

CSS Tutorial

MEMFORMAT TAMPILAN HALAMAN WEB DENGAN CSS

Lisensi Dokumen :

Materi dalam dokumen ini dapat digunakan, dimodifikasi, disebarluaskan secara bebas untuk tujuan bukan komersial (non-profit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan. Tidak diperbolehkan melakukan penulisan ulang atas isi dokumen ini kecuali mendapatkan ijin terlebih dahulu dari Penulis.

Bekasi – November 2013

Muhamad Chatim Web Developer & Trainer

HP. 0856 91800 976 | Facebook.com/muhatim | <http://mht.site50.net> | bekasiweb.blogspot.com

DAFTAR ISI

DAFTAR ISI.....	1
Bagian 1 - PENGENALAN CSS	2
A. Apa itu CSS.....	2
B. Sintaks CSS.....	2
C. Jenis-Jenis Selector pada CSS.....	3
D. Mengintegrasikan CSS pada Dokumen HTML	6
Bagian 2 – BASIC ELEMENT STYLING.....	8
A. Background Styling	8
B. Text Styling.....	13
C. Font Styling	17
D. List Styling	23
E. Link Styling.....	28
F. Table Styling.....	30
Bagian 3 - CSS BOX MODEL	35
A. Apa itu CSS Box Model ?.....	35
B. Margin.....	37
C. Border	40
D. Padding	45
E. Outline	47

Bagian 1 - PENGENALAN CSS

A. Apa itu CSS

Cascading Style Sheet (CSS) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemrograman.

Sama halnya styles dalam aplikasi word processing seperti Microsoft Word yang dapat mengatur beberapa style, misalnya heading, sub-bab, bodytext, footer, images, dan style lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (file). Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

CSS dapat mengatur ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna hyperlink, warna mouse over, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. CSS adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

Sumber dan informasi selengkapnya, bisa ditemukan di :

 <http://www.w3.org>

 http://id.wikipedia.org/wiki/Cascading_Style_Sheets

B. Sintaks CSS

Sintaks Dasar pada CSS

Sintaks pada CSS terdiri dari dua bagian, yakni bagian **Selector** dan bagian **Declaration**. Di dalam Declaration terdapat sebuah **Property** dan **Value**. Ilustrasinya bisa dilihat pada gambar di bawah ini :



Selector adalah bagian pada HTML yang akan menjadi target dari pemformatan oleh CSS, umumnya adalah berupa elemen HTML. Sedangkan **Declaration** adalah bagian dari sintaks CSS yang berisi format-format (*styles*) yang dinyatakan dengan Property dan Value. Sebuah baris kode CSS bisa terdapat satu atau banyak Declaration.

Aturan penulisannya :

- Selector ditulis pada bagian awal baris kode CSS
- Declaration ditulis antara tanda “{ ... }”
- Penulisan Property dan Value pada Declaration dibatasi dengan tanda “:”
- Setiap Declaration diakhiri dengan tanda “;” sebelum ditulis Declaration baru.

FREE TO SHARE

Contoh

Berikut adalah contoh penulisan baris kode CSS :

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

Atau, agar mudah dibaca bisa ditulis sebagai berikut :

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

Penjelasan

Dari contoh di atas, bisa dijelaskan sebagai berikut :

- ✓ Selector yang digunakan adalah “**p**”, yakni sebuah tag HTML yang digunakan untuk mendefinisikan sebuah paragraf.
- ✓ Declaration pada contoh diatas dinyatakan dengan “**color:blue;**” dan “**text-align:center;**”
- ✓ Maka baris CSS tersebut akan menjadikan elemen paragraph (p) dalam dokumen HTML sebagai target pemformatan. Dalam hal ini, konten dalam elemen “p” akan diformat dengan warna biru dan perataan teksnya adalah *center*.

Bentuk Komentar pada CSS (CSS Comments)

Komentar (*comment*) digunakan untuk memberikan penjelasan pada baris kode CSS yang kita buat, dan akan sangat berguna ketika kita akan mengedit baris kode tersebut di kemudian hari. Baris kode yang dinyatakan sebagai komentar, tidak akan ditampilkan pada browser.

