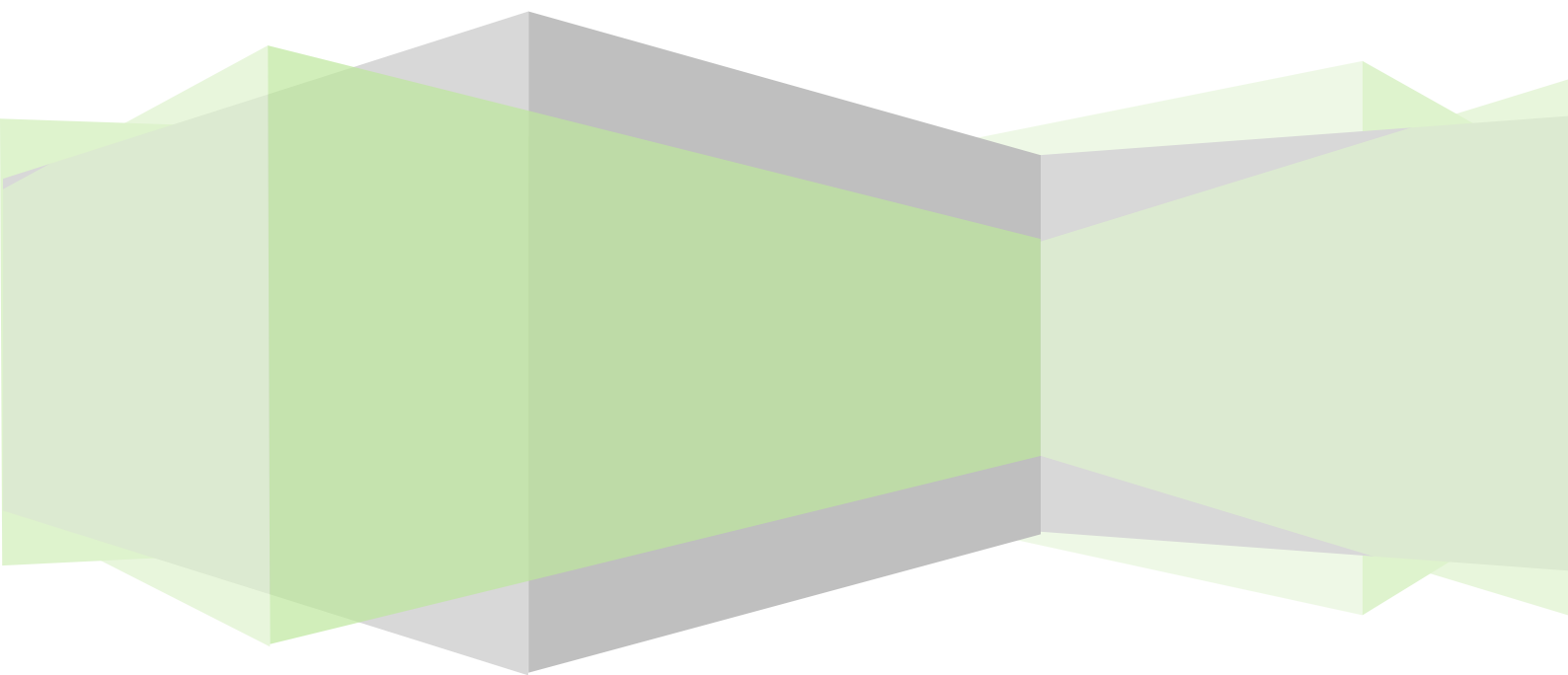


Diktat CSS

Cascading Style Sheet

L. Erawan



DAFTAR ISI

PENDAHULUAN.....	6
CSS Adalah Suatu Solusi	6
CSS Mengefisienkan Pekerjaan.....	6
SINTAKS	7
Memberi Komentar.....	7
CLASS DAN ID	8
ID CSS	8
CLASS CSS.....	8
Beberapa Cara Penerapan CSS	9
External Style Sheet.....	9
Internal Style Sheet.....	9
Inline Style Sheet	10
Multiple Style Sheet.....	10
Kaskada terhadap Multiple Styles.....	11
Urutan Kaskada.....	11
Properti Background CSS.....	12
Kelompok Properti Background	12
Background Color.....	12
Background Image	12
Background Repeat.....	12
Background Position	13
Background Attachment	13
Properti Shorthand Background.....	14
Pengaturan Teks.....	15
Properti Text.....	15
Text Color.....	15
Text Alignment	15
Text Decoration	15
Text Transformation	16
Text Indentation.....	16

CSS Font	17
CSS Font Families	17
Font Family	17
Font Style.....	17
CSS Link.....	18
CSS List.....	19
List.....	19
Berbagai Penanda Item List	19
Gambar Sebagai Penanda	20
Properti Shorthand List.....	20
CSS Table	22
Pengaturan Tabel dengan CSS.....	22
Border Tabel	22
Border Collapse.....	22
Tinggi dan Lebar Tabel.....	22
Perataan Teks Tabel.....	22
Padding Tabel	23
Warna Tabel.....	23
CSS Box Model.....	24
Properti Width dan Height	25
Masalah Kompatibilitas Browser.....	26
Properti-properti Border CSS.....	27
Border-style	27
Border-width.....	27
Border-color.....	28
Pengaturan Individual Sisi Border	28
Shorthand Properti Border.....	29
Outline.....	30
Properti Outline CSS	30
Margin	32
Individual Margin	32
Properti Shorthand Margin	32
Properti Margin.....	33
Pengelompokkan / Persarangan	34

Pengelompokkan.....	34
Persarangan.....	34
Properti Dimensi.....	35
Display dan Visibility.....	36
Menyembunyikan Elemen.....	36
Elemen Inline dan Block	36
Mengubah Tampilan Elemen Block Menjadi Inline dan Sebaliknya	36
Pengaturan Posisi Elemen.....	38
Static Positioning.....	38
Fixed Positioning.....	38
Relative Positioning	38
Absolute Positioning.....	39
Elemen yang Tumpang Tindih.....	39
Float (Mengambang)	41
Mematikan Pengambangan.....	43
Properti-properti Float CSS.....	44
Horizontal Align	45
Perataan Elemen Blok.....	45
Perataan Tengah Menggunakan Properti Margin.....	45
Perataan Kiri dan Kanan Menggunakan Properti Position.....	46
Mengatasi Permasalahan Kompatibilitas Browser	46
Perataan Kiri dan Kanan Menggunakan Properti Float.....	47
Permasalahan Kompatibilitas Browser.....	47
Pseudo-Classes	48
Pseudo-class <i>First-child</i>	49
Contoh-contoh lain:	49
Pseudo-class <i>Lang</i>	50
Pseudo-class Focus	51
Pseudo-elements.....	53
Pseudo-element First-line	53
Pseudo-element First-letter	53
Pseudo-element dan Kelas CSS	54
Multi Pseudo-element.....	55
Pseudo-element Before	55

Pseudo-element After.....	56
Batang Navigasi.....	58
Kode Standar Pengaturan Navigasi.....	58
Navigasi Vertikal.....	58
Navigasi Horisontal.....	58
Galeri Image.....	62
Image Opacity.....	64
Image Sprites.....	67
Efek Hover.....	69
Tipe-tipe Media.....	71
Tipe Media.....	71
Pengaturan dengan @media Rule.....	71
Selektor Attribute.....	73
Nilai Selektor Attribute.....	74
Selektor Attribute yang Bernilai Banyak.....	74
Mengatur Penampilan Form.....	76
Pengecualian CSS.....	78
Atribut Behavior Internet Explorer.....	78
Referensi.....	79

PENDAHULUAN

CSS ADALAH SUATU SOLUSI

CSS singkatan dari Cascading Stylesheet. CSS merupakan bahasa style yang bertugas mengatur bagaimana suatu elemen HTML ditampilkan dalam browser. Jika HTML berfungsi menyusun struktur konten halaman web, maka CSS berfungsi mendekorasi konten halaman web tersebut. CSS ditambahkan ke HTML versi 4.0 sebagai solusi terhadap permasalahan pencampuran kode-kode pembuatan konten dengan kode-kode pengaturan tampilan (standarisasi web).

HTML dibuat dengan tujuan awal untuk mendefinisikan konten sebuah dokumen, seperti:

```
<h1>This is a heading</h1>  
<p>This is a paragraph</p>
```

Sementara itu sebagian tag HTML didefinisikan memiliki atribut-atribut yang digunakan untuk pengaturan tampilan tag. Penggunaan atribut-atribut ini menimbulkan permasalahan dalam proses pengembangan website. Sebagai contoh, pengaturan warna teks menggunakan atribut *color* disetiap teks yang dibuat, membuat proses pengubahan warna teks menjadi suatu pekerjaan yang besar jika terdapat ribuan teks yang harus diedit. Hal ini mengakibatkan proses pengembangan akan menjadi lama dan berbiaya mahal.

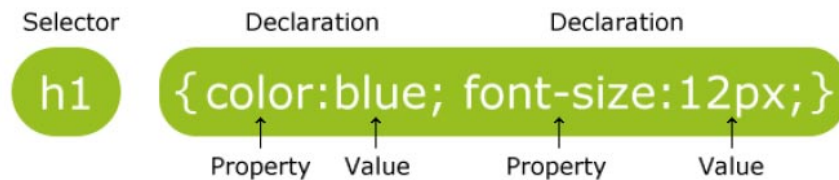
Pada versi HTML 4.0, semua perintah pengaturan tampilan dapat dihapus dari dokumen HTML dan disimpan secara terpisah dalam suatu file dengan ekstensi .css.

CSS MENGEFISIENKAN PEKERJAAN

Perintah pengaturan tampilan tersimpan dalam suatu file terpisah dari dokumen HTML. Perintah ini mengatur tampilan seluruh halaman web dalam sebuah website. Dengan demikian, mengubah tampilan seluruh halaman web dalam suatu website dapat dilakukan hanya dengan membuka sebuah file css.

SINTAKS

Sintaks perintah CSS sebagai berikut:



Keterangan gambar:

- **Selektor** biasanya berupa elemen HTML yang akan diatur formatnya
- Setiap **deklarasi** terdiri dari **properti** dan **nilai**. Properti adalah atribut style yang akan diubah.

Antara nama properti dan nilainya dipisahkan dengan simbol 'titik dua' dan setiap deklarasi diakhiri dengan simbol *semicolon* (titik koma). Sekelompok deklarasi diapit dengan simbol 'kurung kurawal'.

Contoh:

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

Untuk memudahkan membaca kode, biasanya penulisan perintah-perintah css disusun satu perintah satu baris seperti dibawah ini:

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

MEMBERI KOMENTAR

Komentar digunakan untuk memberikan catatan kepada kode-kode program dan akan membantu saat dilakukan pengeditan di kemudian hari. Dalam CSS, sebuah komentar diawali dengan simbol `/*` dan diakhiri dengan `*/`. Contoh:

```
/* Ini adalah komentar */  
P  
{  
Color:green;  
Font-family:arial;  
}
```

CLASS DAN ID

Selain menggunakan elemen HTML sebagai selektor, CSS mengizinkan pembuatan selektor sendiri yang disebut dengan *class* dan *id*.

ID CSS

Selektor id digunakan untuk mengatur tampilan sebuah elemen unik. Mendeklarasikan id menggunakan simbol '#'. Contoh penggunaan id:

```
#para1
{
  Text-align:center;
  Color:black;
}
```

Diatas dideklarasikan sebuah id dengan nama *para1* yang mengatur perataan paragraf ditengah dan warna teks hitam. Pemakaian id ini dalam sebuah dokumen HTML:

```
<p id="para1">
  Teks paragraf
</p>
```

CLASS CSS

Selektor class digunakan untuk menentukan gaya tampilan untuk sekelompok elemen. Tidak seperti id, selektor class dapat digunakan pada beberapa elemen HTML. Dengan demikian, class dapat digunakan untuk menentukan gaya tampilan sekelompok elemen HTML dari berbagai jenis dengan kelas yang sama.

Untuk mendeklarasikan kelas CSS menggunakan atribut HTML 'class' dan kelas CSS dideklarasikan menggunakan simbol 'titik'. Contoh:

```
.center
{
  Text-align:center;
}
```

Contoh penggunaan kelas CSS *center* diatas dalam dokumen HTML seperti ini:

```
<p class="center">
  Teks paragraf
</p>
<h1 class="center">Teks Heading 1</h1>
```

Kelas CSS juga dapat dibatasi hanya berlaku untuk elemen tertentu saja. Contoh:

```
p.center {
  text-align:center;}

```


BEBERAPA CARA PENERAPAN CSS

CSS bekerja dengan cara sebagai berikut: ketika browser membaca sebuah style sheet (perintah-perintah CSS), maka browser akan menampilkan dokumen HTML berdasarkan format yang ditentukan dalam style sheet tersebut.

