

Mudah membuat WEB dengan penguasaan CSS dan HTML



Metode:

Baca

Coba

Lihat

Untung Rahardja
Augury El Rayeb
Asep Saefullah

KATA PENGANTAR

CSS yang merupakan singkatan dari *Cascading Style Sheet* adalah Salah satu fasilitas yang diberikan untuk pemrograman HTML sehingga pengaturan / disain tampilan suatu *web-page* dapat menjadi lebih baik. Hal ini dapat terjadi karena dengan CSS, user memiliki kemampuan untuk; mengatur posisi secara absolut, mengubah warna secara dinamis, merubah besar font secara dinamis, merubah margin secara dinamis dan lain sebagainya.

Terdapat beberapa metode dalam menerapkan CSS ke suatu web-page yaitu melalui sisipan – sisipan yang dilakukan pada dokumen HTML. Untuk dapat melakukan itu tentunya kita harus mengetahui tentang HTML, dengan demikian maka pada buku ini sebagai langkah awal akan dijelaskan tentang HTML.

Kalau kita berbicara mengenai HTML, maka tentunya kita juga akan berbicara tentang internet yang merupakan sumber dari asal – usul munculnya **World Wide Web**. HTML sendiri sebenarnya merupakan basis / dasar untuk membangun (membuat) **HomePage** / **Situs** (selanjutnya penulis akan menyebutnya **HomePage**) di internet.

Sebelum kita membicarakan lebih jauh mengenai CSS yang akan disisipkan pada HTML yang merupakan dasar untuk membangun HomePage (bahkan ada sebagian orang yang mengatakan bahwa HTML merupakan **language** / bahasa pemrogramannya HomePage). Ada baiknya kita mengenal apa itu HTML.

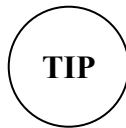
SIMBOL – SIMBOL DAN ATURAN PADA BUKU INI

Sebelum kita membahas lebih jauh tentang HTML dan HomePage ada baiknya kita lihat dulu simbol – simbol dan aturan yang ada pada buku ini

Symbol



Simbol ini menerangkan bahwa kalimat – kalimat pada area simbol ini merupakan informasi pendukung tulisan pada bagian saat itu.



Simbol ini menerangkan bahwa kalimat – kalimat pada area simbol ini merupakan TIP & TRICK dari penulis kepada pembaca, sehubungan dengan topik yang sedang dibahas pada bagian saat itu



Simbol ini menerangkan bahwa kalimat – kalimat pada area simbol ini merupakan Attention atau perhatian atau dapat dibilang peringatan kepada pembaca agar tidak melakukan kesalahan.

Aturan

- Contoh atau potongan HTML yang dapat anda coba ditulis dengan font dan ukuran seperti berikut :

```
<HTML>

<HEAD>
<TITLE>contoh syntax</TITLE>
<HEAD>

<BODY>
ini hanya contoh syntax
</BODY>

</HTML>
```

- Istilah baru / penting akan ditulis dengan bentuk tebal dan miring (bold & italic)

APA ITU HTML ?

HTML → **HyperText Markup Language**, adalah bahasa dari *world wide web*. Setiap kali anda mengakses internet atau lebih tepatnya homepage, anda sesungguhnya mengakses dokumen seseorang yang dibuat menggunakan format HTML. Setiap halaman / dokumen (selanjutnya penulis akan menyebutnya **dokumen**) ditulis dalam format HTML. Semua format dokumen, **hyperlink** yang dapat di-click, gambar grafis, dokumen multimedia, form yang dapat diisi, dan segala sesuatu dalam dokumen HomePage dibuat berdasarkan perintah / *syntax* (selanjutnya penulis akan menyebutnya **syntax**) HTML.

Selanjutnya agar kita dapat membuat suatu dokumen untuk HomePage maka kita perlu mengetahui syntax – syntax HTML. Lalu yang menjadi pertanyaan sekarang, bagaimana syntax – syntax HTML itu ?

Untuk itu anda tidak perlu khawatir syntax – syntax HTML justru akan kita pelajari pada bagian – bagian selanjutnya dalam buku ini, pada bagian ini yang perlu kita ketahui adalah bahwa setiap syntax HTML pasti memiliki penanda / **tag** (selanjutnya penulis menyebutnya **tag**). Tag HTML dapat diketahui dengan mudah yaitu : *dimulai dengan tanda < dan ditutup dengan tanda >* . Penggunaan tag ini juga merupakan sesuatu yang membedakan dokumen HTML dengan dokumen teks biasa.

Berikut ini beberapa contoh tag HTML :

- `<HTML>` dan `</HTML>`,
- `<HEAD>` dan `</HEAD>`,
- `<TITLE>` dan `</TITLE>`,
- `<BODY>` dan `</BODY>`,
- `<BR>`,
- `<P>` , dan lain sebagainya.

INF

- Tahukah anda bahwa HomePage terdiri dari beberapa halaman dan setiap halaman itu merupakan satu dokumen, maka berarti setiap Homepage terdiri dari beberapa dokumen.
 - Tahukah anda arti web itu sebenarnya → jaring laba – laba yang besar, web pada ilmu komputer atau internet khususnya adalah jaringan yang luas mencakup seluruh dunia.
 - *Hyperlink* → link / hubungan yang dibuat pada dokumen untuk menghubungkan dokumen yang sedang aktif / dibuka sekarang dengan dokumen lainnya.
 - Perlu diketahui bahwa penulisan syntax HTML ada yang perlu di tutup dengan tag penutupnya, yaitu tag yang memakai tanda / seperti pada <HTML> ditutup dengan </HTML> dan ada yang tidak perlu tag penutup seperti

-

EDITOR UNTUK MEMBUAT DOKUMEN HTML

Untuk membuat dokumen HTML (dokumen dengan syntax HTML) sebenarnya kita tidak memerlukan suatu editor khusus, yang kita perlukan hanyalah sebuah editor teks. Selanjutnya yang menjadi pertanyaan sekarang adalah bagaimana kita membedakan antara dokumen HTML dengan dokumen teks biasa ?

Dokumen HTML memiliki extension **.htm** atau **.html** jadi untuk itu kita perlu menyimpan dokumen yang kita buat dengan memberikan extension .htm atau .html pada nama file-nya. Contohnya misal kita akan membuat dokumen dengan nama index, maka simpan file dokumen kita dengan nama index.htm atau index.html.

TIP

Untuk menyimpan dokumen yang kita buat dengan menggunakan notepad yaitu :
 Click menu *File* → *Save*, kemudian pada kotak input *File Name* ketikkan namafile yang adan inginkan kemudian tambahkan .htm pada akhir namafile spt contoh berikut : *namafile.htm*
 Pada bagian *Save As Type* pilih → *All Files (*.*)* dan selanjutnya click tombol *Save*
 Jadi kesimpulannya untuk menyimpan dokumen lakukan cara, sebagai berikut :
 Click menu *File* → *Save*
FileName = namafile.htm atau namafile.html
Save As Type = *AllFiles (*.*)*

Selain menggunakan Notepad kita juga dapat membuat dokumen HTML dengan menggunakan editor khusus untuk HTML. Beberapa editor khusus untuk HTML diantaranya adalah :

- **HomeSite** dari Allaire
- **Macromedia Dreamweaver** dari Macromedia
- **Microsoft FrontPage** dari Microsoft
- dan masih banyak yang lainnya

Pada editor – editor khusus HTML seperti disebutkan diatas disediakan berbagai tools dan utility untuk membantu pemakai untuk mempermudah dalam membuat dokumen HTML.

Bila anda menggunakan editor seperti yang disebutkan di atas, penulis menyarankan agar pembaca yang baru belajar, dalam membuat dokumen HTML agar menulis syntax dan isi dokumen dilakukan dengan mengetik langsung (*jangan menggunakan tools atau utility yang disediakan editor*).

Hal ini berguna agar pembaca terbiasa dengan syntax – syntax dan aturan penulisan HTML, sehingga apabila ada kesalahan maka pembaca dapat menganalisa kemungkinan letak kesalahannya.

Demikianlah pengantar dari penulis, selanjutnya mari kita mulai mempelajari HTML dan Web Design ini secara perlahan dan penulis menyarankan agar pembaca mencoba setiap contoh yang ada di buku ini dan pahami apa yang terjadi dengan syntax – syntax yang ditulis pada contoh tersebut.

Tangerang, Januari 2009

Penulis

DOKUMEN HTML DASAR

TAG HTML DASAR YANG DIBUTUHKAN DALAM DOKUMEN

Berikut ini adalah tag dasar yang dibutuhkan dan harus ada dalam dokumen HTML :

- `<HTML>` dan `</HTML>`
- `<HEAD>` dan `</HEAD>`
- `<TITLE>` dan `</TITLE>`
- `<BODY>` dan `</BODY>`

Tag `<HTML>`

Setiap dokumen HTML dimulai dengan tag `<HTML>` dan diakhiri dengan tag `</HTML>`, yang menyatakan suatu dokumen sebagai dokumen HTML.

Contoh penulisan tag `<HTML>` dan merupakan contoh dokumen HTML paling sederhana :

```
<HTML>
Dokumen HTML ku yang pertama
</HTML>
```

Simpan dokumen anda dengan nama : **latihan1.htm**. Ingat selalu tambahkan dokumen

HTML anda dengan tag `<HTML>` pada awal dan akhir dokumen.

TIP

*Ingatlah untuk menutup tag yang digunakan dengan tanda **Slash (/)**, sesuai dengan tag tersebut.*

Seperti tag `<HTML>` di tutup dengan tag `</HTML>`.

Namun perlu diperhatikan bahwa ada beberapa tag yang tidak perlu ditutup dengan slash, seperti tag `<BR>`, `<P>` dan lain sebagainya. (tag ini akan kita pelajari nanti....)

Tag `<HEAD>` dan `<BODY>`

Dokumen HTML terdiri dari dua bagian, yaitu head (bagian kepala) dan body (bagian tubuh). Secara umum **head** berisi *informasi umum mengenai dokumen*, sedangkan **body** *biasanya merupakan isi dokumen yang sebenarnya*.

Contoh penulisan tag `<HEAD>` dan `<BODY>` (*coba dan lihat hasilnya*):

```

<HTML>
<HEAD>
... teks ini ditulis di area atau bagian head
</HEAD>
<BODY>
... teks Dokumen pertamaku ini ditulis di area body
</BODY>
</HTML>

```



Ingat area head (area yang dibatasi oleh tag <HEAD> dan </HEAD>) sebaiknya diisi dengan informasi umum dokumen saja.

Contoh diatas hanya untuk menunjukkan area dari kedua bagian dokumen yaitu area head dan area body, jadi sebaiknya untuk selanjutnya area head jangan diisi dengan teks yang akan ditampilkan pada HomePage. Karena seharusnya hanya berisi informasi umum dokumen saja.

Tag <TITLE>

Tag ini akan mengakibatkan ditampilkannya judul dari dokumen pada TitleBar browser.

tag <TITLE> ditulis di area head.



- Browser adalah aplikasi yang digunakan untuk melakukan browsing (pencarian HomePage) pada internet
 - Browser yang umum dipakai oleh pemakai internet saat ini adalah Internet Explorer dan Netscape namun masih ada browser lain yang juga bisa dibilang bai seperti opera, lynx (browser text mode) dan lain sebagainya.
-

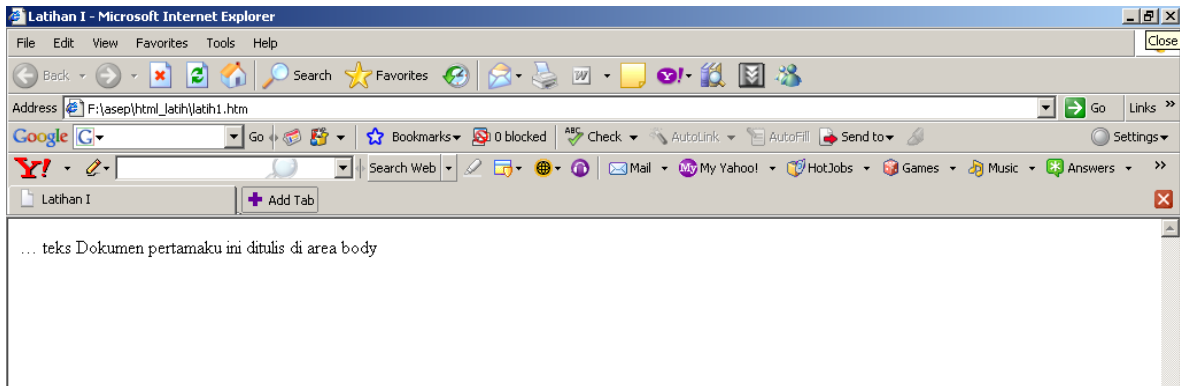
Contoh penulisan :

```

<HTML>
<HEAD>
    <TITLE>Latihan I</TITLE>
</HEAD>
<BODY>
... teks Dokumen pertamaku ini ditulis di area body
</BODY>
</HTML>

```

Coba lihat hasil dari contoh di atas apakah tulisan Latihan I ditampilkan pada browser ? Jawabannya adalah tidak, melainkan ditampilkan pada TitleBar Browser (*coba perhatikan TitleBar browser anda*).



Atribut dasar Tag <BODY>

Tag <body> sebenarnya memiliki atribut yang berguna untuk mengatur body (tubuh) dari suatu dokumen HTML. Dengan memanfaatkan atribut yang ada pada body kita dapat memberikan warna **background** (warna latar belakang), mengatur **margin** dan lain sebagainya. Sebagai permulaan disini akan diberikan beberapa atribut dasar yang umum dipakai pada dokumen HTML.

Atribut dasar tag <BODY>, diantaranya :

- **bgscolor** → warna background dokumen.
- **leftmargin** → ukuran margin kiri dokumen.
- **rightmargin** → ukuran margin kanan dokumen.
- **topmargin** → ukuran margin atas dokumen.
- **bottommargin** → ukuran margin bawah dokumen.
- **link, vlink, alink** → warna link, aktif link dan link yang sudah dikunjungi pada dokumen.

INF

- atribut adalah suatu pendukung yang berguna untuk mengatur perilaku atau tampilan suatu tag. Dengan memanfaatkan atribut pada suatu tag kita dapat mengatur perilaku atau tampilan dokumen yang disebabkan oleh tag yang bersangkutan.
- Tiap – tiap tag HTML memiliki atribut yang berbeda dan ada pula tag HTML yang tidak memiliki atribut.

Atribut bgcolor

Berikut ini aturan nomer hexa (*ingat kita harus memberikan tanda # sebelum angka hexadesimal tersebut*) : **#rrggbb**

Maksud kombinasi **#rrggbb**

rr → untuk nilai warna merah, rr diganti dengan angka 00 – FF.

gg → untuk nilai warna hijau, gg diganti dengan angka 00 – FF

bb → untuk nilai warna biru, bb diganti dengan angka 00 – FF

Misal untuk memberi warna biru pada latar belakang dokumen HTML maka kita cukup mengisikan **bgcolor="blue"** pada tag **<BODY>** seperti berikut :

```
<BODY bgcolor="blue">, atau
<BODY bgcolor="#0000FF">
```

Atribut leftmargin, rightmargin, topmargin dan bottommargin

Nilai yang diberikan pada margin ini dalam satuan pixel. Sehingga hasilnya akan sesuai dengan setting resolusi pada monitor yang membuka /melihat dokumen.

Conoth penulisan:

```
<BODY leftmargin="25" rightmargin="25" topmargin="20"
bottommargin="30">
```

Atribut link, alink, dan vlink

Memberikan setting warna terhadap link yang kita buat pada dokumen.

- **link** memberikan setting warna terhadap seluruh link yang ada pada dokumen
- **vlink** memberikan setting warna terhadap seluruh **visited link** yang ada pada dokumen
- **alink** memberikan setting warna terhadap seluruh **active link** yang ada pada dokumen.

Untuk pemberian nilai warna memakai cara yang sama dengan cara pada bgcolor.



- tahukah anda bahwa nilai hexadesimal hanya memiliki nilai dari 0 – 15, dan angka yang dikenal hanya 0 – 9, sedangkan untuk nilai 10 – 15 diwakili oleh huruf A – F (A, B, C, D, E, F).

- Standard resolusi display umum (yang sering dipakai sebagai acuan/patokan) dalam membuat dokumen adalah 800 x 600.
 - *vlink* (Visited link) adalah link yang pernah dikunjungi / dibuka oleh pengunjung HomePage / dokumen
 - *alink* (active link) adalah link yang sedang aktif saat ini atau link yang sedang terpilih oleh kursor saat ini
-

Contoh penulisan :

```
<HTML>
<HEAD>
  <TITLE>Latihan II</TITLE>
</HEAD>
<BODY bgcolor="blue" link="fuchsia" alink="white"
vlink="silver" leftmargin="25" rightmargin="25"
topmargin="20" bottommargin="30">

... teks Dokumen pertamaku ini ditulis di area body<br>
<a href="latihan1.htm">link ke dokumen latihan I</a><br>
<a href="">ini link 2</a>

</BODY>
</HTML>
```

1. Simpan hasil perubahan itu dengan nama **latihan2.htm**.
2. Jalankan dokumen **latihan2.htm** ini pada browser dan perhatikan TitleBar browser.
3. Perhatikan isi dokumen pada browser.
4. Click pada tulisan **link ke dokumen latihan I**, maka kita akan dihubungkan / berpindah ke dokumen latihan I.
5. Tekan **tombol back** pada browser anda untuk kembali ke dokumen latihan II.

Untuk melihat dampak dari link, alink dan vlink coba pindah – pindahkan dengan cara menekan tombol tab pada keyboard, perhatikan perubahan warna pada link !



FORMAT TAMPILAN SEDERHANA

TAG UNTUK HEADLINE

Untuk membuat teks **Headline** (judul) kita harus melingkupi teks tersebut dengan tag pembuka dan tag penutup headline. Tag Headline ditulis dengan aturan sebagai berikut : tag pembuka **<Hx>** dan tag penutup **</Hx>**, perhatian x diganti dengan angka contoh : **<H1>** atau **<H2>** atau **H3>** dst

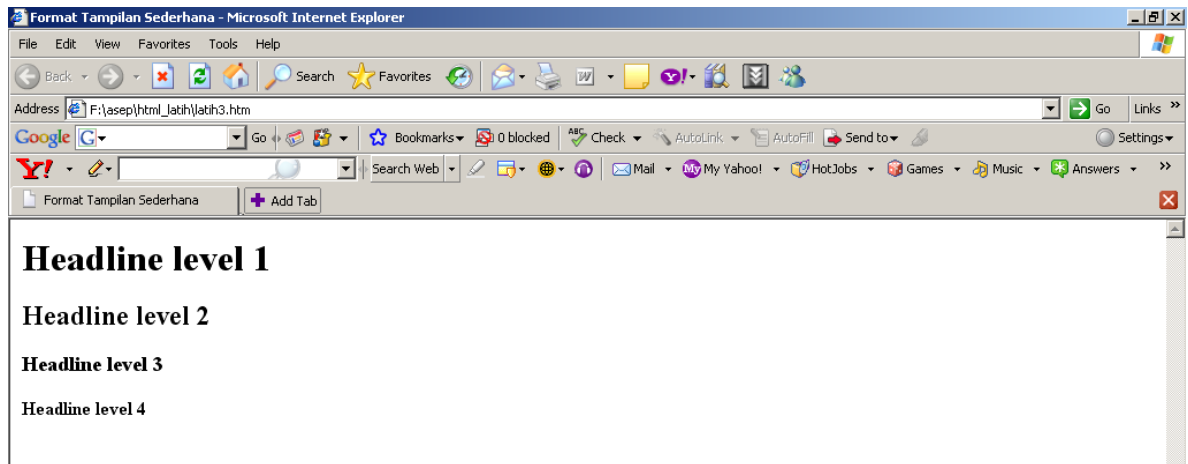
Angka tersebut menentukan besar kecilnya huruf teks judul yang berada diantara tag **<Hx>** dan **</Hx>**.

