**.

**Syntax dasar:**

```
Prompt("Teks","DefaultValue");
```

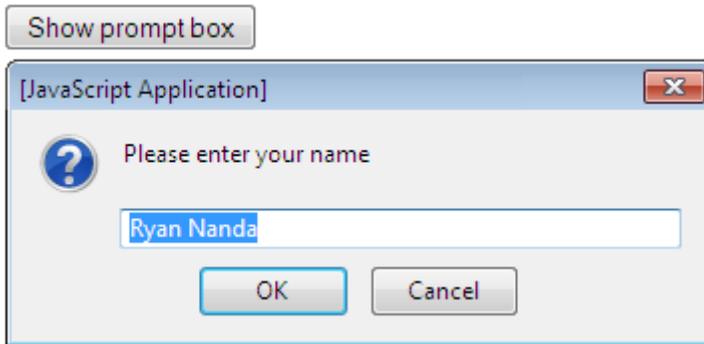
**Contoh:**

```
<html>
<head>
<script type="text/javascript">
function tampil_prompt()
{
var name=prompt("Please enter your name","Ryan Nanda");
if (name!=null && name!="")
{
document.write("Hello " + name + "! How are you today?");
}
}
</script>
</head>
<body>

<input type="button" onclick=" tampil_prompt()" value="Show
prompt box" />

</body>
</html>
```

**Hasil:**



Jika button ok yang ditekan, maka hasilnya:

**Hello Ryan Nanda! How are you today?**

### 6.6 Scripting Function

Scripting Function berfungsi untuk menjaga browser agar tidak menjalankan script pada saat pertama kali halaman web dijalankan. Maka cara yang tepat adalah anda menempatkan script anda ke dalam sebuah fungsi(*function*).

Anda dapat memanggil fungsi dari mana saja, entah fungsi itu ada di dalam halaman web anda ataupun fungsi tersebut berada pada halaman lain / file .js eksternal.

Fungsi dapat didefinisikan baik di <head> dan di bagian <body> dokumen. Namun, untuk memastikan bahwa fungsi dibaca / dibuka oleh browser sebelum fungsi itu disebut, lebih baik fungsi ditempatkan dalam bagian <head>.

**Syntax dasar:**

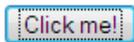
```
function namafunction(var1,var2,...,varX)
{
Script / Code
}
```

Contoh :

```
<html>
<head>
<script type="text/javascript">
function displaymessage()
{
alert("Hello World!");
}
</script>
</head>

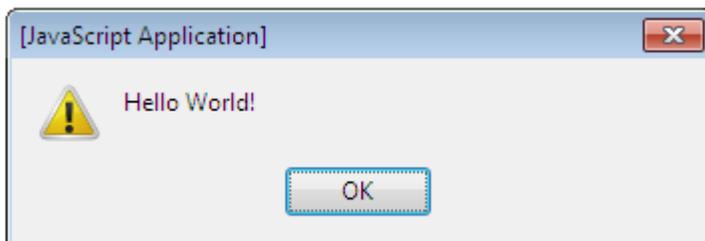
<body>
<form>
<input type="button" value="Click me!" onclick="displaymessage()" />
</form>
<p>Ketika menekan tombol diatas, function displaymessage akan
dijalankan.<br/>
Dan function tersebut akan menampilkan sebuah alert</p>
</body>
</html>
```

Hasil:



Ketika menekan tombol diatas, function displaymessage akan dijalankan.  
Dan function tersebut akan menampilkan sebuah alert

Dan ketika anda menekan tombol "Click me !" maka hasilnya:



### Function Return Statement

Function return statement ini digunakan ketika user ingin menentukan nilai yang akan dikembalikan dari function.

Function yang mengembalikan nilai harus menggunakan pernyataan **return**.

#### Contoh :

```
<html>
<head>
<script type="text/javascript">
function product(a,b)
{
return a*b;
}
</script>
</head>

<body>
<script type="text/javascript">
document.write(product(4,3));
</script>

</body>
</html>
```

#### Hasil:

12

Script pada bagian **body** mengembalikan nilai **4** pada parameter **a** dan mengembalikan nilai **3** pada parameter **b**, sehingga ketika di proses kembali pada script pada bagian **head**, hasilnya adalah **12**.