Ada 3 cara penggunaan CSS dalam dokumen HTML:

- External style sheet
- Internal style sheet
- Inline style

EXTERNAL STYLE SHEET

External Style Sheet merupakan cara paling ideal menggunakan CSS. Dengan cara ini, penampilan seluruh website dapat diubah dengan hanya mengubah satu file saja (file CSS). Cara ini dilakukan dengan membuat sebuah file yang hanya berisi perintah-perintah CSS. Kemudian setiap halaman web harus dihubungkan dengan file style sheet ini menggunakan tag <link>. Tag ini ditulis di bagian HEAD. Contoh perintah menghubungkan dokumen HTML dengan file style sheet:

```
<head>
<link rel="stylesheet" type="text/css" href="styleku.css" />
</head>
```

File style sheet yang digunakan dalam contoh diatas adalah *styleku.css*. Perhatikan bahwa file style sheet disimpan dengan akhiran ".css". File style sheet dapat dibuat dengan sembarang teks editor, misalnya notepad++, atau dreamweaver. Isi file tidak boleh ada perintah HTML, hanya berisi perintah CSS. Contoh isi dari *styleku.css*, misalnya:

```
hr {color:blue;}
p {margin-left:20px;}
body {background-image:url("images/back04.gif");}
```

Sebagai catatan, jangan memberikan spasi diantara nilai properti dengan satuan unitnya seperti ini "margin-left:20 px". Antara nilai properti 20 dengan satuan px terdapat 1 spasi, ini tidak boleh karena tidak akan bekerja pada browser Firefox dan Opera, kecuali Internet Explorer.

INTERNAL STYLE SHEET

Cara internal style sheet sebaiknya digunakan apabila sebuah halaman web memiliki style yang berbeda dengan halaman web yang lain dalam sebuah situs web. Artinya, ini adalah penggunaan yang benar dari internal style sheet.

Seperti diketahui, setiap halaman web yang menyusun sebuah situs web terikat pada satu tema yang sama sehingga akan memiliki style yang sama. Misalnya, situs Udinus bertemakan pendidikan di Universitas Dian Nuswantoro yang setiap

halamannya ditampilkan dengan warna dominan biru yang sesuai dengan logo Universitas. Jika ada sebuah halaman web Udinus yang karena suatu alasan perlu berwarna dominan lain, disinilah saatnya menggunakan internal style sheet pada dokumen HTML dari halaman web tersebut.

Internal style sheet didefinisikan pada bagian HEAD dokumen menggunakan tag `<style>`, berikut ini contohnya:

```
<head>
<style type="text/css">
hr {color:blue;}
p {margin-left:20px;}
body {background-image:url("images/back04.gif");}
</style>
</head>
```

INLINE STYLE SHEET

Cara ini menghilangkan banyak kelebihan CSS dengan mencampurkan aspek konten dengan presentasinya. Gunakan cara ini apabila memang benar-benar mendesak.

Untuk menggunakan cara ini gunakan atribut *style* pada tag yang bersangkutan. Atribut *style* dapat berisi setiap properti CSS. Contoh dibawah ini menunjukkan cara mengubah warna dan margin kiri sebuah paragraf:

```
<p style="color:blue; margin-left:20px">Ini sebuah paragraf</p>
```

MULTIPLE STYLE SHEET

Dalam hal penggunaan CSS dalam suatu halaman web, adakalanya terdapat penggunaan lebih dari satu cara yang dijelaskan diatas. Suatu dokumen HTML dapat menggunakan dua atau tiga cara style sheet diatas sekaligus. Dan seringkali terdapat pengaturan beberapa properti untuk selektor yang sama dalam setiap cara yang digunakan. Maka dalam hal ini diterapkan prioritas pengaturan mana yang akan digunakan. Untuk lebih jelasnya simak contoh berikut ini.

Misalnya dalam suatu dokumen HTML menggunakan eksternal style sheet untuk mengatur selektor `h3` sebagai berikut:

```
h3
{
color:red;
text-align:left;
font-size:8pt;
}
```

Selain itu, juga menggunakan cara internal style sheet untuk mengatur selektor `h3` juga sebagai berikut:

```
h3
{
text-align:right;
font-size:20pt;
}
```

Maka aturan yang akan diterapkan untuk elemen h3 dalam dokumen HTML tersebut adalah:

```
color:red;
text-align:right;
font-size:20pt;
```

Pengaturan warna diturunkan dari eksternal style sheet sedangkan text-alignment dan font-size menggunakan aturan dalam internal style sheet.

KASKADA TERHADAP MULTIPLE STYLES

Seperti dijelaskan sebelumnya, CSS dapat diterapkan dengan tiga cara:

1. Didalam elemen HTML
2. Didalam bagian HEAD dari dokumen HTML
3. Dalam file CSS eksternal

Catatan: Dalam suatu dokumen juga dapat menggunakan lebih dari satu file CSS eksternal.

URUTAN KASKADA

Pengertian pengurutan kaskada disini adalah penerapan skala prioritas jika terdapat penggunaan lebih dari satu cara penerapan CSS dalam satu dokumen HTML. Pertanyaannya adalah style atau aturan CSS mana yang akan digunakan ketika terdapat lebih dari satu style yang dibuat untuk sebuah elemen HTML?

Dapat dikatakan disini bahwa semua style akan digabungkan kedalam suatu style sheet "virtual" yang baru berdasarkan aturan dibawah ini, dimana aturan keempat mempunyai prioritas tertinggi:

1. Default browser
2. Style sheet eksternal
3. Style sheet internal
4. Inline style sheet

Inline style sheet memiliki prioritas tertinggi, artinya aturan didalam inline stylesheet akan "mengalahkan" aturan didalam internal style sheet, didalam external style sheet, atau dalam default browser.

Sebagai catatan, jika link ke style sheet eksternal diletakkan setelah style sheet internal didalam bagian HEAD dokumen, aturan style sheet eksternal akan mengalahkan aturan style sheet internal.

PROPERTI BACKGROUND CSS

KELOMPOK PROPERTI BACKGROUND

Untuk mengatur latar belakang suatu elemen, CSS menyediakan kelompok properti background. Kelompok properti ini terdiri dari beberapa properti berikut ini:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

BACKGROUND COLOR

Properti ini mengatur latar belakang berupa warna dari sebuah elemen. Untuk mengatur latar belakang warna dari halaman web didefinisikan di selektor BODY:

```
body {background-color:#b0c4de;}
```

Warna background dapat ditentukan dengan menggunakan:

- **Nama warna**, misalnya red, blue, yellow
- **RGB**, misalnya rgb(255,0,0)
- **Hex**, misalnya #ff0000

Contoh penggunaan:

```
h1 {background-color:#6495ed;}
p { background-color:rgb(210,150,100);}
div { background-color:blue;}
```

BACKGROUND IMAGE

Properti background ini untuk menentukan gambar yang akan menjadi latar belakang suatu elemen. Secara default, gambar akan diulang sehingga akan menutup seluruh elemen. Contoh penggunaan:

```
body {background-image:url('paper.gif');}
```

BACKGROUND REPEAT

Secara default properti background image akan mengulang gambar secara horisontal dan vertikal sehingga akan menutup seluruh elemen. Terkadang gambar perlu diulang hanya secara horisontal atau vertikal saja. Untuk kebutuhan ini diperlukan properti background-repeat. Jika gambar ingin diulang secara horisontal saja, maka nilai properti yang digunakan adalah *repeat-x*. Contoh:

```
body
{
background-image:url('gradient2.png');
```

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

Jika ingin diulang secara vertikal maka nilai properti yang harus digunakan adalah *repeat-y*.

Gambar latar belakang juga dapat diatur agar tidak diulang baik secara vertikal maupun horisontal. Nilai properti yang digunakan adalah *no-repeat*. Contoh:

```
body
{
background-image:url('image_tree.png');
background-repeat:no-repeat;
}
```

BACKGROUND POSITION

Gambar latar belakang yang diset untuk tidak diulang secara vertikal maupun horisontal terkadang perlu diatur agar tidak terlalu mengganggu teks. Untuk keperluan ini perlu menggunakan properti untuk mengatur posisi gambar latar belakang yaitu *background-position*.

```
body
{
background-image:url('image_tree.png');
background-repeat:no-repeat;
background-position:right top;
}
```

Nilai yang dapat digunakan untuk properti ini adalah:

- left top
- left center
- left bottom
- right top
- right center
- right bottom
- center top
- center center
- center bottom
- *x% y%*
- *xpos ypos*
- inherit

BACKGROUND ATTACHMENT

Properti ini digunakan untuk mengatur gambar agar mengikuti pergerakan scrollbar atau tidak. Artinya, gambar dapat diatur agar ketika posisi halaman yang tampak dilayar diubah dengan scrollbar maka gambar dapat ikut bergerak atau tetap diam pada posisinya.

Properti bermanfaat ketika diinginkan agar gambar latar belakang yang diatur posisinya di kanan atas tetap pada posisinya ketika scrollbar menggulung halaman keatas atau kebawah.

Nilai yang dapat diberikan pada properti ini adalah `fixed`, `scroll`, atau `inherit`. Jika diberikan nilai properti `fixed`, maka gambar tidak akan ikut menggulung ketika halaman menggulung. Sebaliknya, nilai `scroll` akan membuat gambar ikut menggulung bersama halaman.

```
body
{
background-image:url('image_tree.png');
background-repeat:no-repeat;
background-position:right top;
background-attachment:fixed;
}
```

PROPERTI SHORTHAND BACKGROUND

Seperti telah dijelaskan di atas, dalam pengaturan latar belakang ini ada banyak properti yang dapat digunakan. Untuk mempersingkat kode dimungkinkan untuk menentukan seluruh properti tersebut dalam satu properti tunggal. Properti ini disebut properti shorthand. Properti shorthand untuk latar belakang ini sederhana, yaitu *background*. Contoh:

```
body {background:#ffff url('img_tree.png') no-repeat right top;}
```

Ketika menggunakan properti shorthand urutan nilai properti yang harus diperhatikan adalah:

- `background-color`
- `background-image`
- `background-repeat`
- `background-attachment`
- `background-position`

Tidak masalah jika hanya beberapa dari nilai properti tersebut yang digunakan, tetapi penulisannya tetap harus mengikuti urutan diatas.

PENGATURAN TEKS

PROPERTI TEXT

Teks yang ditampilkan pada sebuah halaman web dapat diatur penampilannya menggunakan kelompok aturan CSS untuk teks. Sebuah teks dapat diatur perataannya, warnanya, jarak antar huruf, dan sebagainya. Berikut ini penjelasan untuk properti-properti tersebut.

TEXT COLOR

Digunakan untuk mengatur warna teks, dapat ditentukan dengan:

- Nama warna, misalnya "red"
- Nilai RGB, misalnya "rgb(255,0,0)"
- Nilai hexadesimal, misalnya "#ff0000"

Warna default teks untuk sebuah halaman web dapat diatur menggunakan selektor BODY. Contoh:

```
body {color:blue;}
h1 {color:rgb(0,255,0);}
h2 {color:#ff0000;}
```

Sebagai catatan, agar sesuai dengan standar CSS W3C, penggunaan properti *color* harus diikuti dengan penggunaan properti *background-color*.

TEXT ALIGNMENT

Digunakan untuk mengatur perataan horisontal sebuah teks. Teks dapat diatur agar rata tengah, rata kiri, rata kanan, atau rata *justify*. Contoh:

```
h1 {text-align:center;}
p.date {text-align:right;}
p.main {text-align:justify;}
```

TEXT DECORATION

Digunakan untuk mengatur atau menghilangkan dekorasi dari teks. Biasanya digunakan untuk menghilangkan garis bawah pada suatu link untuk keperluan desain. Contoh:

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

Properti juga dapat digunakan untuk mendekorasi teks:

```
h1 {text-decoration:overline;}
h1 {text-decoration:line-through;}
h1 {text-decoration:underline;}
h1 {text-decoration:blink;}
```

Sebagai catatan, jangan mendekorasi sebuah teks bergaris bawah jika bukan sebuah link karena akan membuat bingung pengguna.