Komentar pada CSS dimulai dengan tanda “/*” dan diakhiri dengan tanda “*/”, contoh :

```
/* ini adalah baris komentar */  
p {  
  color:blue;  
  text-align:center;  
}  
/* ini adalah baris komentar yang lain */
```

C. Jenis-Jenis Selector pada CSS

Selector pada sintaks CSS umumnya adalah berupa tag HTML, namun bisa juga berupa atribut ID ataupun CLASS pada elemen HTML yang akan dilakukan pemformatan tampilan (*styling*).

Selector berupa Tag HTML

Semua tag HTML bisa digunakan sebagai Selector pada CSS, penulisannya adalah dengan menyebutkan nama dari tag tersebut.

Contoh :

```
body {background:#ff0000;}  
h1 {font-size: 36px;color:blue;}  
p {color:#333333;text-align:justify;}
```

FREE TO SHARE

Selector berupa atribut ID

Selector ID digunakan untuk melakukan pendefinisian *style* pada element HTML tunggal dan bersifat *unique*. Selector ID menggunakan atribut ID pada tag HTML, yang kemudian penyebutan dalam sintaks CSS menggunakan tanda “#” (*tanda pagar*).

Contoh :

File [index.html]

```
<p>artikel pada paragraf</>
<p>artikel pada paragraf</>
<p id="penting">artikel pada paragraf</>
```

Style.css

```
P {
  text-align:justify;
  color:grey;
}

#penting {
  text-align:center;
  color:red;
}
```

Pemformatan dokumen HTML dengan bentuk CSS tersebut akan menghasilkan tampilan halaman web dimana dua paragraf pertama berwarna abu-abu (*grey*) dengan *alignment* “*justify*”, sedangkan paragraf ketiga, yang memiliki atribut ID=“penting” akan terformat dengan *alignment* “*center*” dan warna teksnya adalah merah (CSS dengan selector “**#penting**”). Artinya paragraf ketiga, akan berbeda dengan dua paragraf diatasnya yang tidak memiliki atribut ID.

Selector berupa atribut CLASS

Selector CLASS digunakan untuk mendefinisikan pemformatan elemen HTML dengan membaca atribut CLASS pada tag HTML. Berbeda dengan Selector ID yang digunakan untuk pemformatan elemen tunggal, Selector CLASS umumnya digunakan untuk melakukan pemformatan pada satu atau lebih elemen HTML (*group of elements*). Selector CLASS pada penulisan baris kode CSS menggunakan tanda “.” (*tanda titik*).

Contoh :

File [index.html]

```
<p class="merah">konten pada paragraf ke-1</p>
<p class="biru"> konten pada paragraf ke-2</p>
<p class="merah"> konten pada paragraf ke-3</p>
<p class="biru"> konten pada paragraf ke-4</p>
```

Style.css

```
p{
  font-size: 12px;
  text-alignment:justify;
}
.merah{
  color: red;
}
.biru{
```

FREE TO SHARE

```
color: blue;
}
```

Dengan bentuk seperti di atas, maka akan dihasilkan sebuah halaman web sejumlah empat paragraf, dimana tiap paragraf memiliki ukuran huruf sebesar 12px dan *alignment* teksnya adalah *justify* (CSS dengan selector “p”).

Dari keempat paragraf tersebut, juga akan dihasilkan tampilan berupa :

- ✓ Paragraf ke-1 dan ke-3 memiliki warna teks merah (CSS dengan selector “.merah”)
- ✓ Paragraf ke-2 dan ke-4 memiliki warna teks biru (CSS dengan selector “.biru”)

Namun, perlu diingat bahwa bentuk penulisan CSS seperti di atas (contoh pada Selector Class) akan mengakibatkan semua elemen HTML yang memiliki atribut **class="biru"** akan ditampilkan dengan warna biru, dan semua elemen HTML yang memiliki atribut **class="merah"** akan ditampilkan dengan warna merah juga, tidak hanya berlaku pada elemen “p”.

Sebagai contoh, apabila dokumen HTML berbentuk seperti di bawah ini :

```
<p class="merah">konten pada paragraf ke-1</p>
<p class="biru"> konten pada paragraf ke-2</p>
<p class="merah"> konten pada paragraf ke-3</p>
<p class="biru"> konten pada paragraf ke-4</p>
<div class="biru">konten dalam div .... </div>
```

Maka, apabila penulisan CSS seperti pada contoh di atas, akan menghasilkan tampilan konten pada elemen div juga berwarna biru, karena elemen div memiliki atribut **class="biru"**.