## Latihan Modul 6

# MODUL 7

## AND MORE INTERACTION

*Knowledge is of two kinds. We know a subject ourselves, or we know where we can find information upon it.*  
--Samuel Johnson--



### Tujuan

Praktikan dapat memahami dan membuat sebuah halaman web yang lebih kompleks

### Materi

**Scripting Loops, Events, Error Handling, Special Characters**

### Referensi

<a href="http://en.wikipedia.org/Javascript">http://en.wikipedia.org/Javascript</a>	■
<a href="http://www.w3schools.com/Js/js_loops_for.html">http://www.w3schools.com/Js/js_loops_for.html</a>	■
<a href="http://www.w3schools.com/Javascript/js_loops_while.html">http://www.w3schools.com/Javascript/js_loops_while.html</a>	■
<a href="http://www.w3schools.com/Js/js_special_characters.asp">http://www.w3schools.com/Js/js_special_characters.asp</a>	■

## 7.1. Scripting Loops

Scripting loops atau bisa disebut dengan perulangan ini pada dasarnya biasa digunakan untuk anda yang ingin menampilkan hasil yang sama secara berulang-ulang.

Di javascript terdapat dua jenis perulangan yang berbeda, yaitu:

✓ **For Loops**

Perulangan seluruh blok kode dengan kondisi tertentu.

Syntax dasar:

```
for (var=startvalue; var<=endvalue; var=var+increment)
{
  Kode yang akan dijalankan
}
```

Contoh:

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
for (i = 0; i <= 5; i++)
{
  document.write("Perulangan for " + i);
  document.write("<br />");
}
</script>
```

```
<p>Keterangan:</p>
```

```
<p>Perulangan diatas dimulai dari variabel i yang bernilai 0.</p>
```

```
<p>Selama <b>i</b> kurang dari, atau sama dengan 5, perulangan akan terus berjalan.</p>
```

```
<p>Nilai dari variabel <b>i</b> akan bertambah 1 setiap kali perulangan berjalan.</p>
```

```
</body>
```

```
</html>
```

### Hasil:

Perulangan for 0  
Perulangan for 1  
Perulangan for 2  
Perulangan for 3  
Perulangan for 4  
Perulangan for 5

Keterangan:

Perulangan diatas dimulai dari variabel i yang bernilai 0.

Selama i kurang dari, atau sama dengan 5, perulangan akan terus berjalan.

Nilai dari variabel i akan bertambah 1 setiap kali perulangan berjalan.

### ✓ Wile Loops

Perulangan seluruh blok kode ketika suatu kondisi yang sudah ditentukan bernilai benar.

Syntax dasar:

```
while (var<=endvalue)
{
Kode yang akan dijalankan
}
```

Contoh :

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

```
<script type="text/javascript">
i=0;
while (i<=5)
{
document.write("Perulangan while " + i);
document.write("<br />");
i++;
}
</script>
```

```

<p>Keterangan:</p>
<p><b>i</b> bernilai 0.</p>
<p>Nilai dari variabel <b>i</b> akan terus bertambah 1,</p>
<p>jika nilai dari variabel <b>i</b> kurang dari, atau sama
dengan 5</p>