Semakin besar angka tersebut semakin kecil huruf teks judul.

Contoh penulisan tag **<Hx>** simpan dengan nama **latihan3.htm** :

```
<HTML>
<HEAD>
  <TITLE>Format Tampilan Sederhana</TITLE>
</HEAD>
<BODY>
  <H1>Headline level 1</H1>
  <H2>Headline level 2</H2>
  <H3>Headline level 3</H3>
  <H4>Headline level 4</H4>
</BODY>
</HTML>
```

Cobalah contoh di atas dan perhatikan bahwa huruf pada tulisan semakin kebawah semakin kecil. Coba simpulkan berarti (**simpulkan sendiri**).



TAG STYLE SEDERHANA

Disini akan dibahas beberapa **tag style**, tag - tag ini digunakan untuk mengatur format teks baik itu untuk membuat teks tebal, miring atau tulisan seperti bentuk kode program (*teks seperti bentuk listing program atau kalau dalam font yaitu courier new*)

Perhatikan tabel tag berikut :

FUNGSI	TAG LOGIKAL	TAG FISIKAL
Untuk membuat teks tebal		
Untuk membuat teks miring		<I>
Untuk huruf fixed (font jenis fixed)	<CODE>	<TT>

Seperti tag – tag yang lalu, tag ini juga harus ditutup dengan tag penutupnya masing – masing.

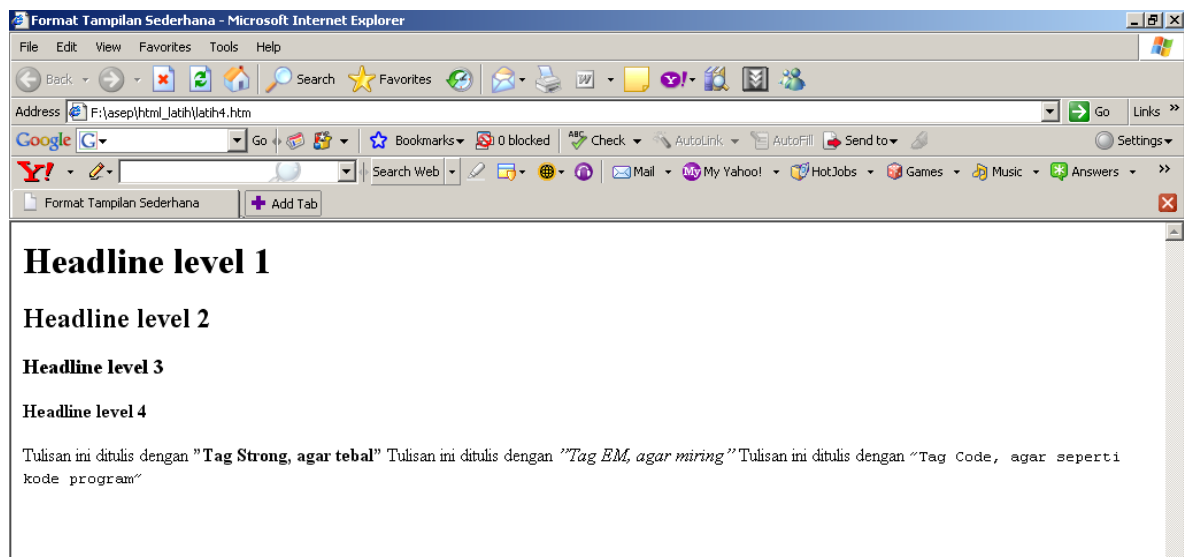


- Tag Fisikal → tampilan akan disesuaikan dengan kemampuan resolusi cetak dari printer pada komputer yang sedang melakukan browsing dokumen kita.
- Tag Logikal → tampilan akan dibuat sebaik mungkin tanpa kemampuan resolusi cetak dari printer pada komputer yang sedang melakukan browsing dokumen kita.
- Penulis menyarankan untuk menggunakan tag logikal.

Buka dan edit file latihan3.htm yang tadi kita buat. Ubahlah atau tambahkanlah pada dokumen tersebut sehingga menjadi seperti berikut ini :

```
<HTML>
<HEAD>
  <TITLE>Format Tampilan Sederhana</TITLE>
</HEAD>
<BODY>
  <H1>Headline level 1</H1>
  <H2>Headline level 2</H2>
  <H3>Headline level 3</H3>
  <H4>Headline level 4</H4>
  Tulisan ini ditulis dengan <STRONG>"Tag Strong, agar
tebal"</STRONG>
  Tulisan ini ditulis dengan <EM>"Tag EM, agar miring"</EM>
  Tulisan ini ditulis dengan <CODE>"Tag Code, agar seperti kode
program"</CODE>
</BODY>
</HTML>
```

Simpan file tersebut dan cobalah, perhatikan bahwa huruf pada tulisan semakin kebawah semakin kecil.



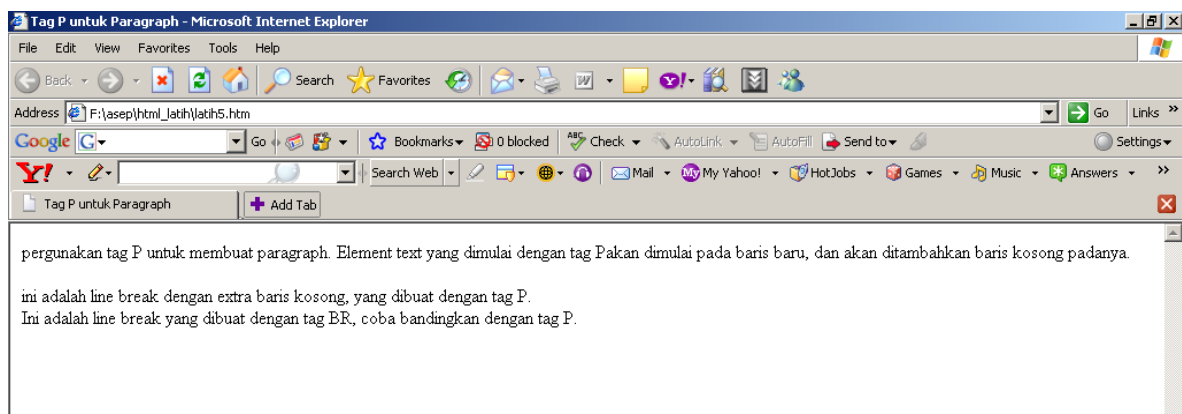
FORMAT ELEMENT TEXT PER BLOK

<P> TAG UNTUK PARAGRAPH

<P> digunakan untuk menampilkan satu blok text dalam bentuk paragraph, dimana blok text tersebut diawali dengan <P> dan ditutup dengan </P>. Penggunaan tag </P> sebagai penutup akan memastikan bahwa paragraph akan disertai dengan extra space (extra 1 baris kosong).

Contoh penulisan:

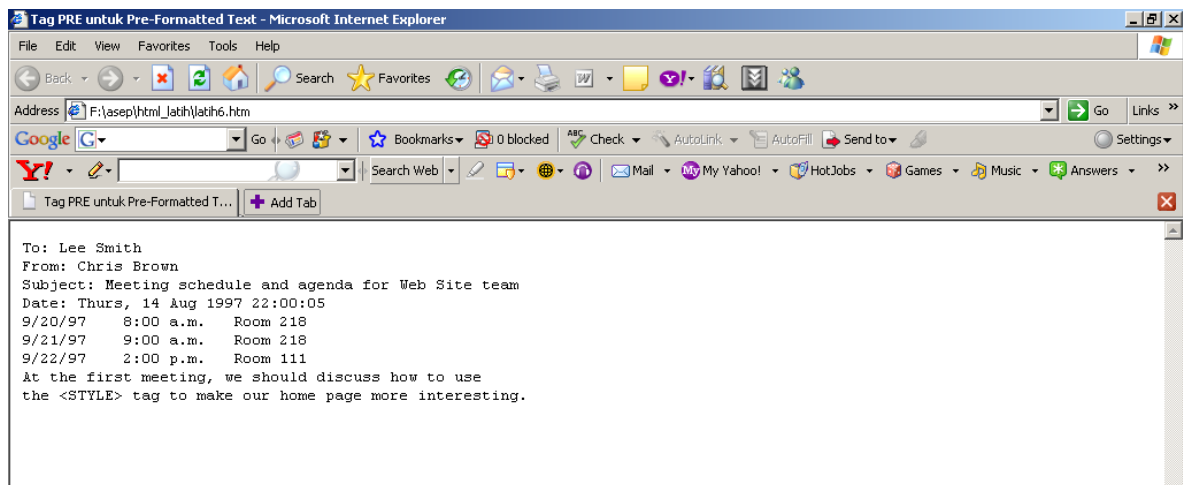
```
<HTML>
<HEAD>
  <TITLE>Tag P untuk Paragraph</TITLE>
</HEAD>
<BODY>
  <P>pergunakan tag P untuk membuat paragraph. Element text
  yang dimulai dengan tag P akan dimulai pada baris baru, dan
  akan ditambahkan baris kosong padanya.</P><P>ini adalah line
  break dengan extra baris kosong, yang dibuat dengan tag
  P.<BR>Ini adalah line break yang dibuat dengan tag BR, coba
  bandingkan dengan tag P.
</BODY>
</HTML>
```



<PRE> TAG PRE- FORMATTED TEXT

<PRE> merupakan tag yang dapat digunakan untuk membuat satu blok tulisan yang tata letaknya dapat kita atur sesuai keinginan kita, sesuai dengan tampilan pada saat pembuatan. Blok text tersebut diawali dengan <PRE> dan ditutup dengan </PRE>.

```
<HTML>
<HEAD>
    <TITLE>Tag PRE untuk Pre-Formatted Text</TITLE>
</HEAD>
<BODY>
<PRE>
To: Lee Smith
From: Chris Brown
Subject: Meeting schedule and agenda for Web Site team
Date: Thurs, 14 Aug 1997 22:00:05
9/20/97    8:00 a.m.    Room 218
9/21/97    9:00 a.m.    Room 218
9/22/97    2:00 p.m.    Room 111
At the first meeting, we should discuss how to use
the &lt;STYLE&gt; tag to make our home page more interesting.
</PRE>
</BODY>
</HTML>
```

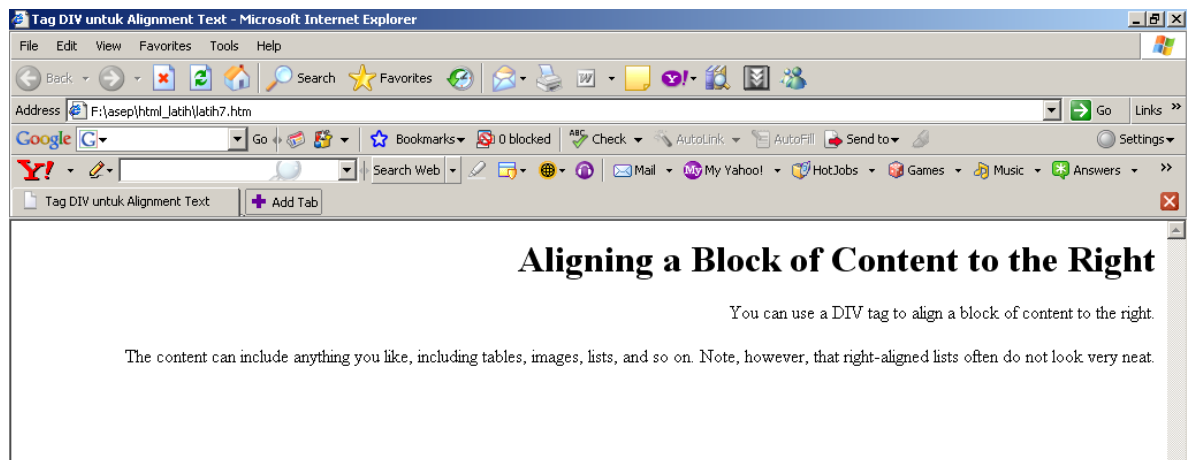


<DIV> TAG UNTUK MEMFORMAT ALIGNMENT SATU BLOK TEXT

Tag <DIV> dapat digunakan untuk mengatur alignment satu blok text, apakah akan kita atur rata kiri atau rata kanan atau tengah. Blok text tersebut diawali dengan <DIV> dan

ditutup dengan `</DIV>`. Sedangkan untuk pengaturan alignment diatur pada parameter tag `<DIV>` tersebut. Yaitu dengan syntax : `<DIV ALIGN="LEFT" | "CENTER" | "RIGHT">`

```
<HTML>
<HEAD>
    <TITLE>Tag DIV untuk Alignment Text</TITLE>
</HEAD>
<BODY>
    <DIV ALIGN=RIGHT>
        <H1>Aligning a Block of Content to the Right</H1>
        <P>You can use a DIV tag to align a block of content to the
        right.</P>
        <P>The content can include anything you like, including tables,
        images, lists, and so on. Note, however, that right-aligned
        lists
        often do not look very neat.</P>
    </DIV>
</BODY>
</HTML>
```



TAG – TAG UNTUK MEMFORMAT TEXT

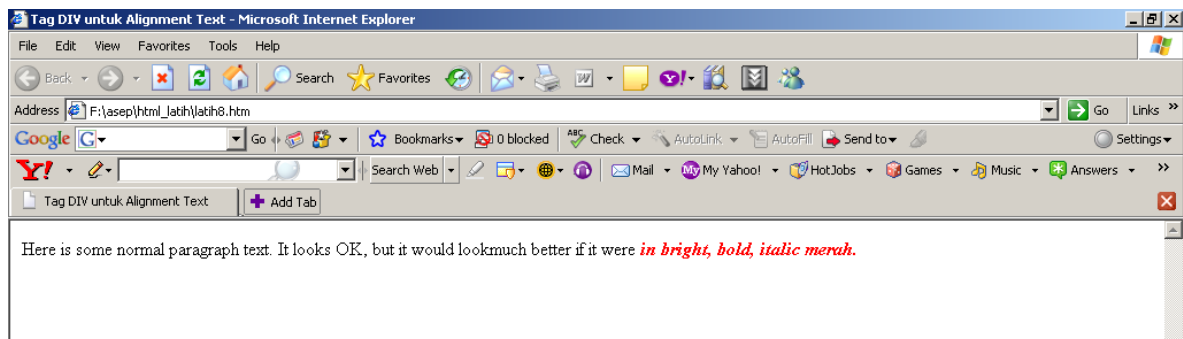
 TAG UNTUK MEMFORMAT TEXT DALAM BARIS

tag digunakan untuk memformat sejumlah text pada suatu baris kalimat tanpa mengganggu text lain dalam baris yang sama.

Syntax:

```
<SPAN STYLE="color:warna; font-weight:bold|Regular; font-style:italic|Normal">
```

```
<HTML>
<HEAD>
  <TITLE>Tag DIV untuk Alignment Text</TITLE>
</HEAD>
<BODY>
  <P>Here is some normal paragraph text. It looks OK,
  but it would lookmuch better if it were
  <SPAN STYLE="color:red; font-weight:bold; font-style:italic">
    in bright, bold, italic merah. </SPAN>
</BODY>
</HTML>
```



<CENTER> TAG UNTUK CENTER TEXT

<CENTER> akan menyebabkan text atau tulisan yang diapit oleh tag <CENTER> dan </CENTER> akan diposisikan center atau ditengah – tengah secara horisontal.

```
<HTML>
<HEAD>
  <TITLE>Tag DIV untuk Alignment Text</TITLE>
</HEAD>
<BODY>
  <CENTER><H1>Perguruan Tinggi Raharja</H1></CENTER>Selamat
    datang di Jurusan Sistem Komputer CCIT
</BODY>
</HTML>
```

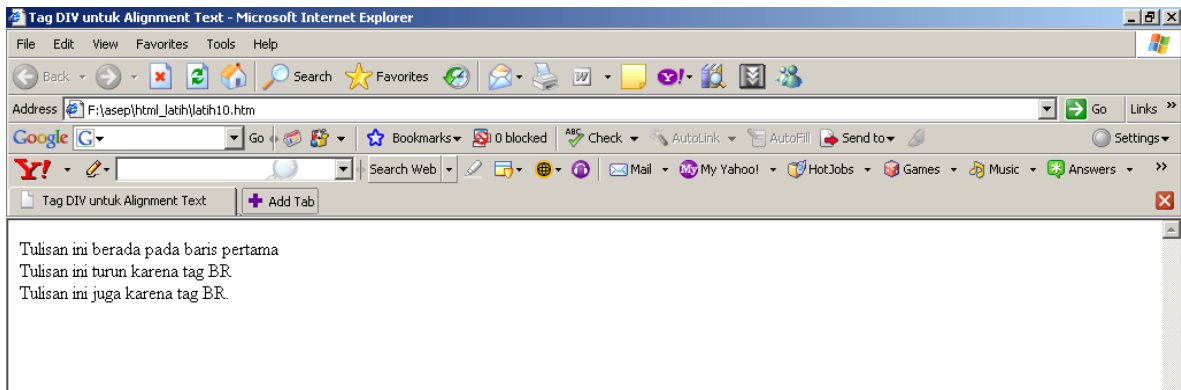


 TAG UNTUK LINE BREAK

 akan menyebabkan text atau tulisan yang ada di sebelah kanannya akan turun kebawah satu baris (*Line Break*). Tag
 tidak memiliki tag penutup.

Perhatikan contoh berikut:

```
<HTML>
<HEAD>
  <TITLE>Tag DIV untuk Alignment Text</TITLE>
</HEAD>
<BODY>
  Tulisan ini berada pada baris pertama <BR>Tulisan ini turun
  karena tag BR <BR>Tulisan ini juga karena tag BR.
</BODY>
</HTML>
```

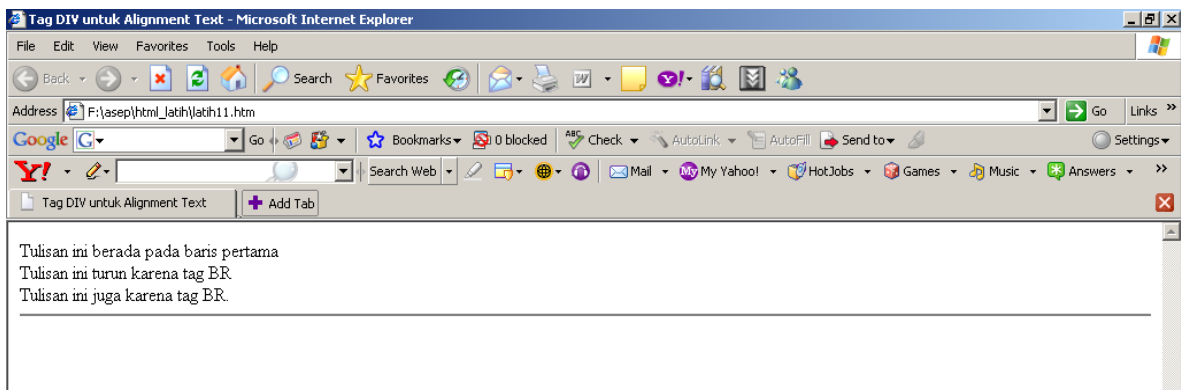


<HR> TAG UNTUK MEMBUAT GARIS PEMISAH HORIZONTAL

Tag <HR> digunakan untuk membuat garis pemisah antar baris atau untuk membuat suatu garis horizontal sepanjang lebar halaman homepage. Tag <HR> tidak memiliki tag penutup.