Untuk menghindari hal tersebut, atau agar pemformatan pada CSS hanya berlaku pada elemen paragraf, maka penulisan pada CSSnya adalah sebagai berikut :

```
p{
  font-size: 12px;
  text-align: justify;
}
p.merah{
  color: red;
}
p.biru{
  color: blue;
}
```

Selector untuk Elemen Turunan

Dalam dokumen HTML, konten sebuah elemen bisa berupa elemen lainnya. Maka diperlukan cara penulisan tersendiri untuk memilih sebuah elemen HTML yang akan ditetapkan sebagai target pemformatan dengan CSS. Mari kita lihat contoh di bawah ini :

File [index.html]

```
<h1>Judul dengan <em>bagian penting</em> di dalamnya</h1>
<p>Artikel yang terdapat <em>bagian penting</em> di dalamnya</p>
```

Apabila kita ingin melakukan pemformatan pada elemen “em” yang berada pada elemen “h1”, maka penulisan CSSnya adalah sebagai berikut :

```
h1 em { color:red; }
```

FREE TO SHARE

Dengan penulisan CSS seperti di atas (menambahkan spasi antar Selector), maka hanya bagian elemen “em” yang berada di dalam elemen “h1” saja yang akan ditampilkan dengan warna merah, sedangkan elemen “em” diluar elemen “h1” akan tetap tampil mengikuti tampilan *default*.

Pengelompokan Selector (Grouping)

Declaration pada CSS bisa diterapkan untuk satu jenis atau beberapa jenis elemen sekaligus. Caranya adalah dengan menuliskan beberapa Selector dalam satu baris kode CSS yang dipisahkan dengan tanda “,” (koma) sebelum penulisan Declaration. Contohnya, pada CSS dinyatakan :

```
h1, h2, h3, h4{
  color:red;
}
```

Maka akan menghasilkan tampilan halaman web dimana konten pada elemen “h1”, “h2”, “h3” dan “h4” memiliki warna yang sama, yakni merah. Demikian juga pemformatan untuk elemen yang lain bisa dilakukan dengan cara yang sama.

D. Mengintegrasikan CSS pada Dokumen HTML

Ada tiga cara untuk menerapkan pemformatan CSS pada dokumen HTML, yakni dengan metode *External Style Sheet*, *Internal Style Sheet* dan *Inline Style Sheet*.

External Style Sheet

Cara pertama disebut dengan *External Style Sheet*, yaitu dengan cara membuat dokumen CSS yang terpisah, kemudian dibuat tautan dengan tag <link> pada elemen *header* dalam dokumen HTML yang akan diterapkan pemformatan dengan CSS tersebut.

Sebagai contoh, baris kode CSS dinyatakan sebagai berikut :

```
body {background:lavender;color:#333333;}
h1{color:red;}
```

Disimpan dalam sebuah dokumen berformat *style sheet* dengan nama : **mystyle.css**

Maka, untuk menerapkan pemformatan tersebut ke dalam dokumen HTML, digunakan tag <link> yang dituliskan pada elemen *head* sebagai berikut :

```
<head>
<title>Contoh penggunaan CSS External</title>
<link rel="stylesheet" type="text/css" href="mystyle.css" />
</head>
```

Note: Dalam contoh ini, file mystyle.css berada dalam satu folder dengan file HTML-nya.

Dengan cara ini, kita bisa menerapkan pemformatan CSS tersebut ke banyak dokumen HTML, dan cukup menambahkan tautan untuk tiap masing-masing dokumen. Keuntungan lainnya, adalah apabila suatu saat kita ingin mengganti format tampilan halaman web dengan format tampilan yang baru, cukup mengganti tautan tersebut ke file CSS baru yang akan digunakan.

Internal Style Sheet

Cara kedua adalah dengan menggunakan *Internal Style Sheet*, yakni dengan cara menuliskan baris kode CSS pada elemen *head* menggunakan tag `<style>` seperti di bawah ini :

```
<head>
<style>
body {background:lavender;color:#333333;}
h1{color:red;}
</style>

<body>
...
```

Cara ini biasa digunakan untuk melakukan pemformatan pada sebuah halaman web yang akan ditampilkan berbeda dengan halaman web lainnya. Misalnya, apabila kita memiliki lima halaman web yang sudah dihubungkan dengan sebuah file CSS, namun ada satu halaman web yang memiliki format yang sedikit berbeda, maka digunakanlah *Internal style sheet*.