</body>
</html>

```

**Hasil:**

```

Perulangan while 0
Perulangan while 1
Perulangan while 2
Perulangan while 3
Perulangan while 4
Perulangan while 5

```

**Keterangan:**

**i** bernilai 0.

Nilai dari variabel **i** akan terus bertambah 1,

jika nilai dari variabel **i** kurang dari, atau sama dengan 5

## 7.2 Scripting Events

Scripting events adalah pemicu tindakan. Pemicu yang dapat diatur kapan dan dimana akan digunakan sesuai dengan keadaan / kebutuhan. Sebagai contoh, kita dapat menggunakan events *onClick* pada button untuk menunjukkan bahwa suatu fungsi / proses akan dijalankan ketika menekan button tersebut.

Dibawah ini adalah nama events dan atributenya:

Nama Event	Attribute
On Load	Onload="Action"
On Unload	Onunload="Action"
On Focus	Onfocus="Action"
On Blur	Onblur="Action"

<b>On Change</b>	Onchange=" Action"
<b>On Submit</b>	Onsubmit=" Action"
<b>On Mouse Over</b>	Onmouseover=" Action"
<b>On Mouse Out</b>	Onmouseout=" Action"
<b>On Click</b>	OnClick=" Action"
<b>On Double Click</b>	Ondoubleclick=" Action"

### 7.3 Scripting Error Handling

Scripting error handling berfungsi untuk menjaga halaman web anda dari kerusakan. Sebagai contoh saat browsing halaman Web di internet, andamungkin pernah melihat kotak peringatan JavaScript yang memberitahukan bahwa terdapat runtime error dan menanyakan "Apakah Anda ingin debug?". Pesan error seperti ini mungkin berguna untuk pengembang tetapi tidak untuk pengguna. Bila pengguna melihat kesalahan, mereka sering meninggalkan halaman web tersebut.

Dalam Javascript terdapat dua macam error handling, yaitu:

✓ **Try ... Catch Statement**

**Try Catch** digunakan untuk menangkap kesalahan yang mungkin tidak anda sadari pada kode / script halaman web anda.

**Syntax dasar:**

```
Try  
{  
Kode yang dijalankan  
}  
Catch(err)  
{  
Penanganan error  
}
```

Contoh:

```
<html>
<head>

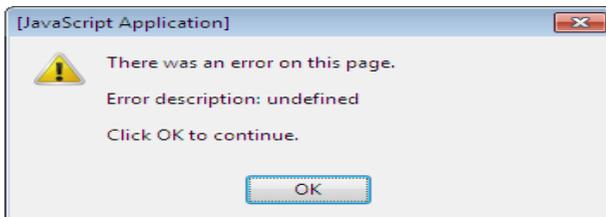
<script type="text/javascript">
var txt="";
function message()
{
    try
    {
        adddler("Welcome guest!");
    }
    catch(err)
    {
        txt="There was an error on this page.\n\n";
        txt+="Error description: " + err.description + "\n\n";
        txt+="Click OK to continue.\n\n";
        alert(txt);
    }
}
</script>

</head>

<body>
<input type="button" value="View message"
onclick="message()" />
</body>

</html>
```

Hasil:



Kode diatas menjalankan apa yang ada pada bagian *catch* karena terdapat kesalahan pada penulisan "*alert*"

✓ **Throw Statement**

**Throw statement** memungkinkan anda untuk membuat pengecualian. Jika Anda menggunakan pernyataan ini bersamaan dengan pernyataan Try ... Catch, Anda dapat mengontrol aliran program dan menghasilkan pesan kesalahan yang akurat.

**Syntax dasar:**

**throw(exception)**

**Contoh:**

**<html>**

**<body>**

**<script type="text/javascript">**

**var x=prompt("Enter a number between 0 and 10:","");**

**try**

**{**

**if(x>10)**

**{**

**throw "Err1";**

**}**

**else if(x<0)**

**{**

**throw "Err2";**

**}**

**else if(isNaN(x))**

**{**

**throw "Err3";**

**}**

**}**

**catch(er)**

**{**

**if(er=="Err1")**

**{**

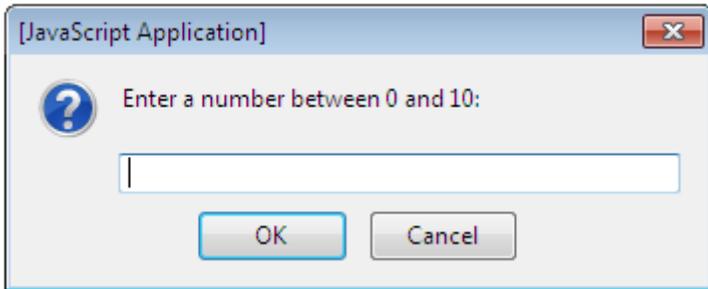
**alert("Error! The value is too high");**

**}**

**if(er=="Err2")**

```
{  
alert("Error! The value is too low");  
}  
if(er=="Err3")  
{  
alert("Error! The value is not a number");  
}  
}  
</script>  
  