TEXT TRANSFORMATION

Digunakan untuk menentukan karakter huruf besar atau huruf kecil dari sebuah teks. Dapat digunakan misalnya untuk membuat teks menjadi berhuruf besar semua atau berhuruf besar diawal setiap kata. Contoh:

```
p.uppercase {text-transform:uppercase;}  
p.lowercase {text-transform:lowercase;}  
p.capitalize {text-transform:capitalize;}
```

TEXT INDENTATION

Digunakan untuk menentukan indentasi baris pertama dari suatu teks. Contoh:

```
p {text-indent:50px;}
```


CSS FONT

Properti-properti font CSS digunakan untuk mengatur jenis huruf, ketebalan, ukuran, dan gaya suatu teks. Terdapat dua kelompok besar jenis huruf, yaitu sans-serif dan serif. Pada layar komputer jenis huruf sans-serif lebih mudah dibaca daripada serif.

CSS FONT FAMILIES

Dalam CSS, terdapat dua jenis nama keluarga font:

- **generic family**, kelompok keluarga font dengan penampilan yang mirip, terdiri dari "Serif", "Sans-serif", dan "Monospace".
- **font family**, keluarga font tertentu. Untuk kelompok generic family "Serif" termasuk didalamnya "Times New Roman" dan "Georgia". Kelompok "Sans-serif" diantaranya "Arial" dan "Verdana". Sedangkan kelompok "Monospace" diantaranya "Courier New", "Lucida Console"

FONT FAMILY

Keluarga font atau jenis font dari suatu teks ditentukan dengan properti *font-family*. Penentuan jenis font dengan properti ini sebaiknya menggunakan beberapa jenis font. Jika browser tidak mendukung jenis font yang pertama, maka browser dapat mencoba jenis font kedua yang disediakan.

Strategi penerapannya dengan pertama menentukan jenis font yang kita inginkan lalu diakhiri dengan sebuah *generic family*, dengan tujuan memberi kesempatan kepada browser untuk mengambil jenis font yang sama didalam *generic family* jika tidak ada jenis font lain yang tersedia.

Sebagai catatan, jika nama jenis font lebih dari satu kata, harus dikurung dengan tanda kutip, seperti *font-family:"Times New Roman"*. Contoh:

```
p {font-family:"Times New Roman", Times, serif;}
```

FONT STYLE

Paling sering digunakan untuk membuat huruf miring (*italic*). Properti mempunyai tiga nilai:

- **normal**, teks ditampilkan secara normal
- **italic**, teks ditampilkan miring
- **oblique**, sangat mirip dengan italic namun kurang didukung

Contoh:

```
p.normal {font-style:normal;}  
p.italic {font-style:italic;}  
p.oblique {font-style:oblique;}
```

CSS LINK

Link dapat diatur dengan sembarang properti CSS, misalnya *color*, *background*, *font-family*. Khusus link dapat diatur penampilannya secara berbeda sesuai dengan kondisinya. Link dapat berada dalam empat kondisi yang berbeda:

- **a: link**, link yang normal, belum dikunjungi
- **a:visited**, link yang telah dikunjungi oleh pengguna
- **a:hover**, kondisi link saat mouse pengguna melintasinya
- **a:active**, kondisi link saat diklik

Contoh:

```
a:link {color:#ff0000; text-decoration:none;}
a:visited {color:#00ff00; text-decoration:none;}
a:hover {color:#0000ff; text-decoration:underline;}
a:active {color:#ff00ff; text-decoration:underline;}
```

Ketika hendak mengatur penampilan berbagai kondisi link tersebut diatas, aturan yang perlu diikuti adalah:

- a:hover harus diset setelah a:link dan a:visited
- a:active harus diset setelah a:hover

Sebagai catatan, properti *text-decoration* sering digunakan untuk menghilangkan garis bawah suatu link.

CSS LIST

LIST

Dalam HTML, terdapat dua jenis list:

- ***unordered list***, item list ditandai dengan bullet
- ***ordered list***, item list ditandai dengan angka atau huruf

Dengan CSS, list dapat diatur lebih lanjut dan gambar dapat digunakan sebagai penanda item list

BERBAGAI PENANDA ITEM LIST

Untuk mengatur penanda item list dapat menggunakan properti *list-style-type*.

Contoh:

```
ul.a {list-style-type:circle;}
ul.b {list-style-type:square;}
ol.c {list-style-type:upper-roman;}
ol.d {list-style-type:lower-alpha;}
```

Nilai penanda yang dapat digunakan untuk *unordered list* adalah:

- ***none***, tidak menggunakan penanda
- ***disc***, penanda default, berbentuk lingkaran berisi
- ***circle***, berbentuk lingkaran
- ***square***, berbentuk persegi empat

Sedangkan nilai penanda untuk *ordered list* adalah:

- ***armenian***, penanda berdasarkan penomoran armenian tradisional
- ***decimal***, penanda berupa angka
- ***decimal-leading-zero***, penanda berupa angka yang diawali dengan angka 0, misalnya 01, 02, 03, dan seterusnya
- ***georgian***, penanda berdasarkan penomoran georgian (an, ban, gan, dan seterusnya)
- ***lower-alpha***, penanda berupa a, b, c, d, dan seterusnya
- ***lower-greek***, penanda berupa alpha, betha, gamma, dan seterusnya
- ***lower-latin***, penanda berupa a, b, c, d, dan seterusnya
- ***lower-roman***, penanda berupa i, ii, iii, iv, dan seterusnya
- ***upper-alpha***, penanda berupa A, B, C, D, dan seterusnya
- ***upper-latin***, penanda berupa A, B, C, D, dan seterusnya
- ***upper-roman***, penanda berupa I, II, III, IV, dan seterusnya

Sebagai catatan, tidak ada versi Internet Explorer yang mendukung nilai properti "decimal-leading-zero", "lower-greek", "lower-latin", "upper-latin", "armenian", atau "georgian", kecuali DOCTYPE ditentukan.

GAMBAR SEBAGAI PENANDA

Untuk menggunakan gambar sebagai penanda, gunakan properti *list-style-image*.

Contoh:

```
ul
{list-style-image:url('sqpurple.gif');}
```

Contoh diatas tidak akan ditampilkan secara sama dalam semua browser. IE dan Opera akan menampilkan gambar penanda tersebut sedikit lebih tinggi dari Firefox, Chrome, dan Safari.

Untuk memecahkan masalah ini, dapat menggunakan solusi lintas browser berikut ini:

```
ul
{
list-style-type: none;
padding: 0px;
margin: 0px;
}
li
{
background-image: url(sqpurple.gif);
background-repeat: no-repeat;
background-position: 0px 5px;
padding-left: 14px;
}
```

Penjelasan kode:

- Elemen UL diatur properti *list-style-type* agar menghilangkan penanda. Kemudian atur *padding* dan *margin* ke opx (untuk kompatibilitas browser)
- Elemen LI diatur agar menggunakan gambar latar yang tidak diulang (*no-repeat*). Kemudian posisikan gambar di lokasi yang diinginkan (kekiri opx dan kebawah 5px). Terakhir, posisikan teks dalam list dengan *padding-left*.

Kode diatas akan menampilkan gambar dengan secara sama di semua browser.

PROPERTI SHORTHAND LIST

List juga memiliki properti shorthand, yaitu *list-style*. Contoh:

```
ul {
list-style:square url('sqpurple.gif');
}
```

Saat menggunakan properti ini, perhatikan urutan nilainya:

- list-style-type
- list-style-position
- list-style-image

Tabel Properti CSS List

Properti	Penjelasan	Nilai
list-style	mengatur seluruh properti list dalam satu deklarasi	<i>list-style-type</i> <i>list-style-position</i> <i>list-style-image</i> inherit
list-style-image	menentukan gambar sebagai penanda item list	URL none inherit
list-style-position	menentukan apakah penanda item list akan ditampilkan didalam (inside) atau diluar (outside) aliran konten.	inside outside inherit
list-style-type	menentukan jenis penanda item list	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

CSS TABLE

PENGATURAN TABEL DENGAN CSS

Elemen `table` merupakan salah satu elemen HTML yang memiliki peranan penting dalam menampilkan informasi secara jelas dan terstruktur. Dengan CSS, penampilan sebuah tabel HTML dapat ditingkatkan. Berikutnya akan dijelaskan beberapa properti elemen yang dapat digunakan untuk memperlindah tampilan sebuah tabel HTML.

BORDER TABEL

Untuk menentukan garis batas sebuah tabel menggunakan properti `border`. Contoh dibawah ini akan menampilkan garis batas tabel dengan warna hitam:

```
table, th, td {  
  border:1px solid black;  
}
```

Perhatikan bahwa tabel hasil pengaturan diatas memiliki border ganda. Hal ini karena baik elemen `table`, `th`, maupun `td` mempunyai border yang terpisah. Untuk menampilkan garis batas tunggal tabel gunakan properti `border-collapse`.

BORDER COLLAPSE

Properti ini berfungsi untuk mengatur apakah border tabel disatukan kedalam border tunggal atau terpisah. Contoh:

```
table {  
  border-collapse:collapse;  
}  
table, th, td {  
  border:1px solid black;  
}
```

TINGGI DAN LEBAR TABEL

Untuk mengatur tinggi dan lebar tabel gunakan properti `height` dan `width`. Contoh dibawah ini mengatur lebar tabel 100% dan tinggi elemen `th` 50px:

```
table {width:100%;}  
th {height:50px;}
```

PERATAAN TEKS TABEL

Teks dalam tabel dapat diratakan dengan properti `text-align` dan `vertical-align`. Nilai properti `text-align` adalah `left`, `right`, atau `center` sedangkan properti `vertical-align` dapat bernilai `top`, `middle`, atau `bottom`. Contoh:

```
td {text-align:right}  
th {height:50px; vertical-align:top;}
```

PADDING TABEL

Untuk mengatur ruang antara border dengan konten didalam suatu tabel, gunakan properti *padding* pada elemen td dan th. Contoh:

```
td {padding:15px;}
```

WARNA TABEL

Sebuah tabel dapat diatur pewarnaan border, teks, dan latar belakangnya. Contoh:

```
table, td, th
{
border:1px solid green;
}
th
{
background-color:green;
color:white;
}
```

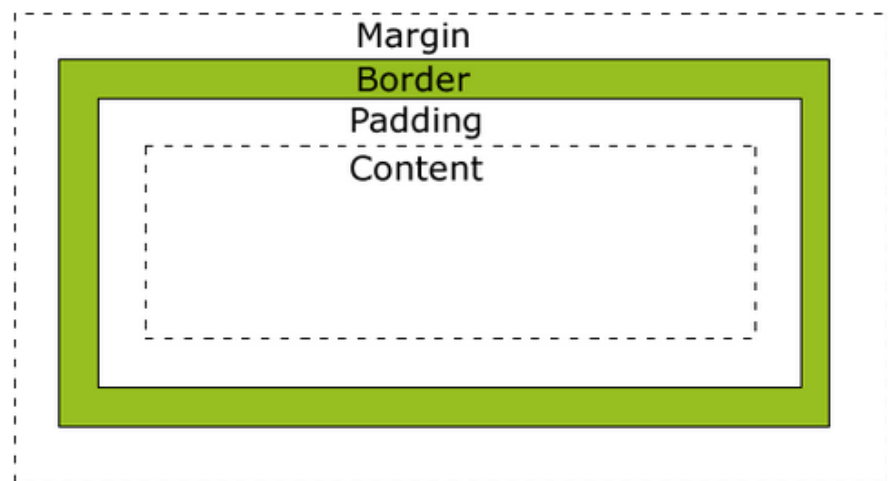
CSS BOX MODEL

Semua elemen HTML dapat dibayangkan sebagai sebuah kotak persegi panjang. Dalam CSS, istilah "box model" digunakan ketika membicarakan desain dan tata letak.

Model box CSS pada dasarnya adalah sebuah area berbentuk kotak persegi panjang yang mengitari elemen HTML, dan terdiri dari beberapa area yaitu: *margin*, *border*, *padding*, dan kontennya.

Model box memungkinkan kita untuk menempatkan *border* di sekitar elemen dan *space/ruang* dari elemen tersebut dalam hubungannya dengan elemen-elemen HTML lain.

Gambar di bawah ini mengilustrasikan model box tersebut:



Penjelasan masing-masing area:

Margin

Adalah area terluar dari suatu elemen HTML yang berbatasan dengan elemen HTML lain. Area ini bersifat transparan sehingga tidak dapat diberi warna latar belakang.