Inline Style Sheet

Cara ketiga adalah menggunakan *Inline Style Sheet*, yaitu dengan menggunakan atribut “style” pada tag-tag HTML yang akan ditambahkan pemformatan CSS. Misal, kita akan membuat tampilan elemen h1 menjadi berwarna merah, maka penulisan elemen h1-nya adalah sebagai berikut :

```
<h1 style="color:red" >Ini adalah judul artikel</h1>
```

Atribut “style” pada tag HTML tersebut, bisa diberikan lebih dari satu property CSS sama halnya dengan menggunakan *Internal Style Sheet* ataupun *External Style Sheet*.

Bagian 2 - BASIC ELEMENT STYLING

Pada bagian ini akan dibahas bagaimana cara melakukan pemformatan (*styling*) pada elemen HTML dengan menggunakan baris kode CSS. Terdapat 6 (enam) hal mendasar yang akan dibahas, yakni : *Background styling, Text styling, Font styling, List styling, Link styling* dan *Table styling*.

Note : Contoh-contoh penerapan CSS pada bagian ini menggunakan metode External Style Sheet

A. Background Styling

Sebuah elemen HTML bisa diberikan efek latar belakang (*background*) dengan menggunakan Property “**background**” pada CSS. Property ini memiliki beberapa bentuk, yakni :

Background-color

Property “**background-color**” digunakan untuk mengatur latar belakang sebuah elemen dengan nilai berupa warna.

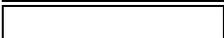
Sintaks-nya :

```
background-color: color-value;
```

Keterangan :

- Nilai warna (*color-value*) pada CSS biasanya menggunakan kode warna dalam beberapa varian, yakni *Hexadesimal color, RGB color* dan *Color-name*.
- Untuk mengetahui kode warna yang bisa digunakan, bisa dilihat dari tabel warna dasar (*basic color*) di bawah ini.

Basic Color Keywords

Color	Color name	HEX	RGB
	black	#000000	0, 0, 0
	silver	#C0C0C0	192, 192, 192
	gray	#808080	128, 128, 128
	white	#FFFFFF	255, 255, 255
	maroon	#800000	128, 0, 0
	red	#FF0000	255, 0, 0
	purple	#800080	128, 0, 128
	fuchsia	#FF00FF	255, 0, 255
	green	#008000	0, 128, 0
	lime	#00FF00	0, 255, 0
	olive	#808000	128, 128, 0
	yellow	#FFFF00	255, 255, 0
	navy	#000080	0, 0, 128
	blue	#0000FF	0, 0, 255
	teal	#008080	0, 128, 128
	aqua	#00FFFF	0, 255, 255

FREE TO SHARE

Background-image

Property “**background-image**” digunakan untuk menjadikan sebuah file gambar (*image*) menjadi latar belakang pada sebuah elemen HTML.

Sintaks-nya :

```
background-image: url('url-file-gambar');
```

Keterangan :

- File gambar yang digunakan bisa berupa file *.jpg, *.png ataupun *.gif.
- Secara *default*, gambar yang dijadikan *background* akan ditampilkan berulang, secara horizontal dan vertikal.
- Url file gambar bisa menggunakan *relative path* ataupun *absolute url*.

Contoh :

Edit Declaration untuk Selector “body” pada file CSS, sehingga menjadi seperti berikut :

```
body {  
  background-color: rgb(153,204,255);  
  background-image : url(wallpaper.jpg);  
}
```

Note : file “wallpaper.jpg” berada dalam folder yang sama dengan file “index.html”, dengan ukuran file 200 x 200 pixel.

Hasilnya akan terlihat seperti gambar berikut :



Background-repeat

Property “**background-repeat**” digunakan untuk mengatur perulangan (*repeatation*) pada gambar yang ditampilkan sebagai *background* dengan Property “background-image”. Perulangan tampilan gambar tersebut, bisa diatur ke arah horizontal (sumbu x) ataupun ke arah vertikal (sumbu y).