</body>  
</html>
```

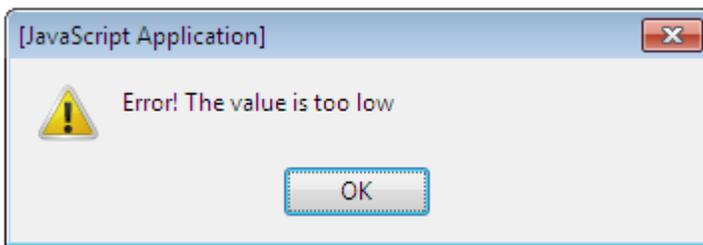
Hasil:



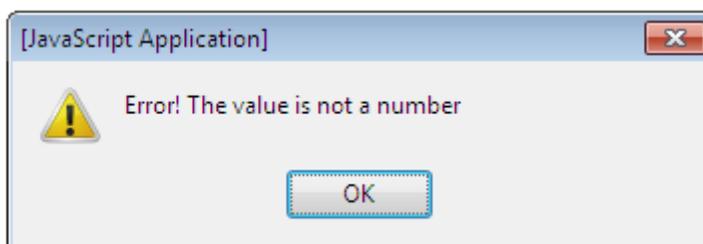
Jika anda mengisikan angka lebih dari 10 maka hasilnya



Jika anda mengisikan angka kurang dari 0 maka hasilnya



Jika anda mengisikan huruf maka hasilnya



### 7.4 Scripting Special Character

Di dalam Javascript anda dapat menambahkan karakter khusus ke dalam teks string dengan menggunakan tanda backslash.

**Syntax untuk karakter khusus:**

Code	Outputs
\'	single quote
\"	double quote
\&	ampersand
\\	backslash
\n	new line
\r	carriage return
\t	tab
\b	backspace
\f	form feed

**Contoh:**

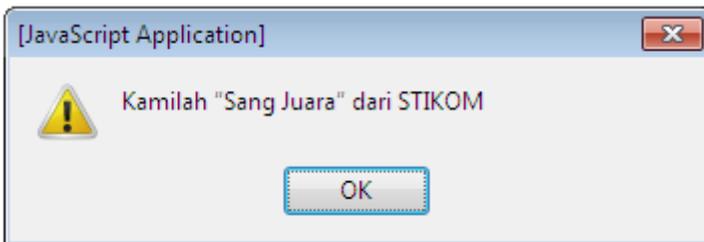
```
var txt ="Kamilah "Sang Juara" dari STIKOM"  
document.write(txt);
```

Di **Javascript**, suatu string dimulai dan diakhiri dengan tanda Kutip ganda. Dan berdasarkan contoh diatas, maka teks "Kamilah" akan dipotong.

Untuk mengatasi masalah ini, anda harus menempatkan backslash (\) di setiap sebelum tanda kutip ganda pada teks "Sang Juara". Seperti contoh dibawah ini:

```
var txt ="Kamilah \"Sang Juara\" dari STIKOM"  
document.write(txt);
```

Maka hasilnya:



## **Latihan Modul 7**

# MODUL 8

## WEB STYLISH

*Sepudar-pudar sebuah tulisan, akan lebih cepat pudar pemikiran cemerlang yang tidak pernah di tulis  
--Al-Ghazali--*



### Tujuan

Praktikan mengerti dan dapat mengkonversi styling dari HTML based ke CSS

### Materi

CSS Syntax, CSS Id And Class, CSS Formatting

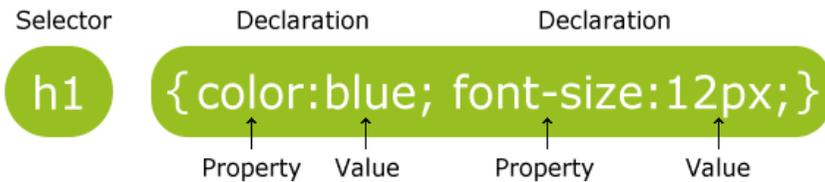
### Referensi

<a href="http://en.wikipedia.org/wiki/Standard_Access_Control_List">http://en.wikipedia.org/wiki/Standard_Access_Control_List</a>	■
Rafiudin, R. (2003). <i>Mengupas Tuntas Cisco Router</i> . Jakarta: PT. Elex Media Komputindo.	■
Wijaya, H. (2004). <i>Cisco Router, Edisi Baru untuk Mengambil Sertifikat CCNA (640-801)</i> . Jakarta : PT. Elex Media Komputindo.	■

CSS (Cascading Style Sheet) mulai diperkenalkan pada HTML versi 4. Manfaat dari CSS yaitu memudahkan kita dalam mengatur tampilan(style) dari suatu web page. Dengan CSS kita cukup membuat satu file CSS untuk mengatur semua tampilan dari halaman-halaman web yang kita miliki.

### 8.1. CSS Syntax

Sebuah CSS terdiri dari dua bagian utama: sebuah selector, dan satu atau lebih declaration.



#### Keterangan Gambar :

- **Selector**  
Selector adalah html elemen yang ingin kita ubah. Selector bisa juga berupa id/class.
- **Declaration**  
Declaration terdiri dari *property* dan *value*.
  - ✓ **Property**  
Property adalah atribut dari style yang ingin kita ubah.
  - ✓ **Value**  
Value adalah nilai dari attribute style yang dideklarasikan.

Penulisan syntax html akan sedikit berbeda dengan syntax CSS. Pada syntax html, **pendeclarasian attributnya selalu diberi tanda petik dua pada bagian value dan tanpa diakhiri titik koma**, sedangkan pendeclarasian attribute CSS **pada bagian value tidak akan diberikan tanda petik dua dan selalu diakhiri dengan titik koma**, dan **declaration groups akan diapit di dalam kurung kurawal**.

**Contoh Syntax:**

- Deklarasi attribute HTML  
`<Td align="Center" width="50%"></Td>`
- Deklarasi attribute CSS  
`P{color : red; Text-align : Center;}`

Agar Syntax CSS lebih mudah dibaca, alangkah baiknya jika penulisannya seperti dibawah ini:

```
P
{
color : red;
Text-align : Center;
}
```

**CSS Comment**

CSS comments digunakan untuk menjelaskan kode CSS yang sudah dibuat. Comment akan diabaikan(*tidak ditampilkan*) oleh browser.

Sebuah CSS comment dimulai dengan `"/**"`, dan diakhiri dengan `**/"`.

```
/* Ini adalah contoh CSS Comments */
P
{
/* Ini juga salah satu CSS Comments */
color : red;
Text-align : Center;
}
```

**8.2. CSS Id dan Class**

Untuk memudahkan kita dalam mengatur elemen-elemen dalam HTML, CSS memperbolehkan kita membuat selector sendiri yaitu **id** dan **class**.

**CSS Id** berfungsi untuk mengatur style dari sebuah elemen. Sebuah Id dimulai dengan tanda `"#"` (*Pagar / Kres*). Style di bawah ini akan diaplikasikan pada semua elemen yang memakai id `"Tulisan"`

### #Tulisan

```
{  
color : red;  
font-family : Arial;  
}
```

Cara pemanggilan Id diatas adalah sebagai berikut:

```
<p id="Tulisan">
```

CSS Class berfungsi untuk mengatur style dari beberapa elemen. Dengan class kita bisa mengatur style lebih dari satu elemen. Sebuah Class dimulai dengan tanda "." (Titik). Style di bawah ini akan diaplikasikan pada semua elemen yang memakai class "center"

### .Center

```
{  
Background-image : Url ('Namafile.jpg')  
Background-position : Center Center;  
}
```

Cara pemanggilan Class diatas adalah sebagai berikut:

```
<Body class="Center">
```

## 8.3. CSS Formatting

CSS formatting berfungsi untuk tata format penulisan syntax CSS seperti *property* dan *value* nya.

Dibawah ini adalah macam - macam property beserta value yang menyertai property tersebut:

## Styling Text

Property	Description	Values	CSS
<a href="#"><u>color</u></a>	Sets the color of a text	color	1
<a href="#"><u>direction</u></a>	Sets the text direction	ltr rtl	2
<a href="#"><u>line-height</u></a>	Sets the distance between lines	normal number length %	1
<a href="#"><u>letter-spacing</u></a>	Increase or decrease the space between characters	normal length	1
<a href="#"><u>text-align</u></a>	Aligns the text in an element	left right center justify	1
<a href="#"><u>text-decoration</u></a>	Adds decoration to text	none underline overline line-through blink	1
<a href="#"><u>text-indent</u></a>	Indents the first line of text in an element	length %	1
<a href="#"><u>text-shadow</u></a>		none color length	
<a href="#"><u>text-transform</u></a>	Controls the letters in an element	none capitalize uppercase lowercase	1
<a href="#"><u>unicode-bidi</u></a>		normal embed bidi-override	2
<a href="#"><u>vertical-align</u></a>	Sets the vertical alignment of an element	baseline sub super top text-top middle bottom text-bottom length %	1
<a href="#"><u>white-space</u></a>	Sets how white space inside an element is handled	normal pre nowrap	1
<a href="#"><u>word-spacing</u></a>	Increase or decrease the space between words	normal length	1

## Styling Background

Property	Description	Values	CSS
<a href="#"><u>background</u></a>	Sets all the background properties in one declaration	<i>background-color</i> <i>background-image</i> <i>background-repeat</i> <i>background-attachment</i> <i>background-position</i> inherit	1
<a href="#"><u>background-attachment</u></a>	Sets whether a background image is fixed or scrolls with the rest of the page	scroll fixed inherit	1
<a href="#"><u>background-color</u></a>	Sets the background color of an element	<i>color-rgb</i> <i>color-hex</i> <i>color-name</i> transparent inherit	1
<a href="#"><u>background-image</u></a>	Sets the background image for an element	url( <i>URL</i> ) none inherit	1
<a href="#"><u>background-position</u></a>	Sets the starting position of a background image	left top left center left bottom right top right center right bottom center top center center center bottom <i>x% y%</i> <i>xpos ypos</i> inherit	1
<a href="#"><u>background-repeat</u></a>	Sets if/how a background image will be repeated	repeat repeat-x repeat-y no-repeat inherit	1

## Styling Fonts

Property	Description	Values	CSS
<a href="#"><u>font</u></a>	Sets all the font properties in one declaration	<i>font-style</i> <i>font-variant</i> <i>font-weight</i> <i>font-size/line-height</i> <i>font-family</i> caption icon menu message-box small-caption status-bar inherit	1
<a href="#"><u>font-family</u></a>	Specifies the font family for text	<i>family-name</i> <i>generic-family</i> inherit	1
<a href="#"><u>font-size</u></a>	Specifies the font size of text	xx-small x-small small medium large x-large xx-large smaller larger length % inherit	1
<a href="#"><u>font-style</u></a>	Specifies the font style for text	normal italic oblique inherit	1
<a href="#"><u>font-variant</u></a>	Specifies whether or not a text should be displayed in a small-caps font	normal small-caps inherit	1
<a href="#"><u>font-weight</u></a>	Specifies the weight of a font	normal bold bolder lighter 100 200 300 400 500 600 700 800 900 inherit	1

## Styling List

Property	Description	Values	CSS
<a href="#"><u>list-style</u></a>	Sets all the properties for a list in one declaration	<i>list-style-type</i> <i>list-style-position</i> <i>list-style-image</i> inherit	1
<a href="#"><u>list-style-image</u></a>	Specifies an image as the list-item marker	URL none inherit	1
<a href="#"><u>list-style-position</u></a>	Specifies if the list-item markers should appear inside or outside the content flow	inside outside inherit	1
<a href="#"><u>list-style-type</u></a>	Specifies the type of list-item marker	none disc circle square decimal decimal-leading-zero armenian georgian lower-alpha upper-alpha lower-greek lower-latin upper-latin lower-roman upper-roman inherit	1

## Latihan Modul 8

Mengambil langkah pertama berarti siap mengambil risiko. Baik itu risiko kesuksesan maupun risiko kegagalan. Ini butuh keberanian yang besar, butuh komitmen yang kuat. Ini berarti mendobrak diri dari rasa takut, rasa bimbang, dan hanya fokus pada tujuan akhir.

*--Andrie wongso--*