Syntax

```
<HR
  ALIGN="CENTER" | "LEFT" | "RIGHT"
  NOSHADE
  SIZE="thickness"
  WIDTH="width"
>
```



ALIGN untuk menentukan apakah rata kiri atau tengah atau kanan

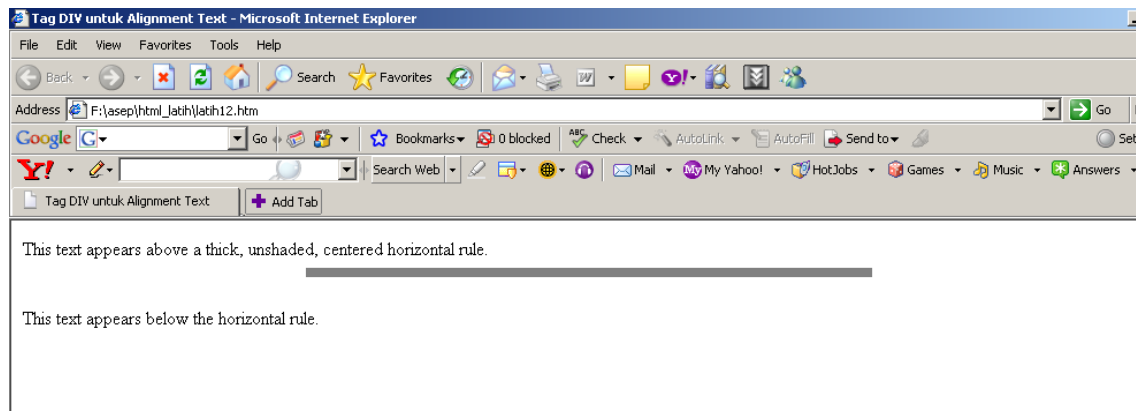
NOSHADE Jika ditulis maka garis dibuat tanpa bayangan, jika tidak maka garisnya memiliki bayangan

SIZE untuk menentukan ketebalan garis (dalam pixel)

WIDTH Untuk menentukan lebar / panjang garis. (dalam persen %)

Perhatikan contoh berikut:

```
<HTML>
<HEAD>
  <TITLE>Tag DIV untuk Alignment Text</TITLE>
</HEAD>
<BODY>
  <P>This text appears above a thick, unshaded, centered
  horizontal rule.
  <HR NOSHADE ALIGN="CENTER" WIDTH="50%" SIZE="8">
  <P>This text appears below the horizontal rule.
</BODY>
</HTML>
```



ANCHOR DAN LINK

Tag <A> memungkinkan kita untuk mendefinisikan anchors dan links (anchor = poin atau tempat dimana kita dapat mengaksesnya sewaktu – waktu, link = pemanggil anchors atau dokumen .html lain, sehingga kita melompat ke anchor tersebut jika link kita click).

<A> sebagai anchor

Sebuah anchor mengidentifikasi suatu tempat dalam suatu dokumen HTML. Untuk mengidentifikasi bahwa tag <A> digunakan sebagai anchor, kita harus mengisi parameter **NAME** pada tag <A>.

Syntax:

```
<A  
  NAME="anchorName"  
>  
...  
</A>
```

<A> sebagai link

Untuk mendefinisikan suatu link, penggunaan tag <A> dengan mengisi parameter **HREF** pada tag <A>, untuk memberitahukan tujuan dari sebuah link jika user meng-Click isi atau text yang berada diantara tag <A HREF> dan .

Nilai yang diisikan pada parameter **HREF** adalah berupa URL (Uniform Resource Locators). Jika kita ingin link membuka dokumen baru, maka nilai **HREF** harus merupakan nama dokumen yang dituju. Jika kita ingin link mengarahkan tampilan ke suatu tempat pada dokumen yang sama, maka nilai **HREF** harus merupakan nama dari *anchor* tujuan yang diawali dengan simbol # (contoh: #bagian1).

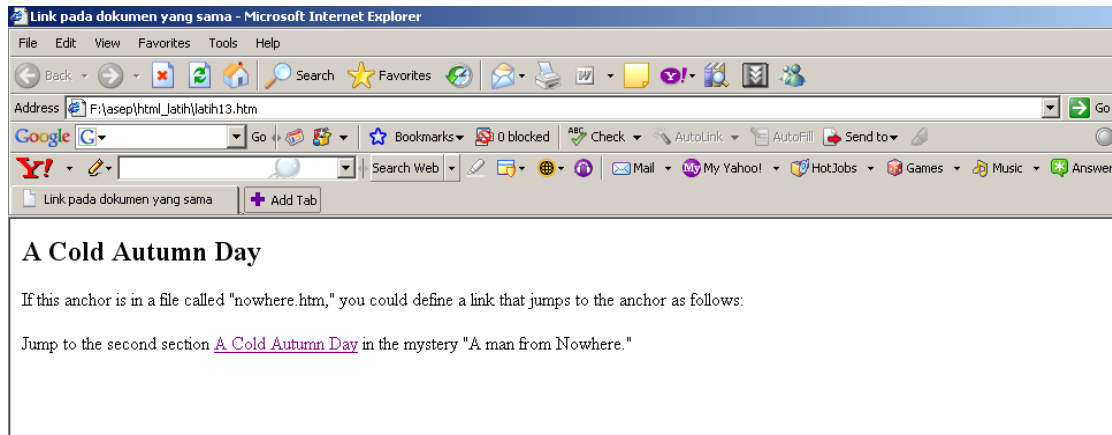
Syntax:

```
<A  
  HREF="location atau nama dokumen"
```

```
>
...
</A>
```

Perhatikan contoh berikut:

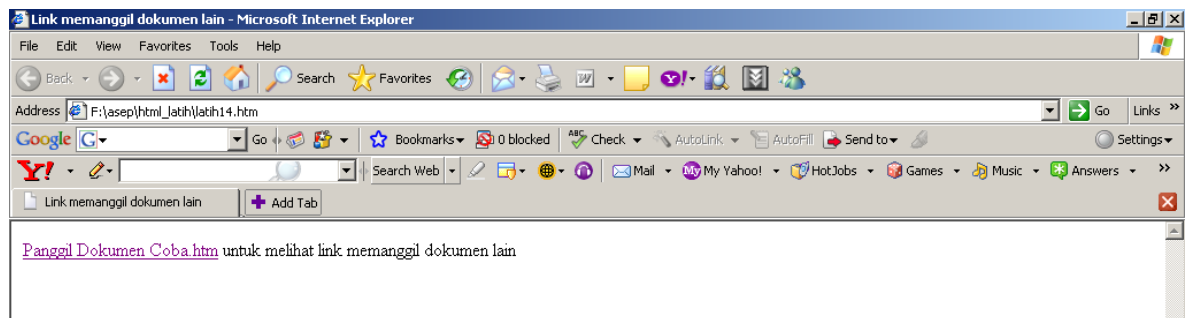
```
<HTML>
<HEAD>
  <TITLE>Link pada dokumen yang sama</TITLE>
</HEAD>
<BODY>
  <A NAME=section2>
    <H2>A Cold Autumn Day</H2></A>
    If this anchor is in a file called "nowhere.htm," you could
    define a link that jumps to the anchor as follows:
    <P>Jump to the second section
    <A HREF="#section2">A Cold Autumn Day</A> in the mystery "A
    man from Nowhere."
  </BODY>
</HTML>
```



Untuk mencoba link ke dokumen lain buat file HTML dengan nama **coba.htm**

Kemudian buat file HTM berikut:


```
<HTML>
<HEAD>
  <TITLE>Link memanggil dokumen lain</TITLE>
</HEAD>
<BODY>
  <A HREF="coba.htm">Panggil Dokumen Coba.htm</A> untuk melihat
  link memanggil dokumen lain
</BODY>
</HTML>
```



IMAGE

 TAG

Tag digunakan untuk menentukan file image yang akan ditampilkan dalam dokumen HTML.

File format image yang umum, digunakan adalah:

- GIF (Graphics Interchange Format)
- JPEG (Joint Photographic Experts Group)

Syntax

```
<IMG  
  SRC="location"  
  ALT="alterntiveText"  
  ALIGN="alignment"  
  BORDER="borderWidth"  
  HEIGHT="height"  
  WIDTH="width"  
>
```

SRC , Menentukan alamat dan nama file image.

LOWSRC ,Menentukan alamat dan nama file image resolusi rendah, yang akan ditampilkan saat menunggu loading image.

ALT , text alternative yang akan ditampilkan jika browser tidak mendukung tag IMG atau user melakukan browsing dalam text mode.

ALIGN Menentukan posisi gambar.

- LEFT Posisi kiri.
- RIGHT Posisi kanan.
- TOP Posisi atas.
- BOTTOM Posisi bawah.

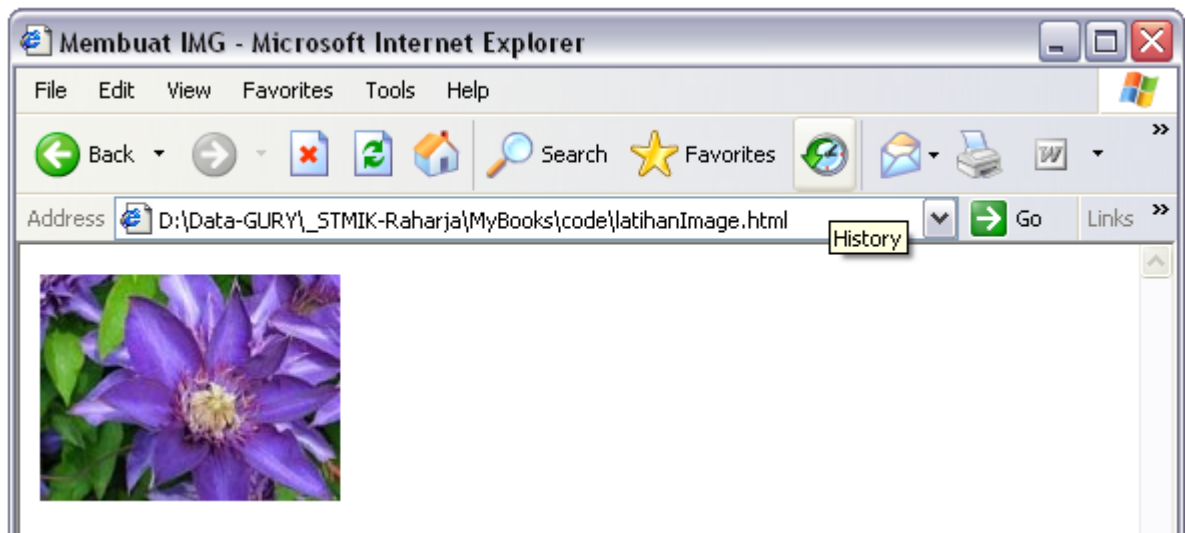
BORDER , Menentukan tebal border disekitar image (diisi dengan nilai integer)

HEIGHT , Menentukan tinggi dari image, diisi dalam pixel atau persen (%)

WIDTH , Menentukan lebar dari image, diisi dalam pixel atau persen (%)

Contoh penulisan:

```
<HTML>
<HEAD>
  <TITLE>Membuat IMG</TITLE>
</HEAD>
<BODY>
  
</BODY>
</HTML>
```



TABLES

Berikut ini akan dijelaskan mengenai tag – tag html yang berhubungan dengan pembuatan tabel pada dokumen html.

<TABLE> TAG

Tag <TABLE> adalah untuk mendefinisikan sebuah tabel. Di dalam tabel berisi baris dan sel, dimana baris dibuat dengan tag <TR> sedangkan untuk baris heading gunakan <TH> dan untuk membuat sel digunakan tag <TD>.

Syntax

```
<TABLE  
  ALIGN="LEFT|RIGHT"  
  BGCOLOR="color"  
  BORDER="value"  
  CELLPADDING="value"  
  CELLSPACING="value"  
  HEIGHT="height"  
  WIDTH="width"  
  COLS="numOfCols"  
  HSPACE="horizMargin"  
  VSPACE="vertMargin"  
>  
...  
</TABLE>
```

ALIGN Untuk mendefinisikan posisi peletakan tabel pada dokumen.

- LEFT
- RIGHT
- CENTER

BGCOLOR Untuk mendefinisikan warna background pada tabel. Tetapi warna background ini dapat di tutupi oleh warna background pada tag <TH>, <TR>, dan <TD> jika parameter bgcolor tag – tag tersebut diisi.

BORDER Untuk menentukan ketebalan garis pada tabel.

CELLPADDING Untuk menentukan lebar spasi antara border dan isi sel (default=1)

CELLSPACING Untuk menentukan lebar spasi antar sel (default=2)

HEIGHT Untuk menentukan tinggi tabel

WIDTH Untuk menentukan lebar tabel

<CAPTION> TAG

Untuk mendefinisikan judul dari suatu tabel. Tag <CAPTION> ini diletakkan di dalam tag <TABLE> (antara <TABLE> DAN </TABLE>).

Syntax

```
<CAPTION ALIGN="BOTTOM" | "TOP">...</CAPTION>
```

<TR> TAG

Untuk mendefinisikan baris dalam tabel. Tag <TR> ini diletakkan di dalam tag <TABLE> (antara <TABLE> dan </TABLE>)

Syntax

```
<TR
  ALIGN="CENTER | LEFT | RIGHT"
  VALIGN="BASELINE | BOTTOM | MIDDLE | TOP"
  BGCOLOR="color"
>
...
</TR>
```

ALIGN Untuk mendefinisikan posisi horisontal peletakan baris dalam tabel.

- LEFT

- RIGHT
- CENTER

VALIGN Untuk mendefinisikan posisi vertikal peletakan baris dalam tabel.

- BASELINE
- BOTTOM
- MIDDLE
- TOP

BGCOLOR Untuk mendefinisikan warna background pada baris. Tetapi warna background ini dapat di tutupi oleh warna background dari `<TD>` jika parameter bgcolor `<TD>` tersebut diisi.

<TD> TAG

Untuk mendefinisikan sel dalam baris. Tag `<TD>` ini diletakkan di dalam tag `<TR>` (antara `<TR>` dan `</TR>`).

Syntax

```
<TD
  ALIGN="CENTER | LEFT | RIGHT"
  VALIGN="BASELINE | BOTTOM | MIDDLE | TOP"
  BGOLOR="color"
  COLSPAN="value"
  ROWSPAN="value"
  HEIGHT="pixelHeight"
  WIDTH="pixelWidth"
  NOWRAP="value"
>
...
</TD>
```

ALIGN Untuk mendefinisikan posisi horisontal peletakan baris dalam tabel.

- LEFT
- RIGHT
- CENTER

VALIGN Untuk mendefinisikan posisi vertikal peletakan baris dalam tabel.

- BASELINE
- BOTTOM
- MIDDLE
- TOP

BGCOLOR Untuk mendefinisikan warna background pada sel.

COLSPAN Untuk mendefinisikan jumlah kolom yang digabungkan

ROWSPAN Untuk mendefinisikan jumlah baris yang digabungkan

HEIGHT Untuk mendefinisikan tinggi sel

WIDTH Untuk mendefinisikan lebar sel

NOWRAP Untuk mendefinisikan bahwa text dalam sel hanya terdiri dari satu baris.

<TH> TAG

Untuk mendefinisikan baris heading dalam tabel. Tag <TH> ini diletakkan di dalam tag <TR> (antara <TR> dan </TR>).

Syntax

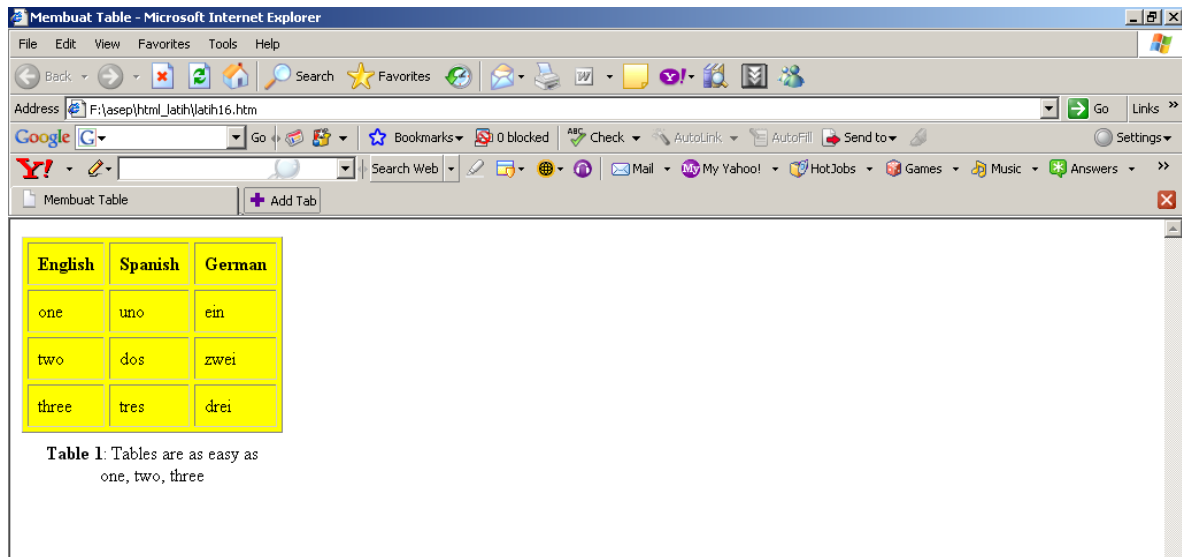
```
<TH
  ALIGN="CENTER | LEFT | RIGHT"
  VALIGN="BASELINE | BOTTOM | MIDDLE | TOP"
  BGCOLOR="color"
  COLSPAN="value"
  ROWSPAN="value"
  HEIGHT="pixelHeight"
  WIDTH="pixelWidth"
  NOWRAP
>
...
</TH>
```

ALIGN	Untuk mendefinisikan posisi horisontal peletakan baris dalam tabel. • LEFT • RIGHT • CENTER
VALIGN	Untuk mendefinisikan posisi vertikal peletakan baris dalam tabel. • BASELINE • BOTTOM • MIDDLE • TOP
BGCOLOR	Untuk mendefinisikan warna background pada sel.
COLSPAN	Untuk mendefinisikan jumlah kolom yang digabungkan
ROWSPAN	Untuk mendefinisikan jumlah baris yang digabungkan
HEIGHT	Untuk mendefinisikan tinggi sel
WIDTH	Untuk mendefinisikan lebar sel
NOWRAP	Untuk mendefinisikan bahwa text dalam sel hanya terdiri dari satu baris.

```

<HTML>
<HEAD>
  <TITLE>Membuat Table</TITLE>
</HEAD>
<BODY>
  <TABLE BORDER CELLPADDING="8" CELLSPACING="4" BGCOLOR=yellow>
    <TR><TH> English </TH><TH> Spanish </TH><TH> German </TH></TR>
    <TR><TD> one      </TD><TD> uno      </TD><TD> ein      </TD></TR>
    <TR><TD> two      </TD><TD> dos      </TD><TD> zwei     </TD></TR>
    <TR><TD> three    </TD><TD> tres     </TD><TD> drei     </TD></TR>
    <CAPTION ALIGN="BOTTOM"> <B>Table 1</B>: Tables are as easy as
      one, two, three
    </CAPTION>
  </TABLE>
</BODY>
</HTML>

```

FRAMES

Frame biasa digunakan untuk membagi satu tampilan pada browser agar menampilkan beberapa dokumen HTML.