Border

Area border terletak diantara area *margin* dan *padding*. Warna border mengikuti warna dari latar belakang kotak.

Padding

Area disekitar *content* yang warna latar belakangnya mengikuti warna latar belakang kotak.

Content

Isi dari elemen HTML itu sendiri berupa teks atau gambar yang tampak pada layar browser

PROPERTI WIDTH DAN HEIGHT

Jika ingin mengatur properti *height* dan *width* dari suatu elemen HTML dengan benar pada semua browser (ada beberapa perbedaan diantara browser dalam memperlakukan area-area model box), semestinya perlu mengetahui terlebih dahulu bagaimana cara kerja dari model box.

Ketika properti *width* dan *height* dari suatu elemen diatur dengan CSS, sebenarnya yang diatur adalah tinggi dan lebar dari area konten (*content*) elemen tersebut. Jika ingin mengetahui ukuran sesungguhnya dari elemen maka perlu dihitung juga area *padding*, *border*, dan *margin*. Sebagai contoh menghitung ukuran sebenarnya suatu elemen, lihat kode CSS dibawah ini:

```
width:250px;
padding:10px;
border:5px solid gray;
margin:10px;
```

Jika suatu elemen diatur dengan kode CSS diatas, maka lebar dari elemen bukan 250 piksel tetapi 300 piksel. Angka 300 tersebut berasal dari lebar *width* ditambah dengan jumlah dari lebar area *padding*, *border*, dan *margin* dengan perhitungan:

lebar width 250px ditambah
20px, lebar *padding* kiri dan kanan,
10px, lebar *border* kiri dan kanan,
20px, lebar *margin* kiri dan kanan.

Jika telah mengetahui perhitungan diatas dan seumpama suatu ketika akan mengatur lebar suatu elemen dimana ruang yang tersedia hanya 250 piksel, maka contoh pengaturan yang benar sebagai berikut:

```
width:220px;
padding:10px;
border:5px solid gray;
margin:0px;
```

Jadi total lebar dari suatu elemen HTML mesti dihitung dengan kalkulasi sebagai berikut:

Total lebar elemen = width + left padding + right padding + left border + right border + left margin + right margin.

dan total tinggi suatu elemen dihitung dengan kalkulasi yang sama yaitu:

Total tinggi elemen = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

MASALAH KOMPATIBILITAS BROWSER

Jika contoh sebelumnya dijalankan menggunakan browser Internet Explorer, maka total lebar tidaklah tepat 250px. Hal ini karena Internet Explorer memasukkan lebar border dan padding suatu elemen kedalam *width* ketika properti ini digunakan. Untuk mengatasi hal ini pada dokumen HTML dapat ditambahkan deklarasi DOCTYPE.

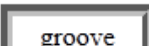
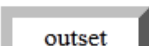
Jadi, pada contoh sebelumnya dapat ditambahkan deklarasi DOCTYPE sebagai berikut:

```
<!DOCTYPE html PUBLIC "-//W3C/DTD HTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<style type="text/css">
div.ex {
width:220px;
padding:10px;
border:5px solid gray;
margin:0px;
}
</style>
</head>
```

PROPERTI - PROPERTI BORDER CSS

BORDER - STYLE

Properti *border-style* digunakan untuk menentukan tipe border yang akan ditampilkan. Properti ini memiliki peranan yang penting karena properti border yang lain tidak akan bekerja sebelum properti ini ditentukan. Nilai-nilai dari properti *border-style*:

Nilai	Keterangan	Bentuk
<code>none</code>	tanpa border	
<code>dotted</code>	border berupa deretan bulatan kecil	
<code>dashed</code>	border berupa deretan garis putus-putus	
<code>solid</code>	border berupa garis tidak terputus	
<code>double</code>	border berupa garis dobel, lebar garis dobel selebar nilai properti <i>border-width</i>	
<code>groove</code>	border berupa 3D groove. Efeknya tergantung nilai properti <i>border-color</i>	
<code>ridge</code>	border berupa 3D ridge. Efeknya tergantung nilai properti <i>border-color</i>	
<code>inset</code>	border berupa 3D inset. Efeknya tergantung nilai properti <i>border-color</i>	
<code>outset</code>	border berupa 3D outset. Efeknya tergantung nilai properti <i>border-color</i>	

BORDER - WIDTH

Properti ini untuk mengatur lebar border. Nilai properti dalam satuan piksel. Dapat juga menggunakan nilai yang sudah tertentu, yaitu: *thin*, *medium*, atau *thick*. Penggunaan properti *border-width* harus didahului dengan penggunaan properti *border-style*, jika tidak maka tidak akan bekerja.

Contoh:

```
p.satu {
    border-style:solid;
    border-width:5px;
}
p.dua {
    border-style:solid;
```

```
border-width:medium;
}
```

BORDER – COLOR

Properti ini digunakan untuk mengatur warna border. Penentuan nilai warna dapat menggunakan beberapa cara:

- menggunakan nama warna, misalnya 'red'
- menggunakan kode warna RGB, misalnya 'rgb(255,0,0)'
- menggunakan satuan hex, misalnya '#ff0000'

Selain nilai-nilai, *border-color* juga dapat menggunakan nilai 'transparent'.

Penggunaan properti ini harus didahului dengan properti *border-style*, jika tidak maka tidak akan bekerja.

Contoh:

```
p.satu {
    border-style:solid;
    border-color:red;
}
p.dua {
    border-style:solid;
    border-color:#98bf21;
}
```

PENGATURAN INDIVIDUAL SISI BORDER

Border memiliki 4 sisi. Setiap sisi border dapat diatur dengan pengaturan yang berbeda-beda.

Contoh:

```
p {
    border-top-style:dotted;
    border-right-style:solid;
    border-bottom-style:dotted;
    border-left-style:solid;
}
```

Contoh diatas dapat juga diatur secara gabungan seperti ini:

```
p {
    border-style:dotted solid;
}
```

Dalam menggunakan pengaturan gabungan seperti contoh diatas, properti *border-style* dapat menggunakan 1 sampai dengan 4 nilai seperti dibawah ini:

- **border-style:dotted solid double dashed;**

Artinya:

- top border ditentukan *dotted*
 - right border ditentukan *solid*
 - bottom-border ditentukan *double*
 - left-border ditentukan *dashed*
- **border-style:dotted solid double;**
Artinya:
 - top border ditentukan *dotted*
 - right border dan left border ditentukan *solid*
 - bottom border ditentukan *double*
 - **border-style:dotted solid;**
Artinya:
 - top dan bottom border berupa *dotted*
 - left dan right border berupa *solid*
 - **border-style:dotted;**
Artinya keempat sisi border berupa *dotted*

Model pengaturan gabungan seperti diatas dapat dilakukan juga pada properti *border-width* dan *border-color*.

SHORTHAND PROPERTI BORDER

Jenis properti yang dimiliki oleh border berjumlah banyak. Dengan banyaknya properti border ini seringkali membuat repot dalam penerapan karena perlu menulis banyak kode. Untuk meringkas kode, semua properti dapat ditulis dalam satu properti yang disebut properti *shorthand*. Properti shorthand untuk border adalah 'border'.

Contoh:

```
border:5px solid red;
```

Dalam menggunakan properti 'border' ini, nilai properti perlu ditulis secara urut dengan urutan sebagai berikut:

- pertama, *border-width*
- kedua, *border-style*
- ketiga, *border-color*

Meskipun urutan sudah tertentu, nilai-nilai ketiga properti tersebut tidak harus ada, kecuali untuk nilai properti *border-style* yang tetap harus ada. Misalnya:

```
border: solid red;
```

atau

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

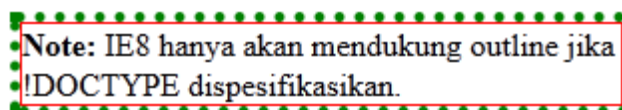
OUTLINE

Outline adalah garis yang mengelilingi elemen yang terletak setelah border. Fungsi garis outline ini untuk membuat kesan menonjol suatu elemen. Contoh penerapan garis outline ini dapat dilihat pada dokumen html berikut ini.

```
<!DOCTYPE html>
<html>
<head>
<style>
p
{
border:1px solid red;
outline:green dotted thick;
}
</style>
</head>

<body>
<p><b>Note:</b> IE8 hanya akan mendukung outline jika !DOCTYPE
dispesifikasikan.</p>
</body>
</html>
```

Hasilnya pada browser:



Note: IE8 hanya akan mendukung outline jika !DOCTYPE dispesifikasikan.

Properti outline ini berbeda dengan properti border. Outline bukan merupakan bagian dari dimensi elemen. Total lebar dan tinggi suatu elemen tidak terpengaruh oleh lebar outline.

PROPERTI OUTLINE CSS

Tabel berikut ini berisi semua properti outline yang dapat digunakan.

Properti	Deskripsi	Nilai	CSS
outline	mengatur semua properti outline dalam satu deklarasi	<i>outline-color</i> <i>outline-style</i> <i>outline-width</i> inherit	2
outline-color	mengatur warna outline	<i>color_name</i> <i>hex_number</i> <i>rgb_number</i> invert inherit	2
outline-style	mengatur style outline	none dotted dashed solid double groove ridge inset outset inherit	2
outline-width	mengatur lebar outline	thin medium thick <i>length</i> inherit	2

MARGIN

Properti margin digunakan untuk mengatur ruang disekitar elemen. Properti ini akan mengosongkan area disekitar elemen diluar border. Margin tidak memiliki warna latar belakang dan bersifat transparan. Top, bottom, left, dan right margin dapat diatur menggunakan properti yang terpisah. Properti shorthand margin juga dapat digunakan untuk mengatur semua margin sekaligus sekali perintah.

Nilai-nilai margin dapat berupa:

- **auto**, browser yang akan menghitung margin
- **length**, menentukan margin dengan satuan px, pt, cm, dan lain-lain. Defaultnya opx
- **%**, menentukan margin sebagai prosentase dari lebar elemen yang ditempatinya
- **inherit**, menentukan margin diturunkan dari elemen orang tuanya

Ketika menentukan nilai margin, nilai negatif dapat digunakan untuk membuat konten bertumpang tindah.

INDIVIDUAL MARGIN

Margin suatu elemen dapat diatur berbeda untuk setiap sisinya (atas, bawah, kiri, kanan).

Contoh:

```
margin-top: 100px;  
margin-bottom: 100px;  
margin-left: 100px;  
margin-right: 100px;
```

PROPERTI SHORTHAND MARGIN

Untuk meringkas kode, seluruh properti margin dapat diatur dalam satu properti yang disebut properti *shorthand*. Properti *shorthand* untuk margin adalah 'margin'.

Contoh:

```
margin:100px 50px;
```

Penggunaan properti *shorthand* margin ini dapat menggunakan satu sampai empat nilai:

- **margin:25px 50px 75px 100px;**

artinya:

```
margin atas diatur 25px  
margin kanan 50px  
margin bawah 75px  
margin kiri 100px
```


- **margin:25px 50px 75px;**

Artinya:

margin atas diatur 25px
margin kanan dan kiri 50px
margin bawah 75px

- **margin:25px 50px;**

Artinya:

margin atas dan bawah diatur 25px
margin kanan dan kiri 50px

- **margin:25px;**

Artinya:

seluruh margin diatur 25px

PROPERTI MARGIN

Property	Description
margin	properti <i>shorthand</i> untuk mengatur semua properti dalam satu deklarasi
margin-bottom	mengatur margin bawah
margin-left	mengatur margin kiri
margin-right	margin kanan
margin-top	margin atas

PENGELOMPOKKAN / PERSARANGAN

PENGELOMPOKKAN

Sering dijumpai beberapa elemen mempunyai gaya pengaturan yang sama. Misalnya berikut ini:

```
h1 {color:red}
h2 {color:red}
p {color:red}
```

Pengaturan terhadap elemen-elemen h1, h2, p tersebut karena sama dapat dikelompokkan menjadi satu seperti ini:

```
h1, h2, p {color:red}
```

Cara ini akan membuat kode lebih efisien. Ketika mengelompokkan pisahkan setiap elemen dengan koma.