Sintaks-nya :

```
Background-repeat: repeat | no-repeat | repeat-x | repeat-y;
```

FREE TO SHARE

Contoh :

Edit Declaration untuk Selector “body” pada file CSS, sehingga menjadi seperti berikut :

```
body {  
  background-color:rgb(153,204,255);  
  background-image : url('wallpaper.jpg');  
  background-repeat : no-repeat;  
  background-position : top center;  
}
```

Simpan pekerjaan dan lihat tampilan halaman web pada browser.

Meringkas Penulisan Property Background

Pada contoh-contoh sebelumnya, kita harus menuliskan beberapa Property untuk mengatur tampilan *background* agar sesuai dengan kebutuhan desain. Namun, kita bisa meringkas penulisan Property tersebut dengan cukup menuliskan *value*-nya dalam satu baris pada Property “background”.

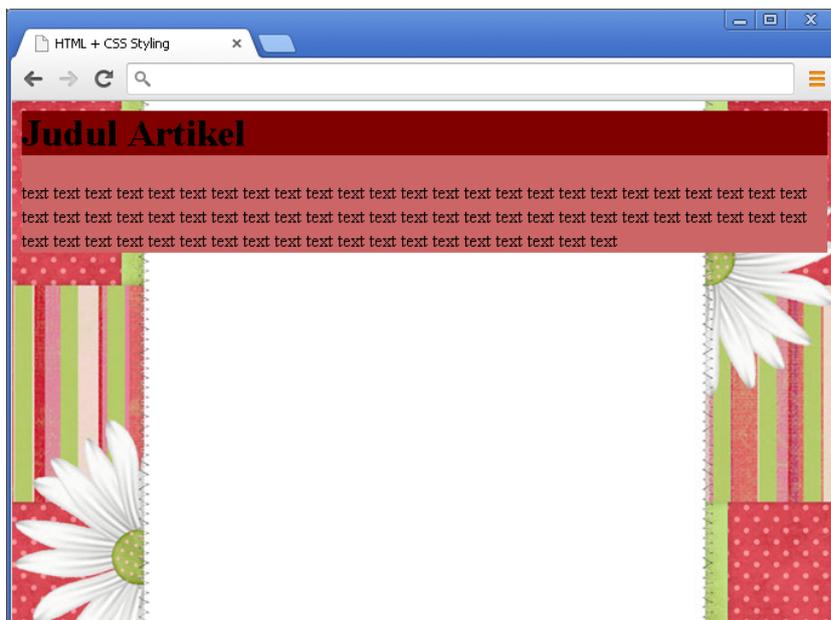
Contoh :

Editlah file CSS pada Selector “body” sehingga menjadi sebagai berikut :

```
body {  
  background: rgb(153,204,255) url(background.jpg) repeat-y top  
  center;  
}
```

Keterangan : Masing-masing *value* dibatasi dengan spasi.

Hasilnya akan terlihat seperti gambar berikut ini :



Text-decoration

Property “**text-decoration**” digunakan untuk menambah atau membuang ornamen tertentu pada teks.

Sintaks-nya :

```
text-decoration : none | underline | overline | line-through |
                blink;
```

Property ini biasa digunakan untuk menghilangkan ornamen *underline* pada hyperlink, atau memberikan efek tertentu untuk bagian-bagian yang dianggap perlu pada sebuah artikel.

Contoh :

Edit file HTML pada contoh sebelumnya sehingga menjadi seperti berikut :

```
<div id="artikel">
  <h1>Judul Artikel<sup>Penting</sup></h1>
  <p>text text text
    text text text text text text text text text text text text text
    text text text text text text <br>
  <a href="#">Baca Selengkapnya...</a>
</p>
</div>
```

Editlah file CSS dengan menambahkan baris berikut :

```
.artikel h1 sup {
  color:#FF0;
  text-decoration:blink;
}
.artikel p a {
  color:#ffffff;
  text-decoration:none;
}
```

Dari struktur dokumen HTML dan pemformatan CSS seperti di atas, akan dihasilkan tampilan halaman web seperti gambar di bawah ini:



Catatan : Ada beberapa browser yang tidak bisa menampilkan teks Blink

FREE TO SHARE

- *Uppercase* : teks ditampilkan dengan huruf besar seluruhnya
- *Lowercase* : teks ditampilkan dengan huruf kecil seluruhnya

Contoh :