<FRAMESET> TAG

Tag <FRAMESET> digunakan untuk mendefinisikan suatu set frame – frame yang akan di tampilkan pada window browser. Di dalam tag <FRAMESET> (antara <FRAMESET> dan </FRAMESET>) terdiri dari beberapa tag <FRAME>, tiap satu tag frame merupakan definisi dari satu frame (bagian).

Syntax

```
<FRAMESET
  COLS="columnWidthList"
  ROWS="rowHeightList"
  BORDER="pixWidth"
  BORDERCOLOR="color"
  FRAMEBORDER="YES" | "NO"
>
...
</FRAMESET>
```

COLS Menentukan lebar dari kolom – kolom pembagian frame, list nilai tersebut ditulis dengan comma-separated. Nilai dapat diisi dalam pixel, persen dan * (selebar mungkin).

ROWS Menentukan tinggi dari baris – baris pembagian frame, list nilai tersebut ditulis dengan comma-separated. Nilai dapat diisi dalam pixel, persen dan * (setinggi mungkin).

BORDER Menentukan ketebalan garis pemisah antar frame.

BORDERCOLOR Menentukan warna garis pemisah antar frame

FRAMEBORDER Menentukan bagaimana garis pemisah antar frame ditampilkan:
YES → outline-3D border.
NO → suppresses the 3D border.

<FRAME> TAG

Tag <FRAME> adalah tag yang digunakan untuk membuat frame. Frame ini merupakan suatu bagian pada web browser yang dapat diatur secara individual (atau bisa dibilang *window di dalam window*).

Tag <FRAME> harus digunakan didalam tag <FRAMESET> (antara <FRAMESET> dan </FRAMESET>).

Syntax

```
<FRAME
  NAME="frameName"
  NORESIZE
  SCROLLING="YES" | "NO" | "AUTO"
  SRC="URL "
>
```

NAME	Menentukan nama dari frame. Nilai dari name harus diawali dengan huruf.
NORESIZE	Apabila ditulis maka frame tersebut tidak dapat di resize (fixed)
SCROLLING	Menentukan apakah frame dapat di-scroll atau tidak. YES → scrollbars selalu ada. NO → tidak ada scrollbars AUTO → scrollbars akan muncul secara otomatis begitu isi dari frame melebihi lebar atau tinggi frame.
SRC	Menentukan alamat (nama dokumen) dari isi yang ditampilkan pada frame

<NOFRAMES> TAG

Tag <NOFRAMES> menampilkan isi dari dokumen apabila browser yang digunakan user tidak mendukung frame.

Tag ini tidak akan diakses apabila browser yang digunakan adalah browser yang mendukung frame.

Syntax

```
<NOFRAMES> . . . </NOFRAMES>
```

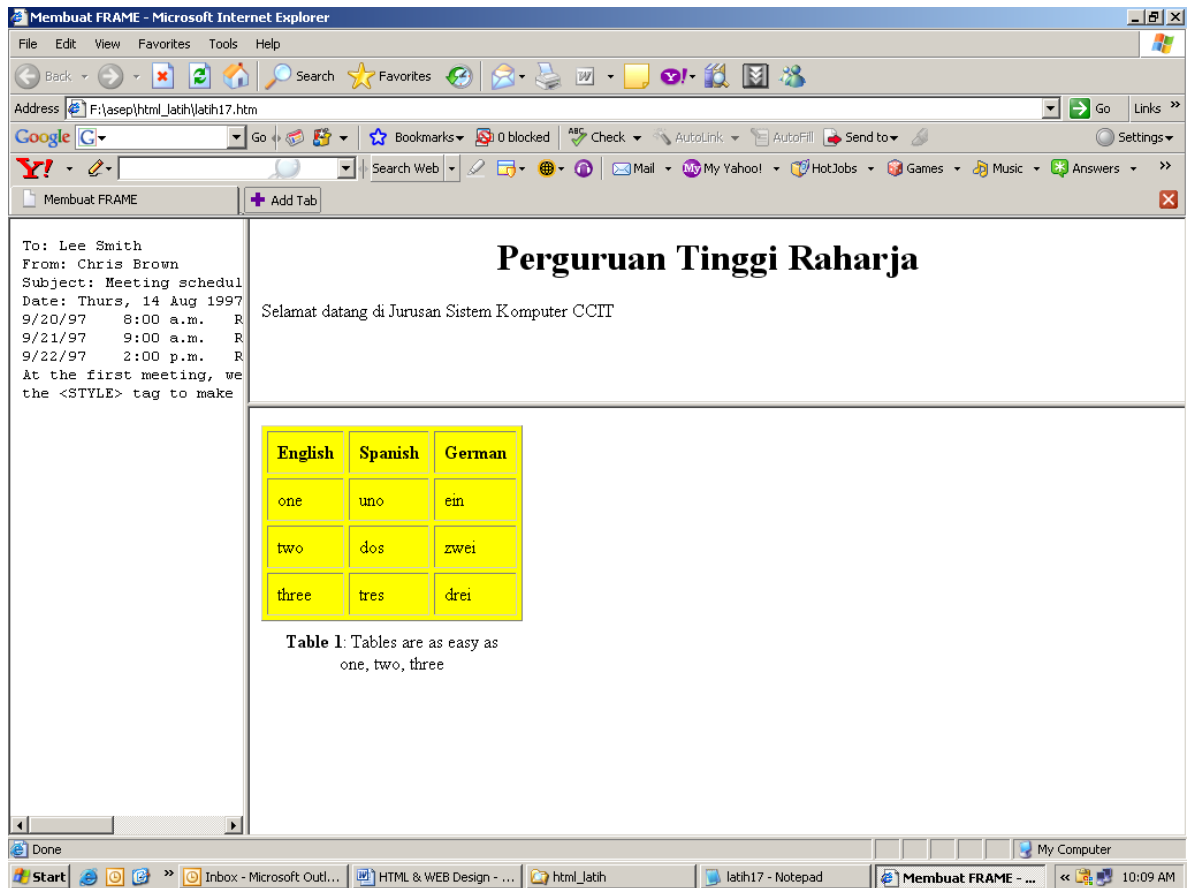
Perhatikan contoh berikut:

```
<HTML>
<HEAD>
  <TITLE>Membuat FRAME</TITLE>
</HEAD>

<FRAMESET COLS="20%,*">
  <FRAME SRC="frLeft.htm" NAME="noname">
    <FRAMESET ROWS="30%,*">
      <FRAME SRC="frTop.htm" NAME="toptoc">
        <FRAME SRC="frMain.htm" NAME="outer">
      </FRAMESET>
    </FRAMESET>
  </FRAMESET>
<NOFRAMES>
<BODY>
  You must use a browser that can display frames to see this page.
</BODY>
</NOFRAMES>
</HTML>
```

Perhatian:

Sebelum mencoba contoh di atas, dianjurkan untuk membuat dokumen HTML dengan nama: frLeft.htm, frTop.htm dan frMain.htm



CSS

CSS → Cascading Style Sheet : Salah satu fasilitas yang diberikan untuk pemrograman HTML sehingga pengaturan/disain tampilan *web-page* menjadi lebih baik.

Dengan atribut CSS, user diberi kemampuan untuk:

- Mengatur posisi secara absolut
- Mengubah warna
- Mengubah besar font
- Mengubah margin
- Dsb.

Ada tiga metode cara penulisan CSS atribut, yaitu:

1. ***Inline Style Sheet***
2. ***Embedded Style Sheet***
3. ***Linked Style Sheet***

INLINE STYLE SHEET

CSS didefinisikan langsung pada tag HTML yang bersangkutan. Cara penulisannya cukup dengan menambahkan atribut **style="** “ didalam tag HTML tersebut.

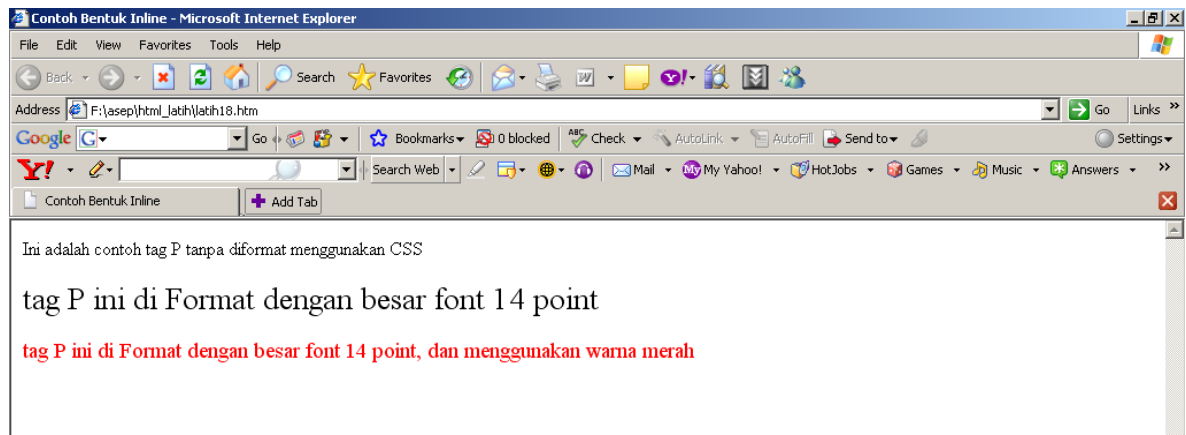
style hanya akan berlaku pada tag yang bersangkutan, dan tidak akan mempengaruhi tag HTML yang lain.

Contoh penulisan CSS dengan metode *inline style sheet*: (simpan dgn nama **CSS1.htm**)

```
<HTML>
<Head>
  <Title>Contoh Bentuk Inline</Title>
</Head>
<Body bgcolor="#FFFFFF">
  <P id="cth1">
    Ini adalah contoh tag P tanpa diformat menggunakan CSS
  </P>

  <P id="cth2" style="font-size:20pt">
    tag P ini di Format dengan besar font 14 point
  </p>

  <P id="cth3" style="font-size:14pt ; color:red">
    tag P ini di Format dengan besar font 14 point, dan
    menggunakan warna merah
  </p>
</Body>
</HTML>
```



EMBEDDED STYLE SHEET

CSS didefinisikan terlebih dahulu dalam tag `<style> .....</style>` di atas tag `<body>`.

Pada pendefinisian ini disebutkan atribut – atribut CSS yang akan digunakan untuk tag-tag HTML yang selanjutnya tinggal digunakan oleh tag HTML yang bersangkutan.

 INF

- Tahukah anda bahwa setiap tag HTML dapat di beri ID.
 - ID pada tag HTML berguna untuk pemanggilan tag tersebut melalui script
-

Contoh penggunaan CSS dengan metode ***embedded style sheet***: (simpan dgn nama ***CSS2.htm***)

```

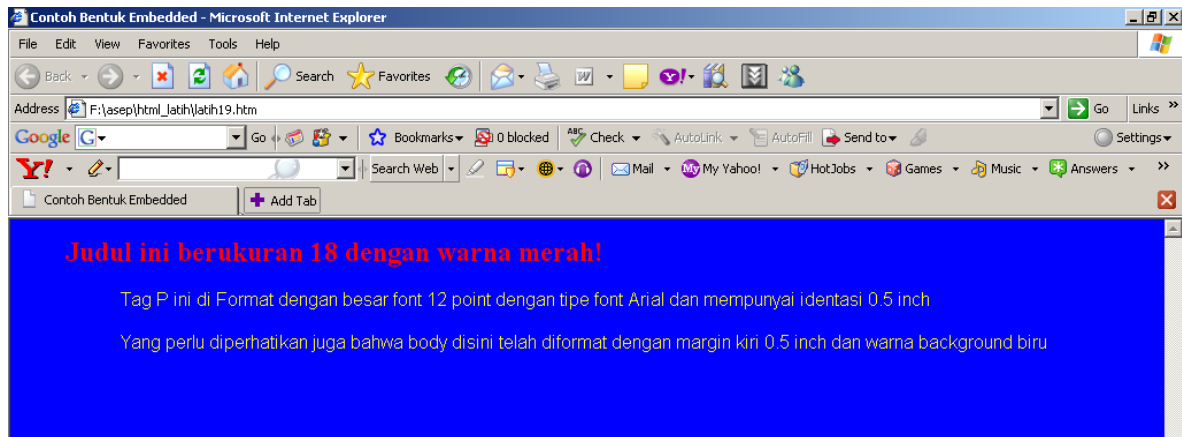
<HTML>
<Head>
  <Title>Contoh Bentuk Embedded</Title>
</Head>

<Style>
  Body {background:#0000FF; color:#FFFF00; margin-left:0.5in}
  H1 {font-size:18pt; color:#FF0000}
  P {font-size:12pt; font-family:arial; text-indent:0.5in}
</Style>

<Body>
  <H1 id="cth1">Judul ini berukuran 18 dengan warna merah!</H1>
  <P id="cth2">
    Tag P ini di Format dengan besar font 12 point dengan
    tipe font Arial dan mempunyai identasi 0.5 inch
  </p>

  <P id="cth3">
    Yang perlu diperhatikan juga bahwa body disini telah
    diformat dengan margin kiri 0.5 inch dan warna
    background biru
  </p>
</Body>
</HTML>

```

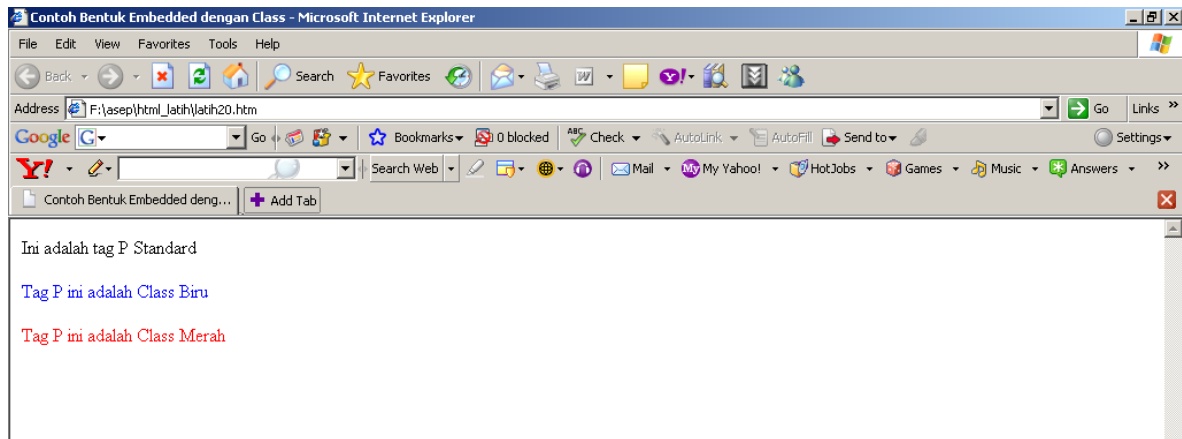



Contoh CSS metode *embedded style sheet* dengan *class*: (simpan dgn nama *CSS3.htm*)

```
<HTML>
<Head>
  <Title>Contoh Bentuk Embedded dengan Class</Title>
</Head>

<Style>
  P.biru {font-size:12pt; color:blue}
  P.merah {font-size:12pt; color:red}
</Style>

<Body>
  <P id="cth1">Ini adalah tag P Standard</P>
  <P id="cth2" Class="biru">Tag P ini adalah Class Biru</p>
  <P id="cth3" Class="Merah">Tag P ini adalah Class Merah</p>
</Body>
</HTML>
```



LINKED STYLE SHEET

Metode ini hampir sama dengan metode *embedded style sheet*, hanya saja pendefinisian tag `<style> .... </style>` dibuat **pada file terpisah** dari file HTML yang membutuhkan CSS tersebut. File lain tersebut di simpan dalam format **.css**

Pada file HTML yang akan menggunakan file CSS tersebut harus dibuat tag `<link>` yang dituliskan diantara tag `<head> ... </head>`

Contoh file .css: (simpan dgn nama **StyleSheet4.css**)

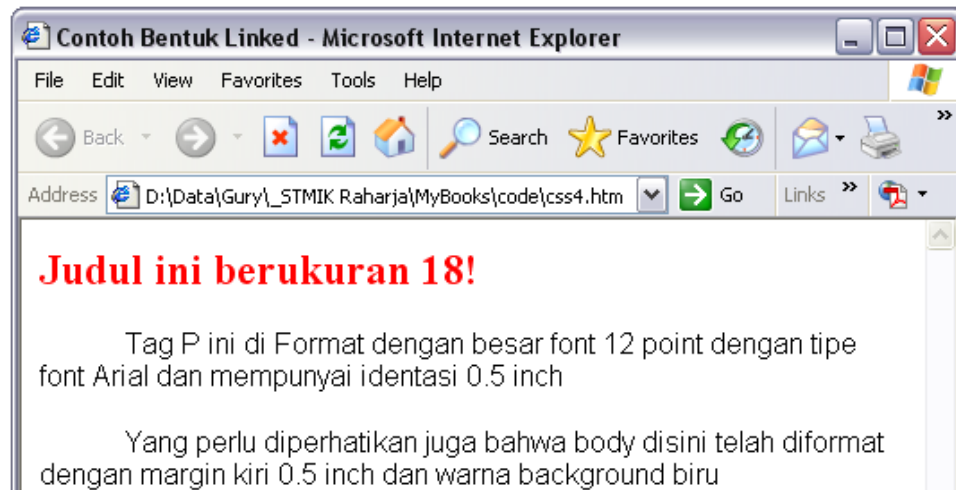
```
<Style>
  Body {background:#0000FF; color:#FFFF00; margin-left:0.5in}
  H1 {font-size:18pt; color:#FF0000}
  P {font-size:12pt; font-family:arial; text-indent:0.5in}
</Style>
```

Contoh file HTML yang menggunakan file .css: (simpan dgn nama **CSS4.htm**)

```
<HTML>
<Head>
  <Title>Contoh Bentuk Linked</Title>
  <Link rel=stylesheet href="StyleSheet4.css" type="text/css">
</Head>

<Body>
  <H1 id="cth1">Judul ini berukuran 18!</H1>
  <P id="cth2">
    Tag P ini di Format dengan besar font 12 point dengan
    tipe font Arial dan mempunyai identasi 0.5 inch
  </p>

  <P id="cth3">
    Yang perlu diperhatikan juga bahwa body disini telah
    diformat dengan margin kiri 0.5 inch dan warna
    background biru
  </p>
</Body>
</HTML>
```



CSS SYNTAX

CSS syntax biasanya terdiri dari tiga bagian, yaitu; **selector**, **property** dan **value**.

selector {property: value}

Selector biasanya berupa HTML elemen / tag Anda ingin tentukan tampilannya, **property** adalah atribut dari selector yang akan diubah, dan tiap property memiliki value (nilai). Property dan value (nilai) dipisahkan oleh sebuah titik-dua, dan ditulisa didalam kurung kurawal.