PERSARANGAN

Pengaturan gaya untuk sebuah selektor dapat diatur didalam sebuah selektor yang lain. Ini disebut nesting atau persarangan. Lebih jelasnya lihat contoh berikut:

```
.iklan p
{
font-size:14px;
font-weight:bold;
color: lightblue;
}
```

maksud dari kode CSS diatas adalah elemen paragraf yang terletak didalam kelas iklan ditampilkan dengan pengaturan gaya tersebut, sedangkan elemen paragraf yang lain tidak. Jadi, hanya elemen paragraf dalam kelas iklan saja yang akan ditampilkan dengan gaya diatas. Agar lebih jelas lihat contoh penerapan pengaturan tersebut berikut ini:

```
<body>
<p>Paragraf ini ditampilkan biasa</p>
<div class="iklan">
<p>Paragraf yang ini ditampilkan dengan gaya pengaturan diatas, yaitu
ukuran huruf 14 piksel, tebal, dan warna teks biru muda</p>
</div>
</body>
```

Contoh diatas akan menampilkan paragraf pertama dengan gaya paragraf biasa, sedangkan paragraf kedua dengan gaya seperti diatur dalam kode CSS.

PROPERTI DIMENSI

CSS mempunyai properti-properti dimensi yang dapat digunakan untuk mengatur tinggi dan lebar dari suatu elemen. Berikut ini properti-properti tersebut:

Properti	Penjelasan	Nilai	CSS
Height	Mengatur tinggi elemen	Auto, length, %, inherit	1
Max-height	Mengatur tinggi maksimum elemen	None, length, %, inherit	2
Max-width	Mengatur lebar maksimum elemen	None, length, %, inherit	2
Min-height	Mengatur tinggi minimal elemen	Length, %, inherit	2
Min-width	Mengatur lebar minimal elemen	Length, %, inherit	2
Width	Mengatur lebar elemen	Auto, length, %, inherit	1

DISPLAY DAN VISIBILITY

Properti display akan menentukan bagaimana suatu elemen akan ditampilkan dan property visibility untuk menentukan suatu elemen ditampilkan atau disembunyikan.

MENYEMBUNYIKAN ELEMEN

Dapat dilakukan dengan menggunakan properti display yang diatur *none* atau properti visibility yang diatur *hidden*. Terdapat perbedaan hasil diantara keduanya. Jika menggunakan properti display, elemen tidak akan ditampilkan dan ruang yang sebelumnya digunakan oleh elemen tersebut dapat digunakan oleh elemen lain. Sedangkan penggunaan properti visibility akan menyebabkan elemen tidak akan ditampakkkan dan membuat ruang kosong yang tidak dapat digunakan untuk menampilkan elemen lain. Contoh:

```
h1.hidden {visibility:hidden}
```

atau

```
h1.hidden {display:none}
```

ELEMEN INLINE DAN BLOCK

Elemen terbagi 2 jenis, yaitu inline dan block. Elemen block tampil di halaman web dengan menggunakan seluruh lebar halaman yang tersedia, artinya elemen block akan menempati area halaman dari pojok kiri sampai kanan halaman pada baris terkait. Selain itu, elemen block juga mempunyai pergantian baris (line break) sebelum dan setelah elemen. Contoh dari elemen block:

```
<h1>, <p>, <div>
```

Sedangkan elemen inline akan menempati area halaman web selebar dirinya dan tidak menghasilkan pergantian baris. Contoh elemen inline:

```
<span>, <a>
```

MENGUBAH TAMPILAN ELEMEN BLOCK MENJADI INLINE DAN SEBALIKNYA

Mengubah elemen inline menjadi block elemen dan sebaliknya dapat bermanfaat untuk membuat halaman web mempunyai tampilan yang unik namun tetap sesuai dengan standar web. Berikut ini contoh cara mengubah tampilan elemen block menjadi inline:

```
li {display:inline;}
```

Sedangkan mengubah tampilan elemen inline menjadi tampilan block:

```
span {display:block;}
```

Pengubahan ini hanya pada penampilan saja, sedangkan jenis elemen tidak akan mengalami perubahan, artinya elemen inline yang diubah tampilannya menjadi elemen block akan tetap berjenis inline sehingga tetap tidak dapat menampung elemen block lain didalamnya.

PENGATURAN POSISI ELEMEN

Properti `position` CSS digunakan untuk mengatur penempatan elemen. Dengan properti ini, suatu elemen dapat diatur posisinya berada dibelakang elemen lain dan juga dapat mengatur penampilan bila konten suatu elemen terlalu besar.

Elemen dapat diatur posisinya dengan menggunakan properti `top`, `bottom`, `left`, dan `right`. Tetapi properti `position` harus diatur terlebih dahulu agar properti-properti tersebut bekerja. Cara kerja properti-properti tersebut akan berbeda tergantung pengaturan properti `position`.

Terdapat 4 metode penempatan posisi:

STATIC POSITIONING

Secara default elemen-elemen HTML ditempatkan secara statis. Elemen dengan posisi statis selalu ditempatkan menurut aliran normal halaman. Posisi elemen statis ini tidak terpengaruh oleh properti `top`, `bottom`, `left`, dan `right`.

FIXED POSITIONING

Elemen yang ditentukan *fixed* akan diposisikan relatif terhadap jendela browser. Elemen tidak akan bergeser bila jendela bergulir. Contoh:

```
p.posisi_tetap
{
  position:fixed;
  top:30px;
  right:5px;
}
```

RELATIVE POSITIONING

Elemen yang ditempatkan secara relatif akan diposisikan relatif terhadap posisi normalnya. Dengan posisi ini akan memungkinkan konten suatu elemen akan menindih konten elemen lain. Tetapi tempat elemen semula akan tetap dipertahankan dalam aliran normal. Artinya, meskipun posisi elemen tampak bergeser dari tempat semula, akan tetapi tempat tersebut tidak dapat ditempati oleh elemen lain karena masih tetap milik elemen yang bergeser. Contoh:

```
h2.pos_kiri
{
  position:relative;
  left:-20px;
}
h2.pos_right
```

```
{
  position:relative;
  left:20px;
}
```

Elemen dengan posisi relatif sering dipergunakan sebagai blok kontainer untuk elemen-elemen berposisi absolut.

ABSOLUTE POSITIONING

Elemen dengan posisi absolut ditempatkan relatif terhadap elemen orangtua pertama yang mempunyai posisi selain *static*. Jika tidak ada elemen orangtua, maka blok kontainernya adalah elemen <html>. Contoh:

```
h2
{
  position:absolute;
  left:100px;
  top:150px;
}
```

ELEMEN YANG TUMPANG TINDIH

Jika suatu elemen diposisikan keluar dari aliran normalnya, maka dia dapat menindih elemen yang lain. Terkait saling menumpuk ini, properti *z-index* dapat digunakan untuk mengatur urutan penumpukan sebuah elemen. Artinya, elemen dapat diatur diatas, atau dibawah elemen yang lain. Nilai properti *z-index* dapat negatif atau positif. Contoh:

```
img
{
  position:absolute;
  left:0px;
  top:0px;
  z-index:-1;
}
```

Elemen dengan nilai *z-index* lebih kecil akan berada dibawah elemen dengan nilai yang lebih besar. Jika dua elemen saling tumpang tindih dan tidak menggunakan properti *z-index*, maka elemen yang diposisikan terakhir akan ditumpuk diatas.

Daftar Lengkap Properti Penempatan CSS

Properti	Penjelasan	Nilai	CSS
bottom	Mengatur tepi batas bawah elemen	Auto, length, %, inherit	2
clip	Memotong elemen berposisi absolut	Shape, inherit	auto, 2

cursor	Menentukan tampilan kursor	jenis	<i>url</i> auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help	2
left	Mengatur tepi batas kiri elemen		Auto, length, % inherit	2
Overflow	Menentukan perlakuan jika konten lebih besar dari isi elemen		Auto, hidden, scroll, visible, inherit	2
Position	Menentukan jenis posisi elemen		Absolute, fixed, relative, static, inherit	2
Right	Mengatur tepi batas kanan elemen		Auto, length, % inherit	2
Top	Mengatur tepi batas atas elemen		Auto, length, % inherit	2
z-index	Menentukan urutan penumpukan elemen	urutan	Number, auto, inherit	2

FLOAT (MENGAMBANG)

Float berarti mengambang, keadaan ini dapat dibayangkan seperti gambar-gambar yang diletakkan diatas sebuah kertas tetapi tidak direkatkan sehingga dapat digeser kekiri dan kekanan.

Elemen mengambang secara horisontal, artinya elemen hanya dapat diambangkan kekiri atau kekanan, tidak keatas atau kebawah. Dengan properti float ini, suatu elemen dapat didorong kekiri atau kekanan. Sering digunakan untuk gambar, tetapi juga berguna untuk pengaturan tata letak halaman.

Elemen float dapat digeser kekiri atau kekanan sepanjang ruang kosong yang ada disebelah kiri atau kanannya. Elemen yang terletak setelah elemen float akan mengalir disekitar elemen float. Artinya elemen tersebut akan ditempatkan persis disisi kiri atau kanan elemen float. Jika sisa ruang sisi kiri atau kanan elemen float sudah habis, maka elemen lain akan ditempatkan di baris selanjutnya.

Jika sebuah gambar diambangkan ke kanan, maka teks yang mengikutinya akan diletakkan di sekitar kanannya.

Jika terdapat beberapa elemen float, maka elemen float satu akan diambangkan setelah elemen float lainnya selama masih ada ruang yang tersisa. Contoh:

```
<!DOCTYPE html>
<html>
<head>
<style>
.thumbnail
{
float:left;
width:110px;
height:90px;
margin:5px;
}
</style>
</head>

<body>
<h3>Image Gallery</h3>
<p>Try resizing the window to see what happens when the images does not
have enough room.</p>



```

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

Kode diatas akan menghasilkan tampilan:



Gambar-gambar yang ditampilkan akan mengalir kekanan, artinya akan diletakkan satu persatu kekanan sampai ruang sebelah kanan gambar habis dan gambar berikutnya akan mengalir ke baris selanjutnya mulai dari pojok kiri terus kekanan.

Jika lebar dari kelas thumbnail diubah dari 110 menjadi 165 dan tingginya diubah dari 90 menjadi 135 maka hasilnya akan menjadi:



Tampak bahwa di setiap barisnya jumlah gambar hanya 3 dari sebelumnya 5 gambar per baris karena ukuran gambar jadi lebih besar.

MEMATIKAN PENGAMBANGAN

Elemen yang terletak setelah elemen float akan dialirkan setelah elemen float tersebut. Untuk menghindari hal ini, dapat digunakan properti *clear*. Properti ini akan mengatur sisi mana dari elemen selain elemen float yang tidak dapat digunakan untuk pengambangan. Contoh:

```
<!DOCTYPE html>
<html>
<head>
<style>
.thumbnail
{
float:left;
width:110px;
height:90px;
margin:5px;
}
.text_line
{
clear:both;
margin-bottom:2px;
}
</style>
</head>

<body>
<h3>Image Gallery</h3>
<p>Try resizing the window to see what happens when the images does not
have enough room.</p>




<bh3 class="text_line">Second row</b>




</body>
</html>
```

Kode yang menggunakan pengaturan properti *clear:both* dicetak tebal pada kode diatas. Tampilan hasilnya pada browser:

Image Gallery

Try resizing the window to see what happens when the images does not have enough room.



Second row



Tampak pada gambar diatas teks "Second row" tidak mengalir (ditempatkan) setelah image ke 4. Demikian pula disamping kanan teks tidak diikuti dengan image ke 5. Ini akibat dari pengaturan properti *clear:both* terhadap penampilan teks "Second row".

PROPERTI - PROPERTI FLOAT CSS

Properti	Penjelasan	Nilai	CSS
Clear	Menentukan sisi yang tidak diperkenankan untuk pengembangan elemen float lain	Left, right, both, none, inherit	1
float	Menentukan apakah elemen akan diambangkan atau tidak	Left, right, none, inherit	1

HORIZONTAL ALIGN

PERATAAN ELEMEN BLOK

Elemen blok adalah sebuah elemen yang mengambil tempat seluruh lebar halaman yang tersedia untuk menampilkan dirinya serta mempunyai perpindahan baris sebelum dan setelah elemen tersebut. Contoh dari elemen blok adalah:

```
<h1>, <p>, <div>
```

PERATAAN TENGAH MENGGUNAKAN PROPERTI MARGIN

Elemen blok dapat dirata-tengahkan menggunakan properti margin yang diseting "auto". Sebagai catatan, penggunaan *margin:auto* ini tidak akan bekerja pada browser IE versi 8 dan sebelumnya, kecuali menggunakan deklarasi !DOCTYPE.