Contoh penulisan syntax css:

```
p {font-family: "sans serif"}
```

Berdasarkan contoh penulisan di atas, berarti tag HTML yang akan didefinisikan adalah tag **p** (<p>) sedangkan property yang akan diberi nilai adalah **font-family** nilai yang diberikan adalah **sans serif**. Dengan demikian maka setiap tulisan yang diapit oleh tag p (berada diantara <p> dan </p>) akan ditampilkan dengan font jenis sans serif.

Mendefinisikan Lebih dari Satu Property

Untuk mendefinisikan lebih dari satu property terhadap selector dapat digunakan pemisah titik-koma (;) untuk memisahkan tiap definisi, perhatikan contoh berikut:

```
p {font-family:arial; text-align:center; color:red}
```

Berdasarkan contoh penulisan di atas, berarti setiap tulisan yang diapit oleh tag **p** akan ditampilkan dengan font jenis arial dengan posisi tengah dan warna merah. Contoh penulisan syntax untuk lebih dari satu property juga dapat ditulis sebagai berikut:

```
P {  
  font-family:arial;  
  text-align:center;  
  color:red  
}
```

Mendefinisikan untuk Lebih dari Satu Selector (Grouping)

Untuk mendefinisikan terhadap lebih dari satu selector secara grouping dapat dilakukan dengan menggunakan koma (,) sebagai pemisah antar selector, perhatikan contoh berikut:

```
h1, h2, h3 { color: green }
```

Berdasarkan contoh penulisan di atas, maka setiap tulisan yang diapit oleh tag **h1** atau **h2** atau **h3** akan memiliki warna hijau.

Class Selector

Dengan kelas selector Anda dapat mendefinisikan berbagai style atau gaya yang berbeda untuk type element HTML yang sama. Katakanlah Anda ingin memiliki dua jenis paragraf (tag **p**) di dokumen Anda: satu jenis untuk paragraph rata kanan (right-aligned) dan satu jenis untuk paragraph rata tengah (center-aligned). Berikut adalah contoh bagaimana melakukannya dengan dengan css:

```
p.kanan {text-align: right}  
p.tengah {text-align: center}
```

Berdasarkan contoh di atas maka pada dokumen HTML untuk menentukan tag p mana yang rata kanan (right-aligned) dan mana yang rata tengah (center-aligned) kita harus memberikan class pada tag p tersebut, perhatikan contoh potongan dokumen HTML berikut:

```
<p class="kanan">  
This paragraph will be right-aligned.  
</p>  
<p class="tengah">  
This paragraph will be center-aligned.  
</p>
```

Untuk membuat class yang berlaku untuk semua element HTML (tag HTML) kita dapat membuatnya dengan tanpa menuliskan selectornya, cukup hanya nama class yang diawali dengan titik (.). Perhatikan contoh berikut:

```
.kanan {text-align: right}  
.tengah {text-align: center}
```

Dengan mendefinisikan seperti di atas maka setiap elemen HTML (tag HTML) yang memiliki `class="kanan"` maka akan mengikuti format sesuai yang didefinisikan pada **.kanan**. Perhatikan contoh potongan dokumen HTML berikut:

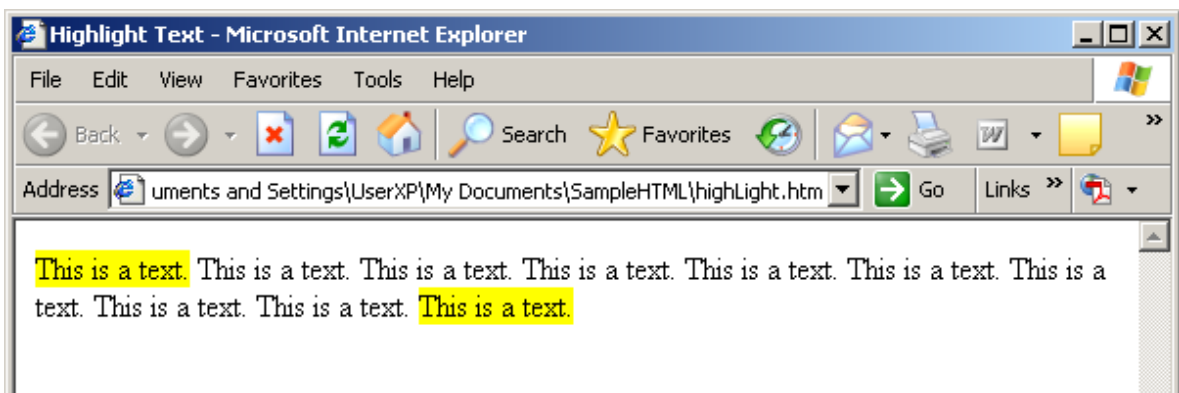
```
<p class="kanan">
This paragraph will be right-aligned.
</p>
<h1 class="tengah">
This paragraph will be center-aligned.
</h1>
```

Membuat HighLight pada Text

```
<html>
<head>
<style type="text/css">
span.highlight
{
    background-color:yellow
}
</style>
</head>

<body>
<p>
<span class="highlight">This is a text.</span> This is a text. This
is a text. This is a text. This is a text. This is a text. This is
a text. This is a text. This is a text. <span
class="highlight">This is a text.</span>
</p>
</body>
</html>
```

Hasil pada browser:

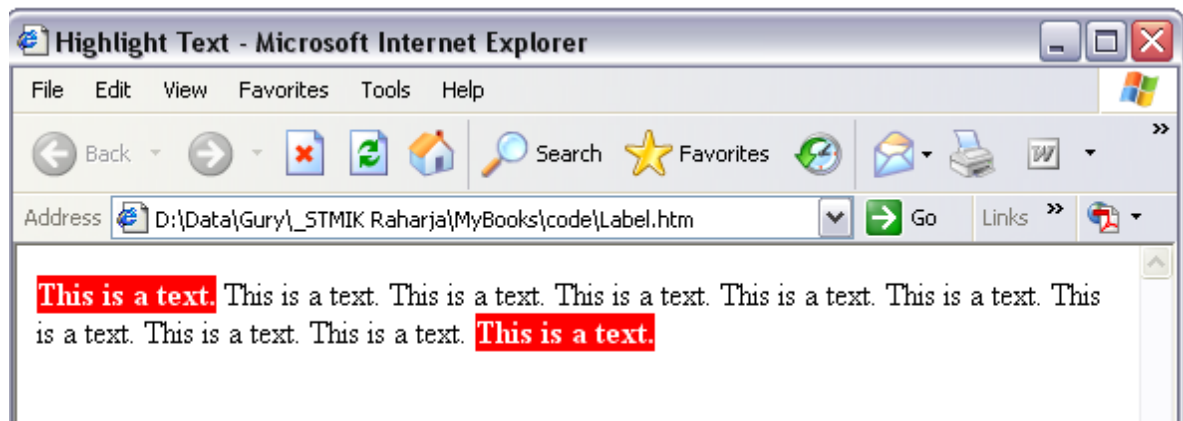


Membuat HighLight pada Text

```
<html>
<head>
<style type="text/css">
span.label
{
    background-color:red;
    font-weight:bold;
    color:white;
}
</style>
</head>

<body>
<p>
<span class="label">This is a text.</span> This is a text. This is
a text. This is a text. This is a text. This is a text. This is a
text. This is a text. This is a text. <span class="label">This is a
text.</span>
</p>
```

Hasil pada browser:

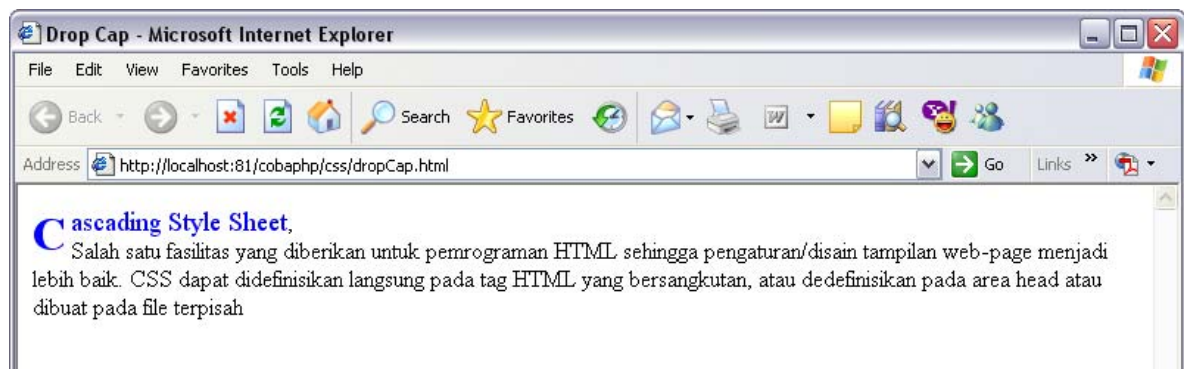


Membuat Drop Cap pada Huruf

```
<html>
<head>
<style type="text/css">
.dropcap {
    font-size:xx-large;
    font-weight:bold;
    color:blue;
    text-transform:capitalize;
    float:left;
}
.normal {
    font-size:large;
    color:blue;
}
</style>
<title>Drop Cap</title>
</head>

<body>
<span class="dropcap">c</span><span class="normal">ascading Style
Sheet</span>, <br />
Salah satu fasilitas yang diberikan untuk pemrograman HTML sehingga
pengaturan/disain tampilan web-page menjadi lebih baik.<br />
CSS dapat didefinisikan langsung pada tag HTML yang bersangkutan,
atau didefinisikan pada area head atau dibuat pada file terpisah
</body>
</html>
```

Hasil pada browser:



Menambahkan Border pada Huruf Drop Cap

```

<html>
<head>
<style type="text/css">
.dropcap {
    font-size:xx-large;
    font-weight:bold;
    color:blue;
    text-transform:capitalize;
    float:left;
    border-top-style: solid;
    border-right-style: solid;
    border-bottom-style: solid;
    border-left-style: solid;
    border-color: blue;
    border-top-width: thin;
    border-right-width: thin;
    border-bottom-width: thin;
    border-left-width: thin;
    padding-right:5px;
    padding-left:5px;
}
.judulBiru {
    font-size:large;
    color:blue;
}
</style>
<title>Drop Cap</title>
</head>

<body>
<span class="dropcap">c</span><span class="judulBiru">ascading Style
Sheet</span>, <br />
Salah satu fasilitas yang diberikan untuk pemrograman HTML sehingga
pengaturan/disain tampilan web-page menjadi lebih baik.<br />
CSS dapat didefinisikan langsung pada tag HTML yang bersangkutan,
atau dedefinisikan pada area head atau dibuat pada file terpisah
</body>
</html>

```

Hasil pada browser:



Nilai yang dapat diberikan pada border style diantaranya:

- dotted
- dashed
- solid
- double
- groove
- inset
- outset

Jika pada border style kita beri nilai sebagai berikut:

```
border-top-style: dotted;  
border-right-style: dotted;  
border-bottom-style: dotted;  
border-left-style: dotted;
```

Hasil pada browser dari perubahan kode css adalah sebagai berikut:



Untuk menggunakan border style double, agar bordernya terlihat double maka border width harus di set menjadi tebal (thick). Perhatikan perubahan yang dilakukan pada kode css berikut:

```
<style type="text/css">
.dropcap {
    font-size:xx-large;
    font-weight:bold;
    color:blue;
    text-transform:capitalize;
    float:left;
    border-top-style: double;
    border-right-style: double;
    border-bottom-style: double;
    border-left-style: double;
    border-color: blue;
    border-top-width: thick;
    border-right-width: thick;
    border-bottom-width: thick;
    border-left-width: thick;
    padding-right:5px;
    padding-left:5px;
}
.normal {
    font-size:large;
    color:blue;
}
</style>
```

Hasil pada browser dari perubahan kode css adalah sebagai berikut:



Jika pada border style kita beri nilai sebagai berikut:

```
border-top-style: groove;  
border-right-style: groove;  
border-bottom-style: groove;  
border-left-style: groove;
```

Hasil pada browser dari perubahan kode css adalah sebagai berikut: (border berupa embosed)



Jika pada border style kita beri nilai sebagai berikut:

```
border-top-style: inset;  
border-right-style: inset;  
border-bottom-style: inset;  
border-left-style: inset;
```

Hasil pada browser dari perubahan kode css adalah sebagai berikut: (border berupa bevel)



Membuat Huruf Iconic

```

<html>
<head>
<style type="text/css">
.dropcap {
    font-size:xx-large;
    font-weight:bold;
    color:blue;
    text-transform:capitalize;
    float:left;
    border-top-style: inset;
    border-right-style: inset;
    border-bottom-style: inset;
    border-left-style: inset;
    border-color:blue;
    border-top-width: thick;
    border-right-width: thick;
    border-bottom-width: thick;
    border-left-width: thick;
    padding-right:5px;
    padding-left:5px;
    background-color:#FF0000
}
.normal {
    font-size:large;
    color:blue;
}</style>
<title>Drop Cap</title>
</head>

<body>
<span class="dropcap">c</span><span class="judulBiru">ascading Style
Sheet</span>, <br />
Salah satu fasilitas yang diberikan untuk pemrograman HTML sehingga
pengaturan/disain tampilan web-page menjadi lebih baik.<br />
CSS dapat didefinisikan langsung pada tag HTML yang bersangkutan,
atau dedefinisikan pada area head atau dibuat pada file terpisah
</body>
</html>

```

Hasil pada browser:



Membuat Hover Link CSS

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Test</title>
<style type="text/css">
<!--
.float {
    float: left;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: xx-large;
    line-height: 100%;
    font-weight: bolder;
    color: #FFFFFF;
    background-color: #0000FF;
    padding: 3px;
    clear: left;
    border: thin solid #FF0000;
}
a {
    text-decoration: none;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #0033FF;
    font-size: large;
}
a:hover {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: large;
    color: #00FF00;
    background-color: #FF0000;
    border-top-style: solid;
    border-right-style: solid;
    border-bottom-style: solid;
    border-left-style: solid;
    border-top-width: thin;
    border-right-width: thin;
    border-bottom-width: thin;
    border-left-width: thin;
}
.spesial1 {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: large;
    color: #FF0000;
    font-weight: bolder;
    padding: 2px;
}
}
```

Lanjutan

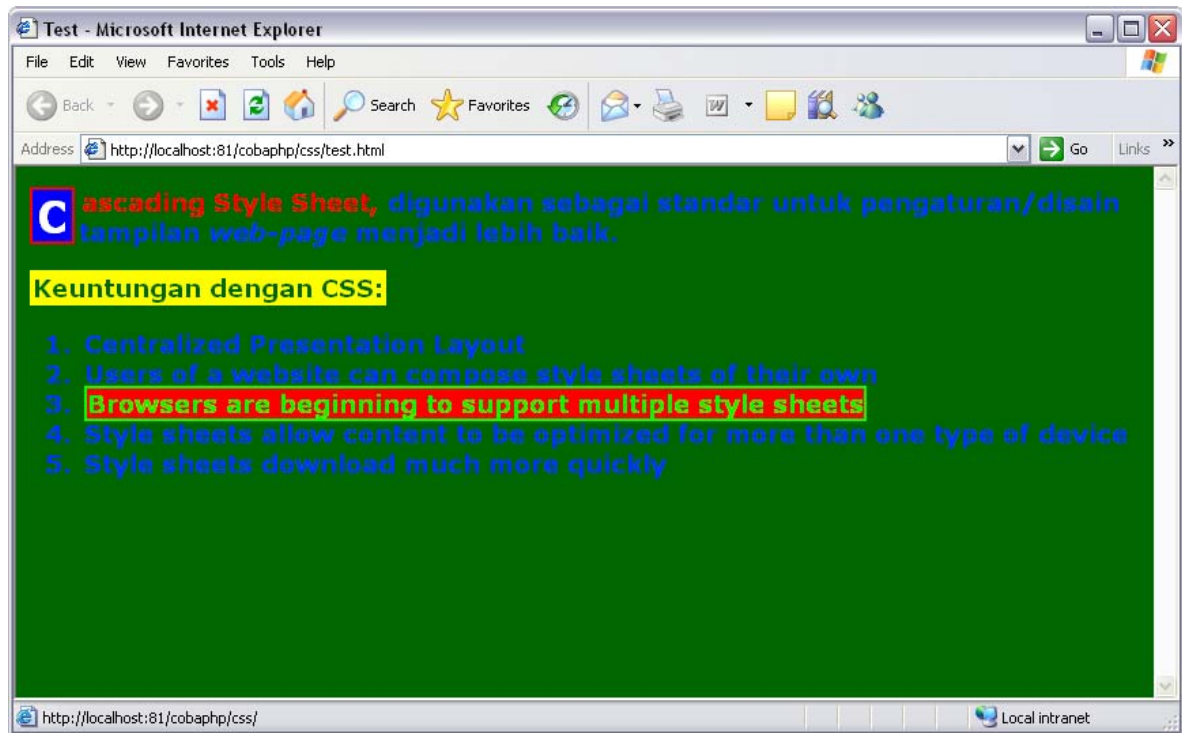
```

.spesial2 {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: large;
    color: #006600;
    font-weight: bolder;
    padding: 2px;
    background-color: #FFFF00;
}
.biasa {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: large;
    font-weight: bold;
    color: #0033FF;
}
body {
    background-color: #006600;
}
-->
</style>
</head>

<body>
<p><span class="float">C</span><span class="spesial1">ascading Style
Sheet,</span> <span class="biasa">digunakan sebagai standar untuk
pengaturan/disain tampilan <em>web-page</em> menjadi lebih
baik.</span></p>
<p><span class="spesial2">Keuntungan dengan CSS:</span>
<ol class="biasa">
    <li><a href="">Centralized Presentation Layout</a></li>
    <li><a href="">Users of a website can compose style sheets of
their own</a><br /></li>
    <li><a href="">Browsers are beginning to support multiple style
sheets</a></li>
    <li><a href="">Style sheets allow content to be optimized for
more than one type of device</a><br /></li>
    <li><a href="">Style sheets download much more quickly</a><br
/></li>
</ol>
</p>
</body>
</html>

```


Hasil pada browser:



ISI ELEMEN STYLE YG DINAMIS

Sekarang kita akan mempelajari cara merubah attribut style suatu CSS pada saat *run-time*. Hal ini dapat dilakukan dengan metode *scripting*, yaitu pengubahan attribut style dengan menggunakan *script*.

Jangan lupa untuk selalu memberikan attribut **ID** pada setiap *tag HTML*, yang nantinya program *script* akan mengakses *tag HTML* tersebut dengan menyebutkan **ID** -nya.

Attribut color

Perhatikan potongan HTML berikut:

```
<P ID="Coba" style="color:blue">coba-coba</P>
```

Tag **<P>** pada potongan HTML di atas memiliki **ID = coba**, sehingga nantinya pada *script*. Apabila kita ingin mengubah warna tulisan **coba-coba** menjadi merah, dapat dilakukan dengan *script* sebagai berikut:

```
Coba.style.color="red"
```

Cobalah buat HTML seperti berikut: (simpan dgn nama **CSS5.htm**)

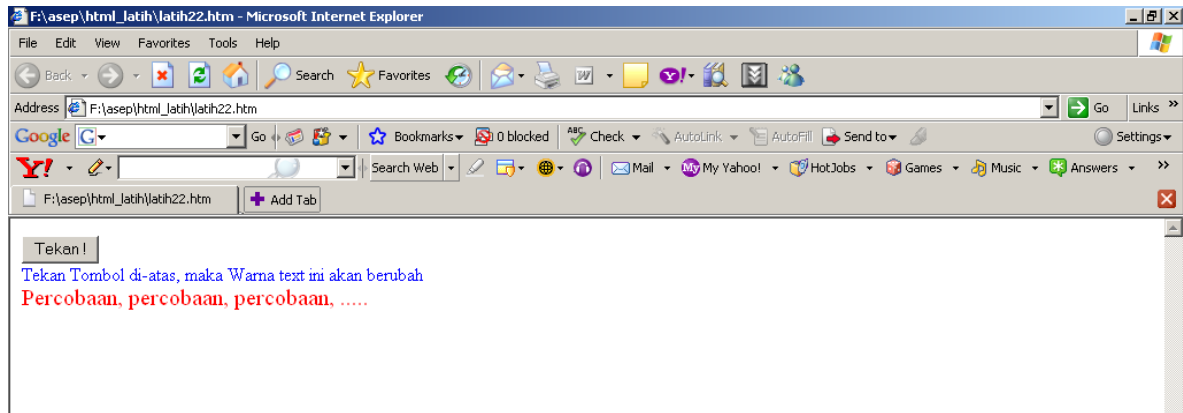
```
<HTML>
<Script Language="JScript">
function ubah()
{
    Kata1.style.color="green"
    Kata2.style.color="purple"
    Kata2.style.fontSize="30pt"
}
</Script>

<Body>
    <Input type=button value="Tekan !" onclick="ubah()"><BR>
    <Span ID="Kata1" style="color:blue">
        Tekan Tombol di-atas, maka Warna text ini akan berubah
    </Span> <BR>
    <Span ID="Kata2" style="color:red ; font-size:15pt">
        Percobaan, percobaan, percobaan, ....
    </Span>
</Body>
</HTML>
```

Attribut font-size

Pada **tag HTML** penulisan attribut style untuk merubah ukuran font adalah **font-size**.

Pada **script** penulisan attribut style untuk merubah ukuran font adalah **fontsize**.



EVENT – EVENT UNTUK TAG HTML

Beberapa **event** untuk **tag HTML** dapat dimanfaatkan untuk membuat *page* terlihat lebih baik. Berikut ini adalah *event – event* tersebut:

- **onMouseOver**
Jika mouse sedang digerakkan dan berada pada daerah ****
- **onMouseOut**
Jika mouse keluar dari daerah ****
- **onClick**
Jika mouse di click pada daerah ****.

Event – event tersebut di atas dapat dimanfaatkan untuk memanggil fungsi pada *script*, hal ini memberikan fleksibilitas bagi *user* untuk memperindah tampilan *page*-nya agar menjadi dinamis.

Cobalah buat HTML seperti berikut: (simpan dgn nama **CSS6.htm**)

```
<HTML>

<Style>
  Span {font-size : 20pt}
</Style>

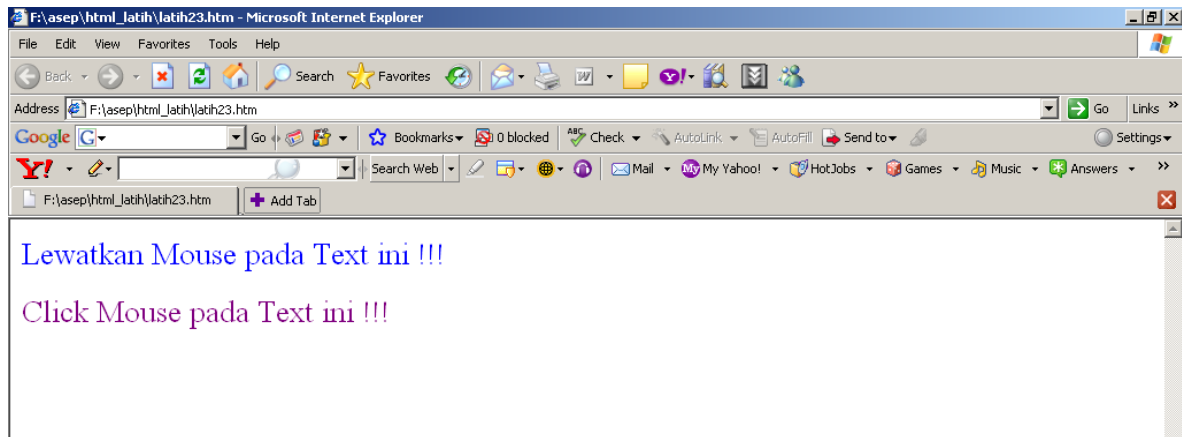
<Script Language="JScript">
  function tekan()
  {
    Kata2.style.color="green"
  }

  function lewat()
  {
    Kata1.style.color="red"
  }

  function keluar()
  {
    Kata1.style.color="blue"
  }
</Script>

<Body>
  <Span ID="Kata1" style="color:blue" onmouseover="lewat()"
  onmouseout="keluar()">
    Lewatkan Mouse pada Text ini !!!
  </Span>
  <BR><BR>
  <Span ID="Kata2" style="color:purple" onclick="tekan()">
    Click Mouse pada Text ini !!!
  </Span>
</Body>

</HTML>
```



Dan Cobalah buat HTML seperti berikut, serta bandingkan dengan yang sebelumnya!
(simpan dgn nama **CSS7.htm**)

```

<HTML>

<Style>
  Span {cursor: hand; font-size: 18pt}
</Style>

<Script Language="JScript">
  function tekan()
  {
    tampung.value="Halo Semua !"
  }

  function lewat()
  {
    Kata1.style.color="red"
  }

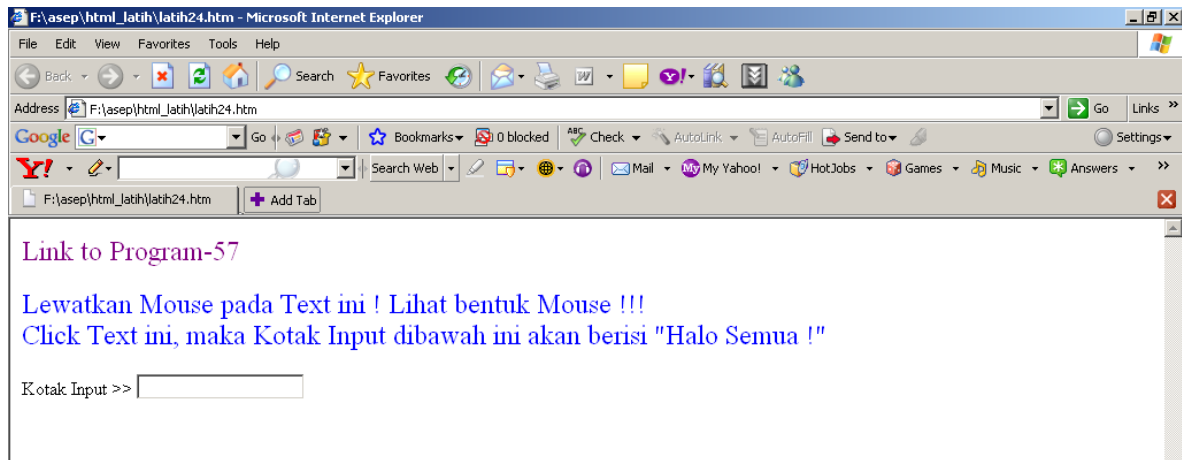
  function keluar()
  {
    Kata1.style.color="blue"
  }
</Script>

<Body>
  <Span ID="Kata2" style="color:purple"
    onclick="window.location.href='CSS6.htm'">
    Link to Program-57
  </Span><BR><BR>

  <Span ID="Kata1" style="color:blue" onmouseover="lewat()"
    onmouseout="keluar()" onclick="tekan()">
    Lewatkan Mouse pada Text ini ! Lihat bentuk Mouse !!!
    <BR>
    Click Text ini, maka Kotak Input dibawah ini akan
    berisi "Halo Semua !"
  </Span>
  <BR><BR>
  Kotak Input >> <input ID="tampung" Type="text">
</Body>

</HTML>

```



ATTRIBUT VISIBILITY & DISPLAY

Attribut ini bisa digunakan untuk mengatur penampakan suatu **tag HTML**. Sehingga penampakan komponen di dokumen dapat lebih dinamis.

Berikut penjelasan mengenai kedua *attribut* tersebut:

- **Attribut Visibility**

Attribut ini dapat diberi dua buah nilai, yaitu :

“**hidden**” berarti tidak tampak

“**visible**” berarti tampak

- **Attribut Display**

Attribut ini dapat diberi dua buah nilai, yaitu :

none berarti tidak tampak

“ ” berarti tampak

Visibility dan **display** sama – sama untuk mengatur penampakan suatu **tag HTML** pada page, tetapi ada satu hal yang membedakan kedua *attribut* ini, yaitu:

- **Visibility** hanya mengatur tampak atau tidaknya suatu **tag HTML**.
- **Display** tidak hanya mengatur tampak atau tidaknya suatu **tag HTML** , tetapi juga mengatur ruang yang ditempati **tag HTML** tersebut.

TIP

Attribut `display` cocok untuk pembuatan menu yang dapat di-collapse dan di-expand.

Bentuk menu yang seperti ini akan menambah dinamis suatu web page.

Dan Cobalah buat HTML seperti berikut, serta bandingkan hasil dari atribut `display` dan `visibility`! (simpan dgn nama *CSS8.htm*)

```
<HTML>
<Style>
  Div {font-size: 12pt;}
</Style>

<Script Language="JScript">

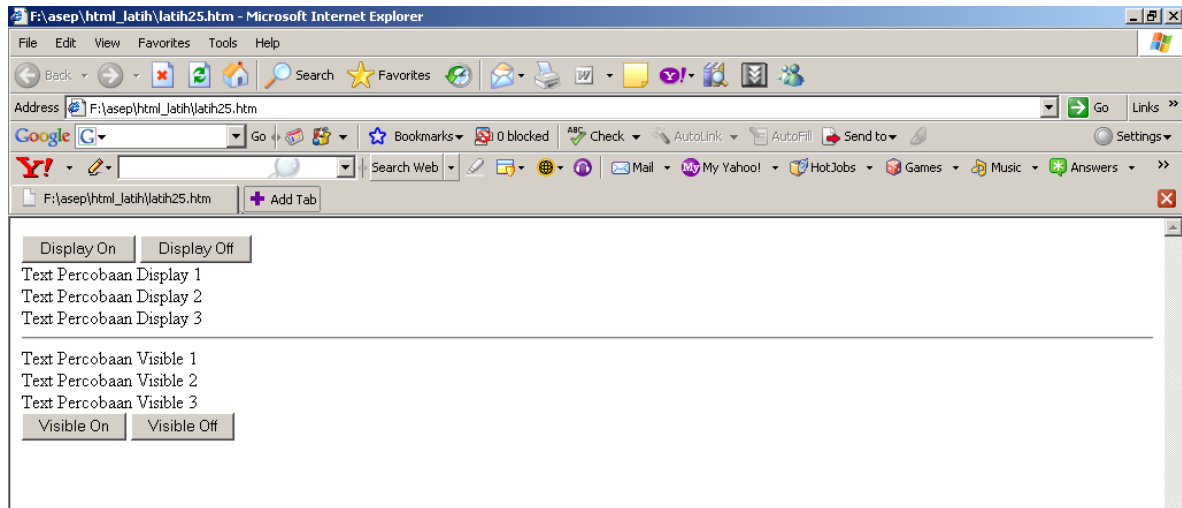
  function disp_on()
  {
    kata12.style.display=""
  }

  function disp_off()
  {
    kata12.style.display="none"
  }

  function vis_on()
  {
    kata22.style.visibility="visible"
  }

  function vis_off()
  {
    kata22.style.visibility="hidden"
  }
</Script>

<Body>
  <Input type="Button" onclick="disp_on()" value="Display On">
  <Input type="Button" onclick="disp_off()" value="Display Off">
  <BR>
  <Div ID="kata11">Text Percobaan Display 1</Div>
  <Div ID="kata12">Text Percobaan Display 2</Div>
  <Div ID="kata13">Text Percobaan Display 3</Div>
  <hr>
  <Div ID="kata21">Text Percobaan Visible 1</Div>
  <Div ID="kata22">Text Percobaan Visible 2</Div>
  <Div ID="kata23">Text Percobaan Visible 3</Div>
  <Input type="Button" onclick="vis_on()" value="Visible On">
  <Input type="Button" onclick="vis_off()" value="Visible Off">
</Body>
</HTML>
```

Cobalah buat HTML seperti berikut, serta pelajari logika dan script-nya!
(simpan dgn nama **CSS9.htm**)

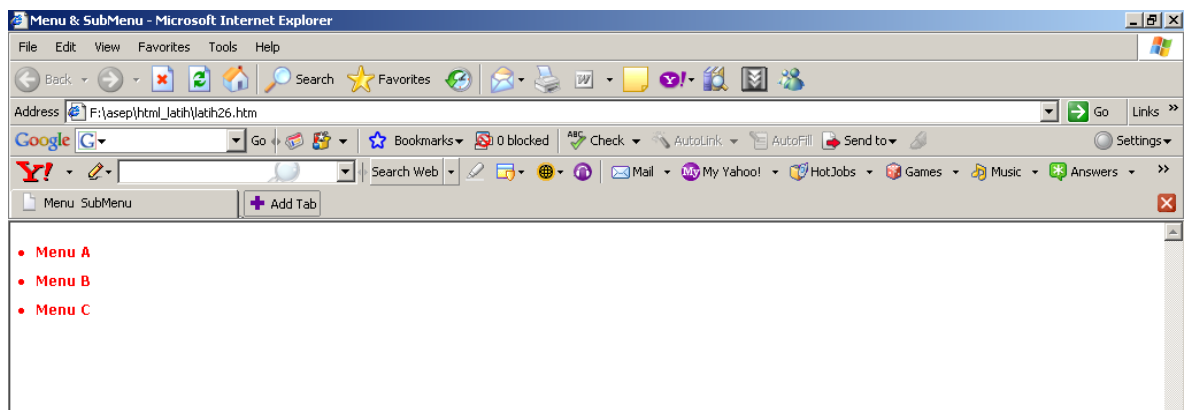
```
<html>
<head>
  <title>Menu & SubMenu</title>
  <style>
    li.menu {
      cursor: hand;
      margin: 6px;
    }
    li.submenu {
      cursor: crosshair;
    }
    ul.menu {
      margin-top: 0px;
      margin-left: 6px;
      font-family: Verdana, Arial, Helvetica, sans-serif;
      font-size: 12px;
      font-weight: bold;
      color: #FF0000;
    }
    ul.submenu {
      font-family: Verdana, Arial, Helvetica, sans-serif;
      font-size: 12px;
      font-weight: normal;
      color: #0066FF;
    }
  </style>
</head>

<script language="JavaScript">
  function buka()
  {
    var z;
    z = 1 + window.event.srcElement.sourceIndex;
    if (document.all(z).style.display=="none")
    {
      document.all(z).style.display=""
    }
    else
    {
      document.all(z).style.display="none"
    }
  }
</script>
```

bersambung ...

lanjutan *css9.htm* :

```
<body>
  <ul class="menu">
    <li class="menu" onClick="buka()">Menu A
      <div style="display: none">
        <ul class="submenu">
          <li class="submenu">Menu A Submenu 1</li>
          <li class="submenu">Menu A Submenu 2</li>
          <li class="submenu">Menu A Submenu 3</li>
        </ul>
      </div>
    </li>
    <li class="menu" onClick="buka()">Menu B
      <div style="display: none">
        <ul class="submenu">
          <li class="submenu">Menu B Submenu 1</li>
          <li class="submenu">Menu B Submenu 2</li>
          <li class="submenu">Menu B Submenu 3</li>
        </ul>
      </div>
    </li>
    <li class="menu" onClick="buka()">Menu C
      <div style="display: none">
        <ul class="submenu">
          <li class="submenu">Menu C Submenu 1</li>
          <li class="submenu">Menu C Submenu 2</li>
          <li class="submenu">Menu C Submenu 3</li>
        </ul>
      </div>
    </li>
  </ul>
</body>
</html>
```



Cobalah buat HTML seperti berikut, serta pelajari logika dan script-nya!

(simpan dgn nama *CSS10.htm*)

Bandingkan hasilnya dengan file css9.htm !!!

```
<html>
<head>
  <title>Menu & Submenu</title>
</head>
<style>
  span.menu {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 16px;
    color: #FF0000;
    font-weight: bold;
    cursor: hand;
  }
  span.submenu {
    font-weight: normal;
    color: #0000FF;
    font-size: 10px;
    font-family: Arial, Helvetica, sans-serif;
    cursor: crosshair;
  }
</style>
<script language="JavaScript">
  function buka()
  {
    var z;
    z=window.event.srcElement.id;
    if (z=="menuA")
    {
      if (submenuA.style.display=="none")
      {
        submenuB.style.display="none";
        submenuA.style.display="";
      }
      else
      {
        submenuA.style.display="none"
      }
    }
  }
</script>
```

bersambung ...

lanjutan *css10.htm* :

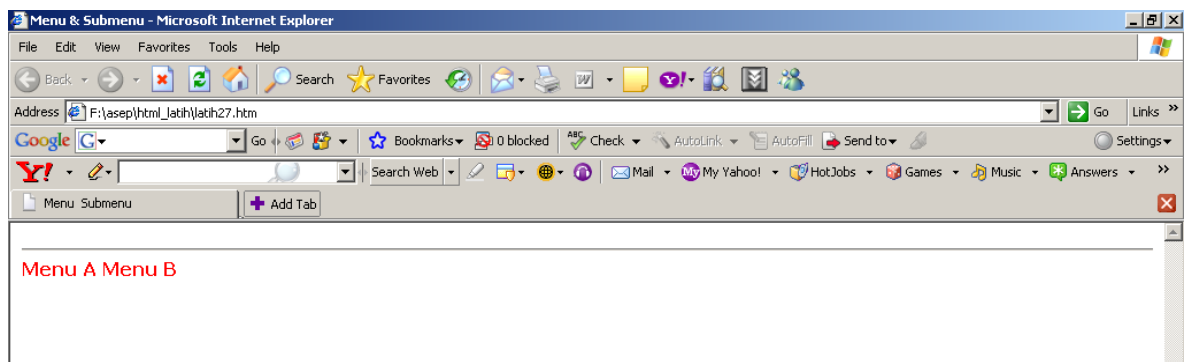
```

        if (z=="menuB")
        {
            if (submenuB.style.display=="none")
            {
                submenuA.style.display="none";

                submenuB.style.display="";
            }
            else
            {
                submenuB.style.display="none"
            }
        }
    }
</script>

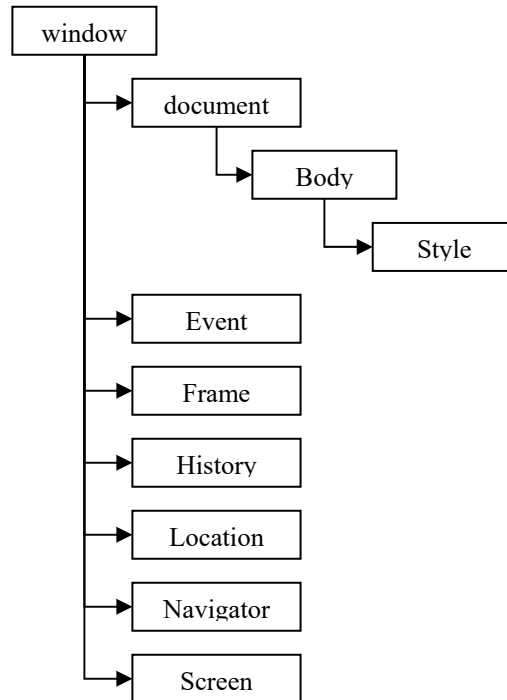
<body>
    <hr>
    <span id="menuA" class="menu" onClick="buka()">Menu A </span>
    <span id="submenuA" class="submenu" style="display: none">
        <span>| Menu A Submenu 1 |</span>
        <span>| Menu A Submenu 2 |</span>
        <span>| Menu A Submenu 3 |</span>
    </span>
    <span id="submenuB" class="submenu" style="display: none">
        <span>| Menu B Submenu 1 |</span>
        <span>| Menu B Submenu 2 |</span>
        <span>| Menu B Submenu 3 |</span>
    </span>
    <span id="menuB" class="menu" onClick="buka()">Menu B </span>
</body>
</html>

```



PENGUNAAN OBJEK

Penggunaan objek model sangat mudah asalkan mengikuti hierarki yang ada. Secara garis besar hierarki objek model pada Internet Explorer dapat digambarkan pada ilustrasi berikut:



Objek window adalah objek tertinggi, di dalam objek **window** terdapat suatu objek **document**. Objek *document* ini adalah objek yang cukup penting karena secara tidak langsung melambangkan badan suatu browser.

Setelah mengetahui hirarki dari objek ini, maka untuk menggunakan objek model dalam **script** dengan mudah tinggal *disusun berdasarkan hirarki tersebut dan dipisah oleh tanda titik*, lihat contoh berikut:

`window.document.write "halo ini percobaan"`

write adalah method yang ada pada objek document.

Contoh penggunaan objek model pada Script :

```
<HTML>
<Script Language="JavaScript">
function KeTekan()
{
    document.write "Halo Kawan"
}
</Script>
<BODY>
    <Form name="Form1">
        <Input Type="button" Name="Button1" value="Tekan"
            OnClick="KeTekan()">
    </Form>
</BODY>
</HTML>
```

Penulisan objek window dapat diabaikan seperti contoh di atas.

PENGUNAAN OBJEK WINDOW

Pada bagian ini akan dibahas beberapa objek window yang sering digunakan, yaitu:

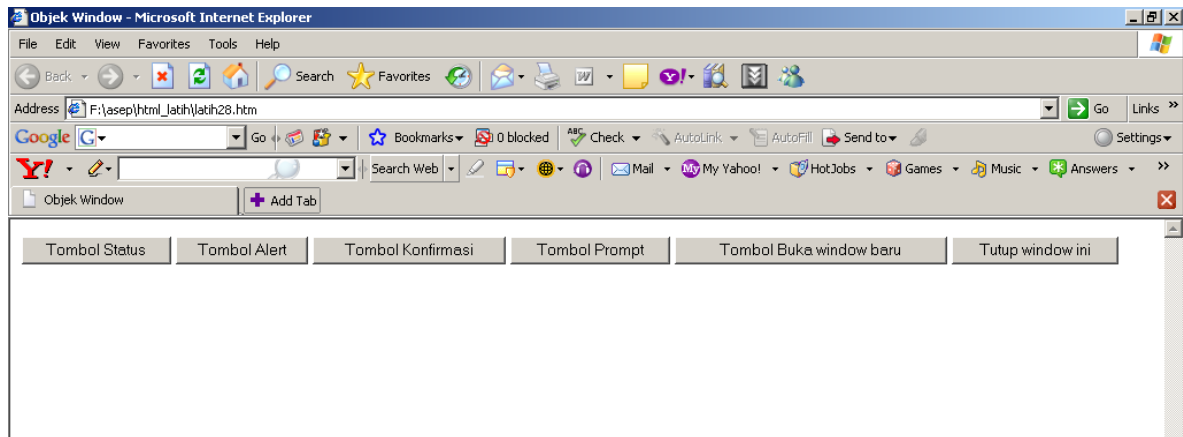
- Property **status**
- Method **alert()**
- Method **confirm()**
- Method **prompt()**
- Method **open()**
- Method **close()**

Untuk mencoba method dan property tersebut, coba listing HTML berikut:

```
<html>
<head>
  <title>Objek Window</title>
</head>

<script language="JavaScript">
function bawah()
{
    window.status="Latihan 2";
}
function perhatian()
{
    window.alert("Tombol Alert di tekan");
}
function konfirmasi()
{
    window.confirm("Tombol konfirmasi di tekan");
}
function tampil()
{
    var isian;
    isian = window.prompt("masukkan data","isi");
    isian = "Data yang anda masukkan = " + isian;
    document.write(isian);
}
function buka()
{
    win=window.open("Lat1.htm","Latihan1");
}
function tutup()
{
    window.close();
}
</script>

<body>
  <form action="" method="get">
    <input name="status" type="button" onClick="bawah()"
      value="Tombol Status">
    <input name="Alert" type="button" onClick="perhatian()"
      value="Tombol Alert">
    <input name="confirm" type="button"
      onClick="konfirmasi()" value="Tombol Konfirmasi">
    <input name="prompt" type="button" onClick="tampil()"
      value="Tombol Prompt">
    <input name="open" type="button" onClick="buka()"
      value="Tombol Buka window baru">
    <input name="close" type="button" onClick="tutup()"
      value="Tutup window ini">
  </form>
</body>
</html>
```

PENGUNAAN OBJEK DOCUMENT

Pada bagian ini akan dibahas beberapa objek document yang unik, yaitu:

- Property **title**
- Property **location**
- Method **write()**

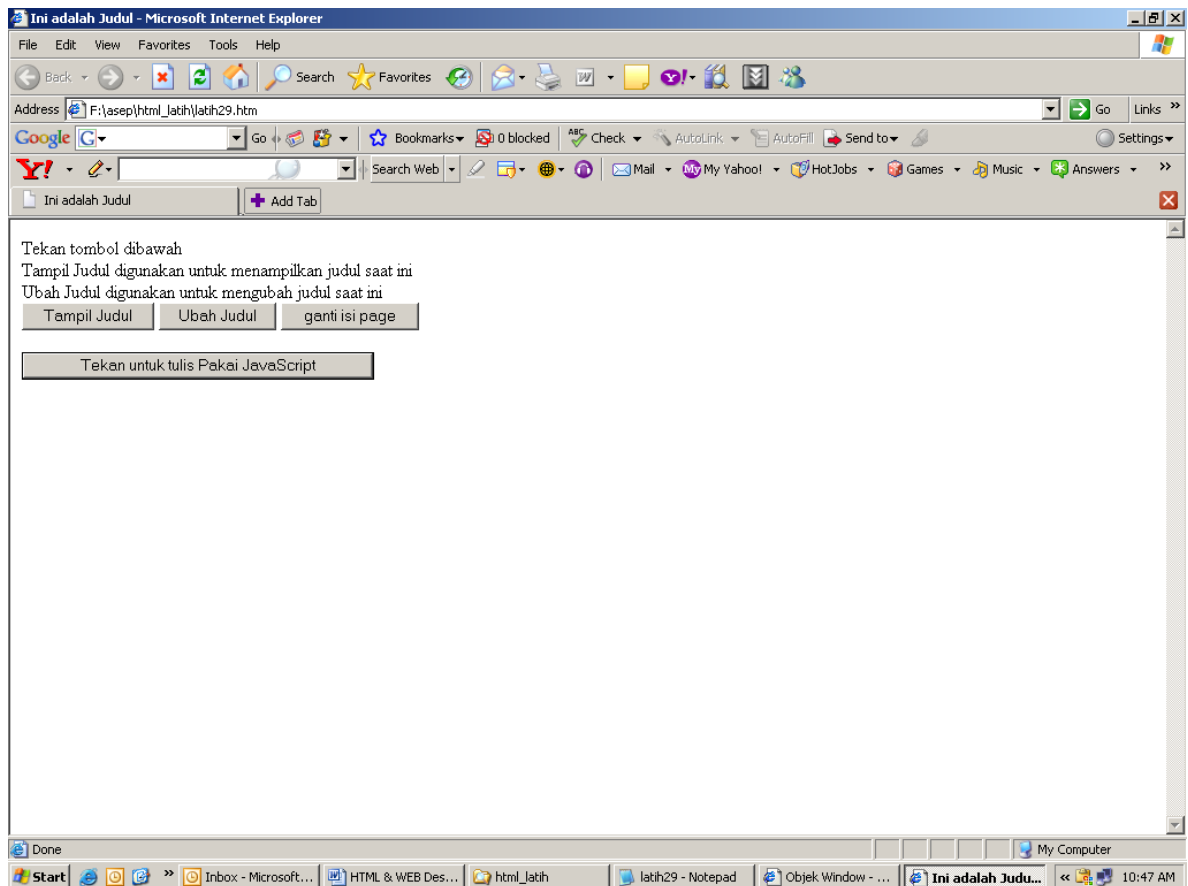
Untuk mencoba method dan property tersebut, coba contoh – contoh berikut:

```
<HTML>
<HEAD>
  <TITLE>Ini adalah Judul</TITLE>
</HEAD>
<Script language="JavaScript">
function tampil()
{
    alert (document.title)
}
function ubah()
{
    document.title='Ini judul yang baru'
}
</Script>

<BODY>
  Tekan tombol dibawah<BR>
  Tampil Judul digunakan untuk menampilkan judul saat ini<BR>
  Ubah Judul digunakan untuk mengubah judul saat ini<BR>
  <Input type="button" onclick="tampil()" value="Tampil Judul">
  <Input type="button" onclick="ubah()" value="Ubah Judul">
</BODY>
</HTML>
```

```
<html>
<head>
  <title>location</title>
</head>
<script language="JavaScript">
function ganti()
{
    document.location="lat1.htm"
}
</script>
<body>
  <input name="ganti" type="button" value="ganti isi page"
    onClick="ganti()">
</body>
</html>
```

```
<html>
<script language="JavaScript">
function jvtulis()
{
    window.document.write("Hello");
}
</script>
<body>
<form name="form1">
    <input name="btntulis1" type="button" onClick="jvtulis();"
        value="Tekan untuk tulis Pakai JavaScript">
</form>
</body>
</html>
```



MYSQL DAN PHP

MySQL adalah suatu database, dalam database pada umumnya informasi tersimpan dalam tabel – tabel yang secara logika merupakan struktur dua dimensi terdiri dari baris (*row* atau *record*) dan kolom(*column* atau *field*). Sedangkan dalam sebuah *database* dapat terdiri dari beberapa *table*.

Beberapa tipe data pada database MySQL yang sering dipakai:

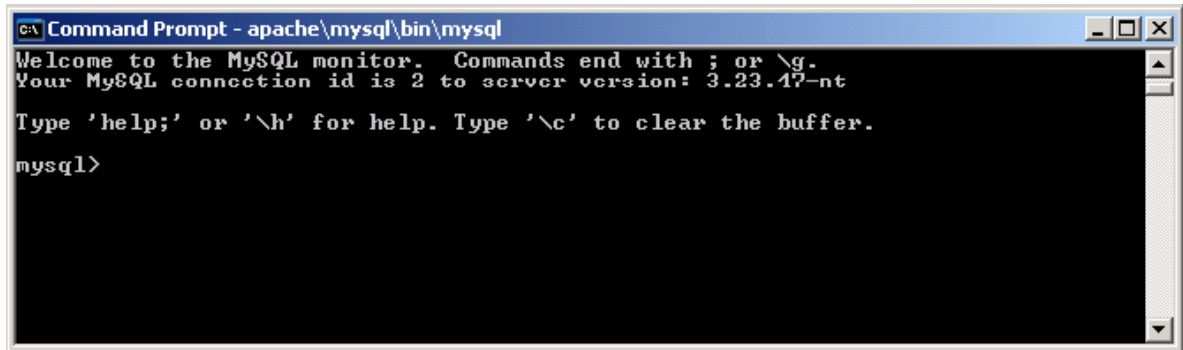
Tipe data	Keterangan
INT(M) [UNSIGNED]	-2147483648 s/d 2147483647 signed 0 s/d 4294967295 unsigned
MEDIUMINT	-8388608 s/d 8388607 signed 0 s/d 16777215 unsigned
SMALLINT	-32768 s/d 32767 signed 0 s/d 65535 unsigned
TINYINT	-128 s/d 127 signed 0 s/d 255 unsigned
FLOAT(M,D)	Angka pecahan
DATE	Tanggal Format : YYYY-MM-DD
DATETIME	Tanggal dan Waktu Format : YYYY-MM-DD HH:MM:SS
CHAR(M)	String dengan panjang tetap sesuai dengan yang ditentukan. Panjangnya 1-255 karakter
VARCHAR(M)	String dengan panjang yang berubah-ubah sesuai dengan yang disimpan saat itu. Panjangnya 1 – 255 karakter
BLOB	Teks dengan panjang maksimum 65535 karakter
LONGBLOB	Teks dengan panjang maksimum 4294967295 karakter

MEMBUAT DATABASE

Sebelum kita dapat membuat database kita harus masuk dahulu ke dalam MySQL. Untuk masuk ke dalam program MySQL pada prompt jalankan perintah berikut ini:

C:\> mysql (Enter)

Kemudian akan masuk kedalam MySQL seperti tampilan dibawah ini:



```
Command Prompt - apache\mysql\bin\mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 3.23.17-nt
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>
```

Cara untuk membuat sebuah database baru adalah dengan perintah:

create database namadatabase;

Setiap perintah SQL harus diakhiri dengan tanda titik-koma “;” .

Contoh:

create database dbprivat;

Untuk membuka sebuah database gunakan perintah berikut ini:

use namadatabase;

Contoh:

use dbprivat;

MEMBUAT TABLE

Sebelum membuat suatu tabel, Kita harus membuka database yang akan menampung tabel tersebut dengan menggunakan perintah **use** seperti di atas.

Perintah untuk membuat tabel baru adalah:

create table namatabel

(

Nama – nama field beserta type, ukuran dan spesifikasinya

);

Contoh:

Misalkan kita ingin menyimpan data anggota yaitu: nomor, nama, email, alamat, kota.

Sedangkan strukturnya seperti tabel dibawah ini:

Kolom/Field	Tipe data	Keterangan
nim	char(8) not null primary key	teks dengan panjang maksimal 8 karakter, sebagai primary key, tidak boleh kosong
nama	varchar (40) not null	teks dengan panjang maksimal 40 karakter, tidak boleh kosong
jur	varchar (30) not null	teks dengan panjang maksimal 30 karakter, tidak boleh kosong
alamat	varchar (80) not null	teks dengan panjang maksimal 80 karakter, tidak boleh kosong
kota	varchar (20) not null	teks dengan panjang maksimal 20 karakter, tidak boleh kosong

Perintah MySQL untuk membuat tabel seperti diatas adalah:

```
create table anggota(
nim char(6) not null primary key,
nama varchar(40) not null,
jur varchar(30) not null,
alamat varchar(80) not null,
kota varchar(20) not null
);
```

Setelah menjalankan perintah di atas, maka sekarang dalam database dbprivat telah terdapat tabel dengan nama anggota.

Untuk melihat tabel – table yang terdapat dalam database dbprivat, ketik perintah berikut:

```
show tables;
```

MENGISI DATA KE TABEL

Untuk memasukkan sebuah baris (record) kedalam tabel MySQL adalah sebagai berikut:

```
insert into namatabel values(kolom1, kolom2, kolom3,...);
```

contoh:

Misalkan kita ingin mengisi tabel anggota dengan data berikut:

NIM	Nama	Jurusan	Alamat	Kota
080232002	Reni Anggraini	Sistem Informasi	Jl. Raya Pondok Gede 74	Bekasi
080232003	Reni Harlina	Sistem Informasi	Jl. Hayam Wuruk 81	Jakarta
080231004	Agus	Manajemen Informatika	Jl. Jati Asih II/182	Bekasi
080231005	Bayu	Manajemen Informatika	Jl. Pemuda 19	Depok
080231006	Rizal	Manajemen Informatika	Jl. Karang tengah 5	Bekasi
080232004	Paula	Sistem Informasi	Jl. Jatiwaringin A-10	Bekasi
080232005	Anita	Sistem Informasi	Jl. Teuku Umar 45	Jakarta
080232006	Yusuf	Sistem Informasi	Jl. Rajawali 78	Jakarta
080232007	Ali	Sistem Informasi	Jl. Hasanudin 3	Jakarta
080232008	Azy	Sistem Informasi	Jl. Kalimantan Raya 9	Jakarta
080232009	Latif	Sistem Informasi	Jl. Merak 171	Jakarta
080132005	Suprizal	Sistem Informasi	Jl. Sudirman 12	Jakarta

Ketik perintah berikut memasukkan sebuah record ke dalam tabel anggota:

```
insert into anggota values('080232002', 'Reni Anggraini', 'Sistem Informasi', 'Jl. Raya Pondok Gede 74', 'Bekasi');
```

MENAMPILKAN ISI TABEL

Isi tabel dapat ditampilkan dengan menggunakan perintah **SELECT**, cara penulisan perintah SELECT adalah:

```
select namakolom from namatable;
```

Contoh:

- Untuk menampilkan kolom (field) nim dan nama pada tabel anggota

```
select nim, nama from anggota;
```

- Untuk menampilkan semua kolom(field) pada tabel anggota

```
select * from anggota;
```

- Untuk menampilkan semua kolom pada tabel anggota yang berada pada kota Bekasi

```
select * from anggota where kota='Bekasi';
```

- Untuk menampilkan semua kolom pada tabel anggota denganurut nama