Pengaturan margin kiri dan kanan ke *auto* akan menentukan margin yang tersedia akan dibagi secara sama kiri dan kanan. Karena margin kiri dan kanan sama lebarnya, maka hasilnya elemen akan terletak ditengah-tengah halaman. Contoh:

```
<!DOCTYPE html>
<html>
<head>
<style>
.center
{
margin:auto;
width:70%;
background-color:#b0e0e6;
}
</style>
</head>
<body>
<div class="center">
<p>In my younger and more vulnerable years my father gave me some
advice that I've been turning over in my mind ever since.</p>
<p>'Whenever you feel like criticizing anyone,' he told me, 'just remember
that all the people in this world haven't had the advantages that you've
had.'</p>
</div>
<p><b>Note: </b>Using margin:auto will not work in IE8, unless a
!DOCTYPE is declared.</p>
</body>
</html>
```

Hasilnya pada browser:

In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since.

'Whenever you feel like criticizing anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.'

Note: Using `margin:auto` will not work in IE8, unless a `!DOCTYPE` is declared.

PERATAAN KIRI DAN KANAN MENGGUNAKAN PROPERTI POSITION

Metode lain perataan elemen adalah dengan menggunakan properti position dengan nilai absolute. Contoh:

```
.right
{
  position:absolute;
  right:0px;
  width:300px;
  background-color:#B0e0e6;
}
```

Elemen yang diposisikan absolut, tidak akan ditampilkan dengan mengikuti aliran normal dan dapat menindih elemen-elemen lain.

MENGATASI PERMASALAHAN KOMPATIBILITAS BROWSER

Saat melakukan perataan terhadap elemen-elemen, akan lebih baik jika sebelumnya mendefinisikan properti margin dan padding untuk elemen `<body>`. Ini untuk menghindari perbedaan penampilan pada berbagai jenis browser.

Ketika menggunakan property position pada browser IE, perlu diketahui satu permasalahan yang perlu ditangani. Jika suatu elemen kontainer mempunyai lebar yang telah ditentukan dan deklarasi `<!DOCTYPE>` tidak ada, IE akan menambahkan 17px margin pada sisi kanan. Sepertinya ini merupakan ruang yang dicadangkan untuk tampilan scrollbar. Oleh karena itu, selalu deklarasikan `!DOCTYPE` ketika menggunakan properti position:

```
body
{
  margin:0;
  padding:0;
}
.container
{
```

```
position:relative;
width:100%;
}
.right
{
position:absolute;
right:0px;
width:300px;
background-color:#b0e0e6;
}
```

PERATAAN KIRI DAN KANAN MENGGUNAKAN PROPERTI FLOAT

Properti float juga dapat digunakan untuk mengatur perataan kiri dan kanan. Contoh:

```
.right
{
float:right;
width:300px;
background-color:#b0e0e6;
}
```

PERMASALAHAN KOMPATIBILITAS BROWSER

Seperti pada penggunaan properti position, akan lebih baik jika mendefinisikan terlebih dahulu properti margin dan padding pada elemen BODY. Browser IE juga akan menambahkan spasi 17px margin pada sisi kanan. Maka atur terlebih dahulu deklarasi DOCTYPE ketika menggunakan properti float untuk perataan. Contoh:

```
body
{
margin:0;
padding:0;
}
.right
{
float:right;
width:300px;
background-color:#b0e0e6;
}
```

PSEUDO - CLASSES

Pseudo class adalah kata kunci yang ditambahkan kepada suatu selektor yang menentukan status khusus selektor tersebut dan biasa digunakan untuk menambahkan efek khusus ke beberapa selektor.

Sintaks dari jenis kelas ini:

```
selektor:pseudo-class {properti:nilai}
```

Juga dapat digunakan pada kelas CSS:

```
selektor.class:pseudo-class {properti:nilai}
```

Contoh penggunaan kelas ini dapat dilihat pada penampilan sebuah link dibawah ini:

```
a:link {color:#ff0000;}  
a:visited {color:#00ff00}  
a:hover: {color:#ff00ff}  
a:active {color:0000ff}
```

Link pada sebuah halaman web dapat mempunyai status yang berbeda-beda. Suatu link dapat berstatus visited apabila telah pernah dikunjungi isinya, berstatus hover apabila tengah dilewati mouse, dan beberapa status lainnya.

Pada contoh diatas, suatu link yang menggunakan pengaturan CSS ini, jika dilewati oleh mouse warna teks link akan berubah menjadi ungu, jika telah pernah diklik akan menjadi berwarna hijau.

Penting untuk diketahui, agar pseudo-class pada link ini dapat bekerja dengan baik, pendefinisian style perlu mengikuti urutan sebagai berikut:

link, visited, hover, lalu active.

Artinya definisikan dahulu style untuk pseudo-class link, dilanjutkan dengan visited, lalu hover, dan terakhir active.

Contoh penerapan pada suatu kelas CSS:

```
a.red:visited {color:#ff0000;}
```

```
<a class="red" href="sintaks.html">Sintaks CSS</a>
```


PSEUDO-CLASS *FIRST-CHILD*

First-child akan mencocokkan elemen yang dispesifikasikan sebagai first child (anak pertama) dari elemen yang lain. Dengan kata lain, elemen yang menggunakan pseudo class ini, akan diperiksa apakah didalam suatu elemen lain dia merupakan elemen pertama dari beberapa elemen lainnya. Jika menggunakan browser IE8 atau sebelumnya, agar kelas ini bekerja, tambahkan deklarasi DOCTYPE. Contoh penerapan kelas ini seperti terlihat dibawah ini:

```
<html>
<head>
<style>
p:first-child {
color:blue;
}
</style>
</head>
<body>
<p>Saya adalah elemen pertama dalam elemen BODY</p>
<p>Saya adalah elemen kedua dalam elemen BODY</p>
</body>
</html>
```

CONTOH-CONTOH LAIN :

Mencocokkan elemen *<i>* pertama pada semua elemen *<p>*

```
<!DOCTYPE html>
<html>
<head>
<style>
p > i:first-child
{
color:blue;
}
</style>
</head>

<body>
<p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>
<p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>
<p><b>Note:</b> For :first-child to work in IE8 and earlier, a DOCTYPE
must be declared.</p>
</body>
</html>
```

Tampilan dilayar browser:

I am a *strong* man. I am a *strong* man.

I am a *strong* man. I am a *strong* man.

Note: For first-child to work in IE8 and earlier, a DOCTYPE must be declared.

MENCOCOKKAN SEMUA ELEMEN <I> PADA SEMUA ELEMEN PERTAMA <P>

```
<html>
<head>
<style>
p:first-child i
{
color:blue;
}
</style>
</head>

<body>
<p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>
<p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>
<p><b>Note:</b> For :first-child to work in IE8 and earlier, a DOCTYPE
must be declared.</p>
</body>
</html>
```

Hasil pada layar browser:

I am a *strong* man. I am a *strong* man.

I am a *strong* man. I am a *strong* man.

Note: For first-child to work in IE8 and earlier, a DOCTYPE must be declared.

PSEUDO-CLASS *LANG*

Pseudo-class ini mendefinisikan aturan khusus untuk bahasa-bahasa yang berbeda. Sekali lagi, IE8 hanya akan mendukung kelas ini jika DOCTYPE dideklarasikan. Contoh:

```
<!DOCTYPE html>
<html>
<head>
<style>
q:lang(no)
{
quotes: "~" "~";
}
```

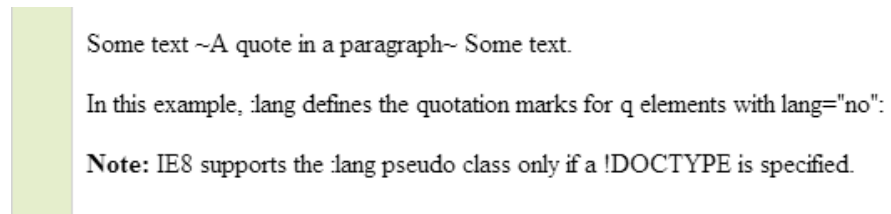
```

}
</style>
</head>

<body>
<p>Some text <q lang="no">A quote in a paragraph</q> Some text.</p>
<p>In this example, :lang defines the quotation marks for q elements with
lang="no":</p>
<p><b>Note:</b> IE8 supports the :lang pseudo class
only if a !DOCTYPE is specified.</p>
</body>
</html>

```

Hasilnya pada browser:



Perhatikan pada potongan teks "A quote in a paragraph" yang menggunakan pseudo-class q:lang(no) diapit dengan tanda "~".

PSEUDO-CLASS FOCUS

Pseudo ini digunakan untuk mengatur style pada elemen yang sedang memperoleh fokus. Contoh:

```

<!DOCTYPE html>
<html>
<head>
<style>
input:focus
{
background-color:yellow;
}
</style>
</head>

<body>
<form action="form_action.asp" method="get">
First name: <input type="text" name="fname" /><br>
Last name: <input type="text" name="lname" /><br>
<input type="submit" value="Submit" />
</form>

```

```
<p><b>Note:</b> IE8 supports the :focus pseudo-class only if a !DOCTYPE is specified.</p>
```

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

Hasilnya pada browser:

The screenshot shows a web form with two input fields: "First name:" and "Last name:". The "First name:" input field is highlighted with a yellow background, indicating it has focus. Below the input fields is a "Submit" button. Below the form, there is a note: "Note: IE8 supports the focus pseudo-class only if a !DOCTYPE is specified."

Perhatikan inputan pertama yang sedang mendapatkan fokus (kursor mouse berada di kotak masukan inputan tersebut) akan mengikuti style yang telah ditentukan.

Selektor	Contoh	Penjelasan Contoh
:link	a:link	semua link yang belum dikunjungi
:visited	a:visited	semua link yang pernah dikunjungi
:active	a:active	link yang aktif
:hover	a:hover	link yang sedang dilewati mouse
:focus	input:focus	elemen input yang mendapat fokus
:first-letter	p:first-letter	huruf pertama setiap elemen p
:first-line	p:first-line	baris pertama setiap elemen p
:first-child	p:first-child	setiap elemen p pertama dari elemen orang tuanya
:before	p:before	menyisipkan konten sebelum setiap elemen p
:after	p:after	menyisipkan konten setelah setiap elemen p
:lang(<i>language</i>)	p:lang(it)	setiap elemen p dengan nilai atribut lang yang diawali dengan "it"

PSEUDO - ELEMENTS

Pseudo element digunakan untuk menambahkan efek khusus pada beberapa selektor. Sintaks dari pseudo ini:

```
selektor:pseudo-element {properti:nilai}
```

juga dapat digunakan pada kelas-kelas CSS:

```
selektor.kelas:pseudo-element {properti:nilai}
```

PSEUDO - ELEMENT FIRST - LINE

Digunakan untuk menambahkan style khusus ke baris pertama dari suatu teks. Contoh dibawah ini browser akan memformat baris pertama dari teks didalam elemen p berdasarkan style yang didefinisikan pada pseudo-element first-line:

```
p:first-line
{
color:#ff0000;
font-variant:small-caps;
}
```

pseudo-element first line hanya dapat digunakan pada jenis elemen blok.

Properti-properti yang dapat digunakan pada pseudo-element first-line:

- font
- color
- background
- word-spacing
- letter-spacing
- text-decoration
- vertical align
- text-transform
- line-height
- clear

PSEUDO - ELEMENT FIRST - LETTER

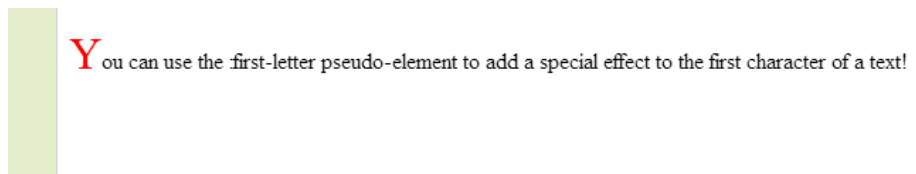
Pseudo-element ini akan menambahkan style khusus pada karakter pertama dari sebuah teks. Hanya dapat digunakan pada elemen blok. Contoh:

```
<!DOCTYPE html>
<html>
<head>
<style>
```

```
p:first-letter
{
color:#ff0000;
font-size:xx-large;
}
</style>
</head>

<body>
<p>You can use the :first-letter pseudo-element to add a special effect to the
first character of a text!</p>
</body>
</html>
```

Tampilan di browser:



Properti-properti yang dapat digunakan pada pseudo elemen ini:

- font
- color
- background
- margin
- padding
- border
- text-decoration
- vertical-align (hanya berlaku jika properti float diset none)
- text-transform
- line-height
- float
- clear

PSEUDO-ELEMENT DAN KELAS CSS

Pseudo-element juga dapat digunakan pada kelas CSS, misalnya:

```
p.artikel:first-letter {color:#ff0000}
<p class="artikel">Sebuah paragraf dalam artikel</p>
```

Contoh diatas akan menampilkan karakter pertama dari seluruh paragraf dengan kelas artikel berwarna merah.

MULTI PSEUDO-ELEMENT

Beberapa pseudo-element dapat dikombinasikan dalam penggunaannya. Sebagai contoh, dibawah ini karakter pertama dari paragraf akan dicetak merah dengan ukuran font *xx-large*. Sisa karakter dalam baris pertama akan berwarna biru dan berjenis *small-caps*. Sisa baris dalam paragraf akan ditampilkan dengan ukuran font dan warna default.

```
p:first-letter
{
color:#ff0000;
font-size:xx-large;
}
p:first-line
{
color:#0000ff;
font-variant:small-caps;
}
```


PSEUDO-ELEMENT BEFORE

Dapat digunakan untuk menyisipkan konten sebelum konten suatu elemen. Contoh berikut akan menampilkan sebuah gambar sebelum elemen H1:


```
<!DOCTYPE html>
<html>
<head>
<style>
h1:before {content:url(smiley.gif);}
</style>
</head>

<body>
<h1>This is a heading</h1>
<p>The :before pseudo-element inserts content before an element.</p>
<h1>This is a heading</h1>
<p><b>Note:</b> IE8 supports the content property
only if a !DOCTYPE is specified.</p>
</body>
</html>
```

Hasil di browser:

 **This is a heading**

The `:before` pseudo-element inserts content before an element.

 **This is a heading**

Note: IE8 supports the content property only if a `!DOCTYPE` is specified.


PSEUDO-ELEMENT AFTER

Dapat digunakan untuk menyisipkan konten setelah konten suatu elemen. Contoh berikut akan menampilkan sebuah gambar setelah elemen H1:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1:after {content:url(smiley.gif);}
</style>
</head>

<body>
<h1>This is a heading</h1>
<p>The :after pseudo-element inserts content after an element.</p>
<h1>This is a heading</h1>
<p><b>Note:</b> IE8 supports the content property only if a !DOCTYPE is
specified.</p>
</body>
</html>
```

Hasil di browser:

 **This is a heading**

The `:after` pseudo-element inserts content after an element.

 **This is a heading**

Note: IE8 supports the content property only if a `!DOCTYPE` is specified.

Berikut adalah seluruh pseudo-class/element yang dapat digunakan:

:link	a:link	Selects all unvisited links
:visited	a:visited	Selects all visited links
:active	a:active	Selects the active link
:hover	a:hover	Selects links on mouse over
:focus	input:focus	Selects the input element which has focus
:first-letter	p:first-letter	Selects the first letter of every <p> element
:first-line	p:first-line	Selects the first line of every <p> element
:first-child	p:first-child	Selects every <p> elements that is the first child of its parent
:before	p:before	Insert content before every <p> element
:after	p:after	Insert content after every <p> element
:lang(language)	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"

BATANG NAVIGASI

Batang navigasi yang berisi deretan menu sangat penting dibuat sedemikian rupa agar mudah digunakan. Navigasi ini diperlukan oleh pengunjung untuk menuju ke berbagai halaman sebuah website sehingga perlu dibuat dengan struktur yang baik dan jelas. Untuk membuat batang navigasi ini diperlukan element list sebagai dasarnya dan style pengaturan CSS untuk membuat tampilan navigasi lebih menarik, jelas, dan atraktif.

Berikut ini adalah contoh pembuatan batang navigasi menggunakan elemen list dan style CSS yang akan menampilkan navigasi horisontal.

KODE STANDAR PENGATURAN NAVIGASI

Tiga kode standar pengaturan CSS untuk membuat batang navigasi horisontal maupun vertikal adalah:

```
ul {  
  list-style-type:none;  
  margin:0;  
  padding:0;  
}
```

list-style-type diset *none* untuk menghilangkan bullet karena tidak diperlukan, *margin* dan *padding* diset 0 untuk menghilangkan pengaturan *default* dari browser.

NAVIGASI VERTIKAL

Jika navigasi yang akan dibuat berbentuk vertikal, maka tambahkan kode pengaturan diatas dengan pengaturan untuk elemen <a>:

```
a {  
  display:block;  
  width:60px;  
}
```

Pengaturan *display:block* dimaksudkan untuk membuat link menjadi elemen blok yang membuat seluruh area link *clickable* (tidak hanya teksnya saja) dan dapat diatur lebarnya. Pengaturan *width:60px;* untuk membuat lebar area elemen blok 60px saja, sedangkan defaultnya akan mencakup seluruh lebar layar browser yang tersedia.

NAVIGASI HORISONTAL

Untuk membuat navigasi horisontal ada dua cara, yaitu menggunakan item list yang *inline* atau item list mengambang (*float*). Keduanya bekerja sama baiknya.

Tetapi jika link yang dibuat perlu berukuran sama, sebaiknya menggunakan metode *floating*. Untuk metode *inline*, caranya dengan menentukan elemen `` sebagai elemen inline (secara default elemen `` adalah elemen block):

```
li {
  display:inline;
}
```

Kode ini ditambahkan pada kode standar diatas.

Kode lengkap navigasi horisontal dengan metode inline elemen ini sebagai berikut:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul
{
list-style-type:none;
margin:0;
padding:0;
padding-top:6px;
padding-bottom:6px;
}
li
{
display:inline;
}
a:link,a:visited
{
font-weight:bold;
color:#FFFFFF;
background-color:#98bf21;
text-align:center;
padding:6px;
text-decoration:none;
text-transform:uppercase;
}
a:hover,a:active
{
background-color:#7A991A;
}
</style>
</head>
<body>
<ul>
<li><a href="#home">Home</a></li>
```

```

<li><a href="#news">News</a></li>
<li><a href="#contact">Contact</a></li>
<li><a href="#about">About</a></li>
</ul>
<p><b>Note:</b> If you only set the padding for a elements (and not ul), the
links will go outside the ul element. Therefore, we have added a top and
bottom padding for the ul element.</p>
</body>
</html>

```

Tampilan di browser:



Note: If you only set the padding for a elements (and not ul), the links will go outside the ul element. Therefore, we have added a top and bottom padding for the ul element.

Kalau dengan menggunakan metode *floating* maka elemen `` dibuat mengambang dan lebar dari elemen `<a>` ditentukan:

```

li {
float:left;
}
a {
display:block;
width:60px;
}

```

Penggunaan pengaturan *float:left* untuk membuat elemen-elemen blok `<a>` tersusun berderet horisontal.

Kode lengkap metode ini sebagai berikut:

```

<!DOCTYPE html>
<html>
<head>
<style>
ul
{
list-style-type:none;
margin:0;
padding:0;
overflow:hidden;
}
li
{
float:left;
}

```

```
a:link,a:visited
{
display:block;
width:120px;
font-weight:bold;
color:#FFFFFF;
background-color:#98bf21;
text-align:center;
padding:4px;
text-decoration:none;
text-transform:uppercase;
}
a:hover,a:active
{
background-color:#7A991A;
}

</style>
</head>

<body>
<ul>
<li><a href="#home">Home</a></li>
<li><a href="#news">News</a></li>
<li><a href="#contact">Contact</a></li>
<li><a href="#about">About</a></li>
</ul>
</body>
</html>
```

Tampilan hasilnya pada browser:



GALERI IMAGE

CSS dapat digunakan juga untuk membuat suatu galeri gambar. Disini diberikan contoh pembuatan galeri gambar yang tersusun secara horisontal menggunakan teknik floating dan properti *display*.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.img
{
margin: 2px;
border: 1px solid #0000ff;
height: auto;
width: auto;
float: left;
text-align: center;
}
div.img img
{
display: inline;
margin: 3px;
border: 1px solid #ffffff;
}
div.img a:hover img {border: 1px solid #0000ff;}
div.desc
{
text-align: center;
font-weight: normal;
width: 120px;
margin: 2px;
}
</style>
</head>
<body>

<div class="img">
<a target="_blank" href="klematis_big.htm"></a>
<div class="desc">Add a description of the image here</div>
</div>
<div class="img">
```

```
<a target="_blank" href="klematis2_big.htm"></a>
<div class="desc">Add a description of the image here</div>
</div>
<div class="img">
<a target="_blank" href="klematis3_big.htm"></a>
<div class="desc">Add a description of the image here</div>
</div>
<div class="img">
<a target="_blank" href="klematis4_big.htm"></a>
<div class="desc">Add a description of the image here</div>
</div>

</body>
</html>
```

Dilayar browser, hasilnya sebagai berikut:



IMAGE OPACITY

Gambar transparan dapat dibuat dengan CSS secara mudah. Properti yang digunakan adalah *opacity* yang merupakan spesifikasi CSS 3. Contoh gambar berikut sebelum dibuat transparan:



Kemudian dibuat transparan dengan pengaturan CSS:

```
img {
  opacity:0.4;
  filter:alpha(opacity=40); /* untuk IE versi 8 dan sebelumnya */
}
```

Hasilnya:



IE 9, Firefox, Chrome, Opera, Safari, menggunakan properti *opacity* untuk membuat gambar menjadi transparan. Nilai yang dapat diberikan antara 0.0 - 1.0, semakin kecil nilainya semakin transparan gambar.

Sedangkan untuk IE 8 menggunakan properti `filter:alpha(opacity=x)`. Rentang nilai `x` dari 0 sampai dengan 100. Semakin kecil nilainya semakin transparan gambar.

Contoh berikutnya menampilkan teks dalam sebuah kotak yang dibuat transparan dengan kode sebagai berikut:

```
<html>
<head>
<style>
div.background
{
  width:500px;
```



```
        height:250px;
        background:url(klematis.jpg) repeat;
        border:2px solid black;
    }
div.transbox
{
    width:400px;
    height:180px;
    margin:30px 50px;
    background-color:#ffffff;
    border:1px solid black;
    opacity:0.6;
    filter:alpha(opacity=60); /* For IE8 and earlier */
}
div.transbox p
{
    margin:30px 40px;
    font-weight:bold;
    color:#000000;
}
</style>
</head>

<body>

<div class="background">
<div class="transbox">
<p>This is some text that is placed in the transparent box.
This is some text that is placed in the transparent box.
This is some text that is placed in the transparent box.
This is some text that is placed in the transparent box.
This is some text that is placed in the transparent box.
</p>
</div>
</div>

</body>
</html>
```

Hasilnya seperti ini:



Disini, pertama kali dibuat elemen DIV dengan kelas background dengan tinggi dan lebar yang tetap (fixed), sebuah gambar latar, dan border. Lalu dibuat elemen DIV yang lebih kecil dengan kelas transbox didalam elemen DIV pertama. DIV kelas transbox memiliki lebar yang tetap, berlatar warna, dan border serta transparan. Didalam elemen DIV transparan ini kemudian ditambahkan teks dengan elemen P.

IMAGE SPRITES

Halaman yang mengandung banyak gambar akan memerlukan waktu loading yang lebih lama. Setiap gambar yang digunakan memerlukan load dan membuat sebuah request ke server. Semakin banyak gambar semakin banyak request yang diperlukan ke server.

Untuk mengatasi hal ini, ada teknik yang dapat digunakan untuk mengurangi waktu loading yang lama karena penggunaan banyak gambar. Teknik *image sprites* ini akan menggabungkan gambar-gambar yang digunakan menjadi satu gambar tunggal. Dengan demikian jumlah request ke server akan dapat dikurangi dan sekaligus menghemat bandwidth.

Untuk menggunakan teknik ini, pertama gambar-gambar yang akan digunakan dikumpulkan menjadi satu gambar. Kemudian gambar hasil gabungan ini akan menggunakan kode pengaturan CSS untuk menentukan bagian gambar yang mana yang akan ditampilkan.

Dibawah ini adalah langkah-langkah pembuatan sebuah batang navigasi dengan teknik ini. Gambar hasil penggabungan beberapa gambar yang digunakan disimpan dengan nama *img_navsprites.gif*:



Kode pengaturan CSS untuk menentukan bagian gambar yang akan ditampilkan sebagai berikut:

```
<!DOCTYPE html>
<html>
<head>
<style>
img.home
{
width:46px;
height:44px;
background:url(img_navsprites.gif) 0 0;
}
img.next
{
width:43px;
height:44px;
background:url(img_navsprites.gif) -91px 0;
}
</style>
```