```
select * from anggota order by nama;
```

- Untuk menghitung jumlah record pada tabel anggota
select count(*) from anggota;
- Untuk menampilkan kota pada tabel anggota
select kota from anggota;
- Untuk menampilkan kota dengan tidak menampilkan kota yang sama pada tabel anggota
select distinct kota from anggota;
- Untuk menampilkan nama dan jur yang mempunyai nama dengan awalan 'R'
select nama, jur from anggota where nama = 'R%';

MENGHAPUS RECORD

Untuk menghapus suatu record dengan kriteria tertentu digunakan perintah sebagai berikut:

delete from namatabel where kriteria;

Contoh:

Menghapus record dari tabel anggota yang bernomor '3'

delete from anggota where nomor='080232004';

MEMODIFIKASI RECORD

Untuk memodifikasi (merubah) isi record tertentu adalah dengan menggunakan perintah sebagai berikut:

update namatabel set kolom1=nilaibaru1, kolom2=nilaibaru2 ... where kriteria;

Contoh:

Merubah e-mail dari anggota yang bernomor 12 menjadi 'supri@yahoo.com' dalam tabel anggota.

update anggota set jur='Manajemen Informatika' where nomor='080232005';

STRUKTUR KONTROL PADA PHP

IF

Konstruksi IF digunakan untuk melakukan eksekusi suatu statement secara bersyarat. Cara penulisannya adalah sebagai berikut:

if (syarat)

{ statement }

atau:

if (syarat)

{ statement }

else

{ statement lain }

atau:

if (syarat pertama)

{ statement pertama }

elseif (syarat kedua)

{ statement kedua }

else

{ statement lain }

File contoh5.php:

```
<?php
$a=4; $b=9;
if ($a>$b)
{
    echo("a lebih besar dari pada b");
}
elseif ($a<$b)
{
    echo("a lebih kecil b");
}
Else
{
    echo("a sama dengan b");
}
?>
```

WHILE

Bentuk dasar dari statement While adalah sebagai berikut:

while (syarat) { statement }

Arti dari statemant While adalah memberikan perintah untuk menjalankan statement dibawahnya secara berulang-ulang, selama syaratnya terpenuhi.

File contoh6.php:

```
<?php
$a=1;
while ($a<10)
{
    echo($a);
    $a++;
}
?>
```

FOR

Cara penulisan statement FOR adalah sebagai berikut:

**for (ekspresi1; ekspresi2 ; ekspresi3)
statement**

ekspresi1 menunjukkan nilai awal untuk suatu variable ekspresi2 menunjukkan syarat yang harus terpenuhi untuk menjalankan statemant ekspresi3 menunjukkan pertambahan nilai untuk suatu variable

File contoh7.php:

```
<?php
for ($a=0;$a<10;$a++)
{
    echo("Nilai A = "); echo("$a"); echo("<br>");
}
?>
```

SWITCH

Statement SWITCH digunakan untuk membandingkan suatu variable dengan beberapanilai serta menjalankan statement tertentu jika nilai variable sama dengan nilai yang dibandingkan. Struktur Switch adalah sebagai berikut:

```

switch (variable) {
case nilai:
    statement;
    break;
case nilai:
    statemant;
    break;
case nilai:
    statement;
    break;
}

```

File contoh8.php:

```

<?php
$a=2;
switch($a) {
case 1:
    echo("Nilai variable a adalah satu");
    break;
case 2:
    echo("Nilai variable a adalah dua");
    break;
case 3:
    echo("Nilai variable a adalah tiga");
    break;
}
?>

```

REQUIRE

Statement Require digunakan untuk membaca nilai variable dan fungsi-fungsi dari sebuah file lain. Cara penulisan statement Require adalah:

require(namafile);

Statement Require ini tidak dapat dimasukkan didalam suatu struktur looping misalnya while atau for. Karena hanya memperbolehkan pemanggilan file yang sama tersebut hanya sekali saja.

File contoh9.php:

```

<?php
$a="Saya sedang belajar PHP";
function tulistebal($teks)
{
    echo("<b>$teks</b>");
}
?>

```

File contoh10.php:

```
<?php
require("contoh9.php");
tulistebal("Ini adalah tulisan tebal");
echo("<br>");
echo($a);
?>
```

INCLUDE

Statement Include akan menyertakan isi suatu file tertentu. Include dapat diletakkan didalam suatu looping misalkan dalam statement for atau while.

File contoh11.php:

```
<?php
echo("-----<br>");
echo("PHP adalah bahasa scripting<br>");
echo("-----<br>");
echo("<br>");
?>
```

File contoh12.php:

```
<?php
for ($b=1; $b<5; $b++)
{
    include("contoh11.php");
}
?>
```

MENGHUBUNGKAN PHP DENGAN MYSQL

Agar script PHP yang kita buat dapat berhubungan dengan database dari MySQL dapat menggunakan fungsi yg disimpan dalam file berikut ini:

koneksi.php:

```
<?php
function open_connection()
{
    $host="localhost";
    $username="root";
    $password="";
    $databasename="dbprivat";
    $link=mysql_connect($host,$username,$password) or die ("Hubungan dengan
    database gagal!");
    mysql_select_db($databasename,$link);
    return $link;
}
?>
```

Isi dari variabel \$host, \$username, \$password dan \$databasename dapat disesuaikan sesuai dengan setting pada MySQL server yang ada.

MENAMPILKAN ISI TABEL DENGAN PHP

Sebelum kita dapat mengolah atau menampilkan isi tabel dengan menggunakan script PHP, maka kita harus memanggil file koneksi.php di atas dari script PHP, dengan perintah:

```
require("koneksi.php");
```

Untuk menampilkan isi tabel anggota, coba pelajari script berikut:

Contoh:

Menampilkan data anggota yang telah dibuat dengan menggunakan script PHP.

File isiAnggota.php:

```
<?php
// ----- ambil isi dari file utama.php
require("utama.php");
// ----- hubungkan ke database
$link=open_connection();
// ----- menentukan nama tabel
$tablename="anggota";
// ----- perintah SQL dimasukkan ke dalam variable string
$sqlstr="select * from $tablename";
```

```
// ----- jalankan perintah SQL
$rsHasil = mysql_query ($sqlstr) or die ("Kesalahan pada perintah SQL!");
?>
<HTML>
<HEAD>
    <TITLE>DAFTAR ANGGOTA</TITLE>
</HEAD>
<BODY>
<?php
// ----- ambil isi masing-masing record
while ($row_rsHasil = mysql_fetch_assoc($rsHasil)) {
// ----- mengambil isi setiap kolom
$nim = $row_rsHasil ['nim'];
$nama = $row_rsHasil ['nama'];
$jur = $row_rsHasil ['jur'];
$alamat = $row_rsHasil ['alamat'];
$kota=$row_rsHasil ['kota'];
// ----- menampilkan di layar browser
Echo $nim . " " . $nama . " " . $jur . " " . $alamat . " " . $kota . "<br>";
}
?>
</BODY>
</HTML>
<?
// ----- putus hubungan dengan database
mysql_close($link);
?>
```

TABEL REFERENSI CSS2

Browser support: IE: Internet Explorer, M: Mac IE only, F: Firefox, N: Netscape.

W3C: The number in the "W3C" column indicates in which CSS recommendation the property is defined (CSS1 or CSS2).

BACKGROUND

Property	Description	Values	IE	F	N	W3C
background	A shorthand property for setting all 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>	4	1	6	1
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page	scroll fixed	4	1	6	1
background-color	Sets the background color of an element	<i>color-rgb</i> <i>color-hex</i> <i>color-name</i> transparent	4	1	4	1
background-image	Sets an image as the background	url(URL) none	4	1	4	1
background-position	Sets the starting position of a background image	top left top center	4	1	6	1

		top right center left center center center right bottom left bottom center bottom right x% y% xpos ypos				
background-repeat	Sets if/how a background image will be repeated	repeat repeat-x repeat-y no-repeat	4	1	4	1

BORDER

Property	Description	Values	IE	F	N	W3C
border	A shorthand property for setting all of the properties for the four borders in one declaration	<i>border-width</i> <i>border-style</i> <i>border-color</i>	4	1	4	1
border-bottom	A shorthand property for setting all of the properties for the bottom border in one declaration	<i>border-bottom-width</i> <i>border-style</i> <i>border-color</i>	4	1	6	1
border-bottom-color	Sets the color of the bottom border	<i>border-color</i>	4	1	6	2
border-bottom-style	Sets the style of the	<i>border-style</i>	4	1	6	2

	bottom border					
border-bottom-width	Sets the width of the bottom border	thin medium thick <i>length</i>	4	1	4	1
border-color	Sets the color of the four borders, can have from one to four colors	<i>color</i>	4	1	6	1
border-left	A shorthand property for setting all of the properties for the left border in one declaration	<i>border-left-width</i> <i>border-style</i> <i>border-color</i>	4	1	6	1
border-left-color	Sets the color of the left border	<i>border-color</i>	4	1	6	2
border-left-style	Sets the style of the left border	<i>border-style</i>	4	1	6	2
border-left-width	Sets the width of the left border	thin medium thick <i>length</i>	4	1	4	1
border-right	A shorthand property for setting all of the properties for the right border in one declaration	<i>border-right-width</i> <i>border-style</i> <i>border-color</i>	4	1	6	1
border-right-color	Sets the color of the right border	<i>border-color</i>	4	1	6	2

<u>border-right-style</u>	Sets the style of the right border	<i>border-style</i>	4	1	6	2
<u>border-right-width</u>	Sets the width of the right border	thin medium thick <i>length</i>	4	1	4	1
<u>border-style</u>	Sets the style of the four borders, can have from one to four styles	none hidden dotted dashed solid double groove ridge inset outset	4	1	6	1
<u>border-top</u>	A shorthand property for setting all of the properties for the top border in one declaration	<i>border-top-width</i> <i>border-style</i> <i>border-color</i>	4	1	6	1
<u>border-top-color</u>	Sets the color of the top border	<i>border-color</i>	4	1	6	2
<u>border-top-style</u>	Sets the style of the top border	<i>border-style</i>	4	1	6	2
<u>border-top-width</u>	Sets the width of the top border	thin medium thick	4	1	4	1

		<i>length</i>				
border-width	A shorthand property for setting the width of the four borders in one declaration, can have from one to four values	thin medium thick <i>length</i>	4	1	4	1

CLASSIFICATION

Property	Description	Values	IE	F	N	W3C
clear	Sets the sides of an element where other floating elements are not allowed	left right both none	4	1	4	1
cursor	Specifies the type of cursor to be displayed	<i>url</i> auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize	4	1	6	2

		text wait help				
display	Sets how/if an element is displayed	none inline block list-item run-in compact marker table inline-table table-row-group table-header-group table-footer-group table-row table-column-group table-column table-cell table-caption	4	1	4	1
float	Sets where an image or a text will appear in another element	left right none	4	1	4	1
position	Places an element in a static, relative, absolute or fixed position	static relative absolute fixed	4	1	4	2

visibility	Sets if an element should be visible or invisible	visible hidden collapse	4	1	6	2
----------------------------	---	-------------------------------	---	---	---	---

DIMENSION

Property	Description	Values	IE	F	N	W3C
height	Sets the height of an element	auto <i>length</i> %	4	1	6	1
line-height	Sets the distance between lines	normal <i>number</i> <i>length</i> %	4	1	4	1
max-height	Sets the maximum height of an element	none <i>length</i> %	-	1	6	2
max-width	Sets the maximum width of an element	none <i>length</i> %	-	1	6	2
min-height	Sets the minimum height of an element	<i>length</i> %	-	1	6	2
min-width	Sets the minimum width of an element	<i>length</i> %	-	1	6	2
width	Sets the width of an element	auto % <i>length</i>	4	1	4	1

FONT

Property	Description	Values	IE	F	N	W3C
font	A shorthand property for setting all of the properties for a font 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	4	1	4	1
font-family	A prioritized list of font family names and/or generic family names for an element	<i>family-name</i> <i>generic-family</i>	3	1	4	1
font-size	Sets the size of a font	xx-small x-small small medium large x-large xx-large smaller larger <i>length</i>	3	1	4	1

		%				
font-size-adjust	Specifies an aspect value for an element that will preserve the x-height of the first-choice font	none <i>number</i>	-	1	-	2
font-stretch	Condenses or expands the current font-family	normal wider narrower ultra-condensed extra-condensed condensed semi-condensed semi-expanded expanded extra-expanded ultra-expanded	-	-	-	2
font-style	Sets the style of the font	normal italic oblique	4	1	4	1
font-variant	Displays text in a small-caps font or a normal font	normal small-caps	4	1	6	1
font-weight	Sets the weight of a font	normal bold bolder lighter 100 200	4	1	4	1

	300				
	400				
	500				
	600				
	700				
	800				
	900				

GENERATED CONTENT

Property	Description	Values	IE	F	N	W3C
content	Generates content in a document. Used with the :before and :after pseudo-elements	<i>string</i> <i>url</i> <i>counter(name)</i> <i>counter(name, list-style-type)</i> <i>counters(name, string)</i> <i>counters(name, string, list-style-type)</i> <i>attr(X)</i> <i>open-quote</i> <i>close-quote</i> <i>no-open-quote</i> <i>no-close-quote</i>		1	6	2
counter-increment	Sets how much the counter increments on each occurrence of a selector	<i>none</i> <i>identifier number</i>				2

counter-reset	Sets the value the counter is set to on each occurrence of a selector	none <i>identifier number</i>				2
quotes	Sets the type of quotation marks	none <i>string string</i>	-	1	6	2

LIST AND MARKER

Property	Description	Values	IE	F	N	W3C
list-style	A shorthand property for setting all of the properties for a list in one declaration	<i>list-style-type</i> <i>list-style-position</i> <i>list-style-image</i>	4	1	6	1
list-style-image	Sets an image as the list-item marker	none <i>url</i>	4	1	6	1
list-style-position	Sets where the list-item marker is placed in the list	inside outside	4	1	6	1
list-style-type	Sets the type of the list-item marker	none disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-alpha upper-alpha lower-greek	4	1	4	1

		lower-latin upper-latin hebrew armenian georgian cjk-ideographic hiragana katakana hiragana-iroha katakana-iroha				
marker-offset		auto <i>length</i>		1	7	2

MARGIN

Property	Description	Values	IE	F	N	W3C
margin	A shorthand property for setting the margin properties in one declaration	<i>margin-top</i> <i>margin-right</i> <i>margin-bottom</i> <i>margin-left</i>	4	1	4	1
margin-bottom	Sets the bottom margin of an element	auto <i>length</i> %	4	1	4	1
margin-left	Sets the left margin of an element	auto <i>length</i> %	3	1	4	1
margin-right	Sets the right margin of an element	auto <i>length</i>	3	1	4	1

		%				
margin-top	Sets the top margin of an element	auto <i>length</i> %	3	1	4	1

OUTLINES

Property	Description	Values	IE	F	N	W3C
outline	A shorthand property for setting all the outline properties in one declaration	<i>outline-color</i> <i>outline-style</i> <i>outline-width</i>	-	1.5	-	2
outline-color	Sets the color of the outline around an element	<i>color</i> invert	-	1.5	-	2
outline-style	Sets the style of the outline around an element	none dotted dashed solid double groove ridge inset outset	-	1.5	-	2
outline-width	Sets the width of the outline around an element	thin medium thick <i>length</i>	-	1.5	-	2

PADDING

Property	Description	Values	IE	F	N	W3C
padding	A shorthand property for setting all of the padding properties in one declaration	<i>padding-top</i> <i>padding-right</i> <i>padding-bottom</i> <i>padding-left</i>	4	1	4	1
padding-bottom	Sets the bottom padding of an element	<i>length</i> <i>%</i>	4	1	4	1
padding-left	Sets the left padding of an element	<i>length</i> <i>%</i>	4	1	4	1
padding-right	Sets the right padding of an element	<i>length</i> <i>%</i>	4	1	4	1
padding-top	Sets the top padding of an element	<i>length</i> <i>%</i>	4	1	4	1

POSITIONING

Property	Description	Values	IE	F	N	W3C
bottom	Sets how far the bottom edge of an element is above/below the bottom edge of the parent element	<i>auto</i> <i>%</i> <i>length</i>	5	1	6	2
clip	Sets the shape of an element. The element is clipped into this shape, and displayed	<i>shape</i> <i>auto</i>	4	1	6	2

<u>left</u>	Sets how far the left edge of an element is to the right/left of the left edge of the parent element	auto % <i>length</i>	4	1	4	2
<u>overflow</u>	Sets what happens if the content of an element overflow its area	visible hidden scroll auto	4	1	6	2
<u>position</u>	Places an element in a static, relative, absolute or fixed position	static relative absolute fixed	4	1	4	2
<u>right</u>	Sets how far the right edge of an element is to the left/right of the right edge of the parent element	auto % <i>length</i>	5	1	6	2
<u>top</u>	Sets how far the top edge of an element is above/below the top edge of the parent element	auto % <i>length</i>	4	1	4	2
<u>vertical-align</u>	Sets the vertical alignment of an element	baseline sub super top text-top middle bottom	4	1	4	1

		text-bottom <i>length</i> %				
z-index	Sets the stack order of an element	auto <i>number</i>	4	1	6	2

TABLE

Property	Description	Values	IE	F	N	W3C
border-collapse	Sets whether the table borders are collapsed into a single border or detached as in standard HTML	collapse separate	5	1	7	2
border-spacing	Sets the distance that separates cell borders (only for the "separated borders" model)	<i>length length</i>	5M	1	6	2
caption-side	Sets the position of the table caption	top bottom left right	5M	1	6	2
empty-cells	Sets whether or not to show empty cells in a table (only for the "separated borders" model)	show hide	5M	1	6	2

table-layout	Sets the algorithm used to display the table cells, rows, and columns	auto fixed	5	1	6	2
------------------------------	---	---------------	---	---	---	---

TEXT

Property	Description	Values	IE	F	N	W3C
color	Sets the color of a text	<i>color</i>	3	1	4	1
direction	Sets the text direction	ltr rtl	6	1	6	2
line-height	Sets the distance between lines	normal <i>number</i> <i>length</i> %	4	1	4	1
letter-spacing	Increase or decrease the space between characters	normal <i>length</i>	4	1	6	1
text-align	Aligns the text in an element	left right center justify	4	1	4	1
text-decoration	Adds decoration to text	none underline overline line-through blink	4	1	4	1
text-indent	Indents the first line of text in an element	<i>length</i> %	4	1	4	1

text-shadow		none <i>color</i> <i>length</i>				
text-transform	Controls the letters in an element	none capitalize uppercase lowercase	4	1	4	1
unicode-bidi		normal embed bidi-override	5			2
white-space	Sets how white space inside an element is handled	normal pre nowrap	5	1	4	1
word-spacing	Increase or decrease the space between words	normal <i>length</i>	6	1	6	1

PSEUDO-CLASSES

Pseudo-class	Purpose	IE	F	N	W3C
:active	Adds special style to an activated element	4	1	8	1
:focus	Adds special style to an element while the element has focus	-	1.5	8	2
:hover	Adds special style to an element when you mouse over it	4	1	7	1
:link	Adds special style to an unvisited link	3	1	4	1
:visited	Adds special style to a visited link	3	1	4	1

:first-child	Adds special style to an element that is the first child of some other element	-	1	7	2
:lang	Allows the author to specify a language to use in a specified element	-	1	8	2

PSEUDO-ELEMENTS

Pseudo-element	Purpose	IE	F	N	W3C
:first-letter	Adds special style to the first letter of a text	5	1	8	1
:first-line	Adds special style to the first line of a text	5	1	8	1
:before	Inserts some content before an element		1.5	8	2
:after	Inserts some content after an element		1.5	8	2

DAFTAR PUSTAKA

1. Cameron Adams, Jina Bolton, David Johnson, Steve Smith, Jonathan Snook, The Art & Science of CSS, Sitepoint, 2007
2. <http://www.w3.org/TR/CSS21/page.html>
3. <http://css-discuss.incutio.com/>
4. <http://techrepublic.com.com/>
5. <http://www.webmonkey.com/>