```

</head>

<body>

<br><br>

</body>
</html>

```

- ``, gambar yang digunakan pada elemen image hanya berupa gambar kecil transparan yang akan ditampilkan karena atribut `src` tidak boleh kosong.
- `width:64px; height:44px;`, mendefinisikan bagian gambar yang akan digunakan
- `background:url(img_navsprites.gif) 0 0;`, mendefinisikan gambar background dan posisinya (left opx dan top opx)

Hasilnya dilayar browser:



Langkah selanjutnya adalah menggunakan elemen list HTML untuk membuat menu navigasi. Berikut ini kode CSS pengaturannya:

```

<!DOCTYPE html>
<html>
<head>
<style>
#navlist{position:relative;}
#navlist li{margin:0;padding:0;list-style:none;position:absolute;top:0;}
#navlist li, #navlist a{height:44px;display:block;}

#home{left:0px;width:46px;}
#home{background:url('img_navsprites.gif') 0 0;}

#prev{left:63px;width:43px;}
#prev{background:url('img_navsprites.gif') -47px 0;}

#next{left:129px;width:43px;}
#next{background:url('img_navsprites.gif') -91px 0;}
</style>
</head>

<body>

```

```

<ul id="navlist">
  <li id="home"><a href="default.asp"></a></li>
  <li id="prev"><a href="css_intro.asp"></a></li>
  <li id="next"><a href="css_syntax.asp"></a></li>
</ul>
</body>
</html>

```

- `#navlist {position:relative;}`, posisi dibuat relative agar dapat diisi dengan elemen berposisi absolute didalamnya.
- `#navlist li {margin:0; padding:0; list-style:none; position:absolute; top:0;}`, item dari elemen list dibuat dengan properti ini.
- `#navlist li, #navlist a {height:44px; display:block;}`, ukuran tinggi gambar dibuat semua sama 44px;

Kode berikutnya adalah penentuan posisi dan style dari bagian gambar yang telah ditentukan:

- `#home{left:0px;width:46px;}` - meletakkan semua gambar kekiri dan lebar gambar diatur 46px
- `#home{background:url(img_navsprites.gif) 0 0;}` - menentukan gambar latar dan posisinya (left 0px, top 0px)
- `#prev{left:63px;width:43px;}` - pengaturan ini memposisikan gambar berikutnya berada 63px di kanan gambar sebelumnya (home) dan lebar gambar 43px.
- `#prev{background:url('img_navsprites.gif') -47px 0;}` - mengatur gambar latar 47px kekanan (`#home` lebar 46px + 1px garis pembatas)
- `#next{left:129px;width:43px;}`- memposisikan 129px ke kanan (dimulai dari `#prev` is 63px + `#prev width` 43px + ruang ekstra), dan lebar gambar 43px.
- `#next{background:url('img_navsprites.gif') no-repeat -91px 0;}` - mengatur posisi gambar latar 91px ke kanan (`#home` lebar 46px + 1px line divider + `#prev` lebar 43px + 1px garis pembatas)

Hasilnya pada browser:



Jika salah satu gambar diklik, maka halaman yang ditentukan oleh link akan ditampilkan di browser.

EFEK HOVER

Agar lebih menarik lagi, menu navigasi diatas ditambah dengan efek hover. Gambar yang digunakan gabungan dari 3 gambar navigasi dan 3 gambar untuk hover sebagai berikut:



Kemudian kode bagian pengaturan CSS ditambah kode sehingga menjadi (bagian yang tebal adalah kode yang ditambahkan):

```
#navlist{position:relative;}
#navlist li{margin:0;padding:0;list-style:none;position:absolute;top:0;}
#navlist li, #navlist a{height:44px;display:block;}

#home{left:0px;width:46px;}
#home{background:url('img_navsprites_hover.gif') 0 0;}
#home a:hover{background: url('img_navsprites_hover.gif') 0 -45px;}

#prev{left:63px;width:43px;}
#prev{background:url('img_navsprites_hover.gif') -47px 0;}
#prev a:hover{background: url('img_navsprites_hover.gif') -47px -
45px;}

#next{left:129px;width:43px;}
#next{background:url('img_navsprites_hover.gif') -91px 0;}
#next a:hover{background: url('img_navsprites_hover.gif') -91px -
45px;}
```

Tampilan hasilnya akan sama seperti saat sebelum ditambahkan efek hover, hanya saja ketika mouse melewati link home, prev, atau next, maka gambar masing-masing akan berubah menjadi lebih gelap.

TIPE - TIPE MEDIA

TIPE MEDIA

Output CSS dapat diarahkan ke media layar, printer, atau perangkat genggam. Dengan pengaturan tipe media ini, sebuah halaman dapat mempunyai sebuah layout untuk screen (layar), sebuah untuk printer (cetak), dan sebuah untuk perangkat genggam, dan lain-lain.

Beberapa properti CSS dirancang hanya untuk media tertentu. Contohnya, properti *voice-family* dirancang untuk user agen *aural*. Beberapa yang lain dirancang untuk berbagai tipe media. Contohnya, properti *font-size* dapat digunakan untuk media layar dan cetak yang mungkin dengan nilai yang berbeda. Biasanya sebuah dokumen pada layar memerlukan ukuran huruf yang lebih besar daripada ukuran di media kertas. Selain itu, jenis huruf sans-serif lebih mudah dibaca pada media layar, sementara untuk media kertas lebih mudah dibaca jika menggunakan jenis huruf serif.

PENGATURAN DENGAN @MEDIA RULE

Untuk membuat kode pengaturan yang berbeda untuk setiap media dalam satu style sheet yang sama dapat menggunakan @media rule. Berikut ini contoh pengaturan gaya yang berbeda untuk media layar yang akan menampilkan teks dengan jenis huruf Verdana dan ukuran huruf 14px, sedangkan untuk media cetak menggunakan ukuran 10px dengan jenis huruf Times. Semua media diatur mencetak teks dengan huruf tebal.

```
<html>
<head>
<style>
@media screen
{
p.test {font-family:verdana,sans-serif; font-size:14px;}
}
@media print
{
p.test {font-family:times,serif; font-size:10px;}
}
@media screen,print
{
p.test {font-weight:bold;}
}
</style>
</head>
```

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

Beberapa Tipe Media

Media Type	Description
all	Used for all media type devices
aural	Used for speech and sound synthesizers
braille	Used for braille tactile feedback devices
embossed	Used for paged braille printers
handheld	Used for small or handheld devices
print	Used for printers
projection	Used for projected presentations, like slides
screen	Used for computer screens
tty	Used for media using a fixed-pitch character grid, like teletypes and terminals
tv	Used for television-type devices

SELEKTOR ATTRIBUTE

Selain elemen HTML, class, dan ID, sebuah selektor dalam CSS dapat berupa atribut dari suatu elemen HTML. Artinya, atribut-atribut elemen HTML dapat juga diatur penampilannya menggunakan kode CSS. Dalam hal ini perlu diketahui bahwa untuk IE 7 dan IE 8 hanya akan mendukung apabila deklarasi DOCTYPE! digunakan. Fitur ini tidak didukung oleh IE versi 6 ke bawah.

Contoh berikut ini mengatur tampilan dari atribut *title*:

```
<!DOCTYPE html>
<html>
<head>
<style>
[title]
{
color:blue;
}
</style>
</head>

<body>
<h2>Will apply to:</h2>
<h1 title="Hello world">Hello world</h1>
<a title="W3Schools" href="http://w3schools.com">W3Schools</a>
<hr>
<h2>Will not apply to:</h2>
<p>Hello!</p>
</body>
</html>
```

Hasilnya pada browser:

Will apply to:

Hello world

[W3Schools](http://w3schools.com)

Will not apply to:

Hello!

NILAI SELEKTOR ATTRIBUTE

Selektor attribute dapat diatur nilainya seperti contoh dibawah ini:

```
<!DOCTYPE html>
<html>
<head>
<style>
[title=W3Schools]
{
border:5px solid green;
}
</style>
</head>

<body>
<h2>Will apply to:</h2>

<br>
<a title="W3Schools" href="http://w3schools.com">W3Schools</a>
<hr>
<h2>Will not apply to:</h2>
<p title="greeting">Hi!</p>
<a class="W3Schools" href="http://w3schools.com">W3Schools</a>
</body>
</html>
```

Hasilnya di browser:

Will apply to:



[W3Schools](http://w3schools.com)

Will not apply to:

Hi!

[W3Schools](http://w3schools.com)

SELEKTOR ATTRIBUTE YANG BERNILAI BANYAK

Contoh berikutnya mengatur tampilan elemen HTML yang memiliki atribut dengan nilai yang mengandung nilai tertentu:

```
<!DOCTYPE html>
<html>
```

```
<head>

<style>
[title~=hello]
{
color:blue;
}
</style>
</head>

<body>
<h2>Will apply to:</h2>
<h1 title="hello world">Hello world</h1>
<p title="student hello">Hello CSS students!</p>
<hr>
<h2>Will not apply to:</h2>
<p title="student">Hi CSS students!</p>
</body>
</html>
```

Hasilnya pada layar browser:

Will apply to:

Hello world

Hello CSS students!

Will not apply to:

Hi CSS students!

Contoh berikutnya akan mengatur tampilan semua elemen dengan atribut lang yang mengandung nilai tertentu:

```
<!DOCTYPE html>
<html>
<head>
<style>
[lang|=en]
{
color:blue;
}
</style>
</head>
```

```

<body>
<h2>Will apply to:</h2>
<p lang="en">Hello!</p>
<p lang="en-us">Hi!</p>
<p lang="en-gb">Ello!</p>
<hr>
<h2>Will not apply to:</h2>
<p lang="us">Hi!</p>
<p lang="no">Hei!</p>
</body>
</html>

```

Hasilnya pada browser:

Will apply to:

Hello!

Hi!

Ello!

Will not apply to:

Hi!

Hei!

MENGATUR PENAMPILAN FORM

Selektor attribute ini terutama bermanfaat untuk mengatur tampilan suatu form tanpa class atau ID. Contoh:

```

<!DOCTYPE html>
<html>
<head>
<style>
input[type="text"]
{
width:150px;
display:block;
margin-bottom:10px;
background-color:yellow;
}
input[type="button"]
{
width:120px;
margin-left:35px;
display:block;
}

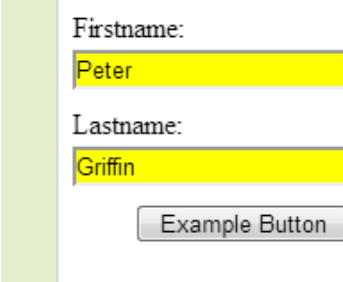
```

```
</style>
</head>
<body>

<form name="input" action="" method="get">
Firstname:<input type="text" name="Name" value="Peter" size="20">
Lastname:<input type="text" name="Name" value="Griffin" size="20">
<input type="button" value="Example Button">

</form>
</body>
</html>
```

Hasilnya pada layar browser:



Firstname:
Peter

Lastname:
Griffin

Example Button

PENGECUALIAN CSS

Meskipun CSS dapat berkolaborasi secara baik dengan HTML serta tampil di berbagai jenis browser, disini terdapat beberapa teknologi yang sebaiknya tidak digunakan bersama CSS. Dibawah ini adalah beberapa teknologi tersebut beserta alasan menghindari pemakaiannya bersama-sama CSS

ATRIBUT BEHAVIOR INTERNET EXPLORER

Browser Internet Explorer versi 5 memperkenalkan sebuah atribut *behavior*. Atribut ini untuk menambahkan beberapa perilaku pada suatu elemen HTML dengan menggunakan perintah CSS. Atribut ini hanya didukung oleh Internet Explorer, jadi jika ditampilkan menggunakan browser lain tidak akan bekerja. Berikut ini contoh dokumen HTML yang menggunakan atribut behavior pada elemen <h1>:

```
<html>
<head>
<title></title>
<style type="text/css">
h1 {behavior:url(behave.htc);}
</style>
</head>
<body>
<h1>Lewatkan mouse diatas teks ini</h1>
</body>
</html>
```

File dokumen HTML diatas menggunakan dokumen XML 'behave.htc' yang berisi:

```
<attach for="element" event="onmouseover" handler="hig_lite" />
<attach for="element" event="onmouseout" handler="low_lite" />
<script type="text/javascript">
function hig_lite(){
    element.style.color='red';
}
function low_lite(){
    element.style.color='blue';
}
</script>
```

REFERENSI

Situs w3schools.com