Edit Declaration untuk Selector “.artikel h1” dan “.artikel h1 sup”, sehingga menjadi seperti berikut :

```
.artikel h1 {
  background-color:maroon;
  color:#ffffff;
  text-align:center;
  text-transform: uppercase;
}
.artikel h1 sup {
  color:#FF0;
  text-decoration:blink;
  text-transform: lowercase;
}
```

Hasilnya, konten pada Class “artikel” elemen H1 akan diubah menjadi bentuk huruf kapital, sedangkan pada elemen SUP menjadi huruf kecil. Perhatikan gambar di bawah ini :



C. Font Styling

CSS juga memungkinkan kita untuk melakukan pemformatan pada font, seperti menentukan ukuran font atau memilih jenis font yang akan digunakan. Property pada CSS yang digunakan untuk melakukan pemformatan font antara lain : *font-size*, *font-family*, *font-style* dan *font-weight*.

Font-size

Property “font-size” digunakan untuk menentukan ukuran dari font (huruf) yang akan ditampilkan. Property ini bisa diterapkan untuk elemen-elemen HTML yang berhubungan dengan pengolahan teks, seperti *heading* dan *paragraf*.

FREE TO SHARE

Namun perlu dicatat bahwa, meskipun kita bisa melakukan pemformatan pada ukuran font, tetapi sangat tidak dianjurkan untuk menggunakan Property ini untuk menjadikan *heading* terlihat seperti paragraf dengan mengecilkan ukuran font-nya ataupun sebaliknya, menjadikan paragraf seperti *heading* dengan memperbesar ukuran font-nya. Tetap gunakan Tag Heading untuk membuat *heading* pada artikel.

Sintaks-nya :

```
font-size: <absolute-size> | <relative-size> | <lenght> | <percent>;
```

Keterangan :

- *Value* pada Property “font-size” bisa berupa :
 - ✓ *Absolute-size*, yakni ditentukan dengan nilai : *xx-small*, *x-small*, *small*, *medium*, *large*, *x-large*, dan *xx-large*.
 - ✓ *Relative-size*, yakni ditentukan dengan nilai : *larger* atau *smaller*
 - ✓ *Lenght*, yakni dengan menentukan ukuran font dalam satuan *pixel* atau dalam *em*.
 - ✓ *Percent*, yakni menentukan ukuran font dengan nilai persen (%)
- Jika ukuran font tidak ditentukan, maka akan digunakan nilai default, yakni untuk teks normal seperti paragraf akan memiliki nilai 16px (1em = 16px = 100%).

Mengatur Ukuran Font dalam Satuan Pixel

Berikut ini adalah contoh penentuan ukuran font pada CSS dengan satuan *Pixel* :

```
h1 { font-size: 40px; }  
h2 { font-size: 30px; }  
p { font-size: 14px; }
```

Dengan menentukan ukuran font seperti di atas, maka akan dihasilkan tampilan halaman web dengan ukuran font untuk h1 sebesar 40px, h2 sebesar 30px dan font normal untuk paragraf sebesar 14px.

Pada contoh di atas, memungkinkan browser (seperti Internet Explorer 9, Firefox, Chrome, Opera, dan Safari) untuk *me-resize* ukuran font apabila *user* mengaktifkan fungsi *zoom-in* ataupun *zoom-out* pada browser. Khusus untuk Internet Explorer, hanya berlaku untuk versi IE9.

Mengatur Ukuran Font dalam Satuan Em

Berikut ini adalah contoh penentuan ukuran font pada CSS dengan satuan *Em*.

```
h1 { font-size: 2.5em; }           >> 40px/16 = 2.5em  
h2 { font-size: 1.875em; }       >> 30px/16 = 1.875em  
p { font-size: 0.875em; }       >> 14px/16 = 0.875em
```

Untuk menghitung nilai *pixel* ke dalam satuan *em*, yaitu dengan berpatokan bahwa 1em adalah ukuran normal untuk sebuah teks, dimana 1em = 16px. Sehingga untuk mengkonversi nilai *pixel* ke dalam *em* bisa dihitung dengan rumus $em = pixel/16$.

Dengan menggunakan satuan *em*, memungkinkan semua browser (termasuk IE versi sebelum IE9) untuk *re-size* ukuran font saat *user* mengaktifkan fungsi *zoom-in* ataupun *zoom-out*. Penggunaan satuan *em* ini juga telah direkomendasikan oleh W3C.

Namun, pada contoh diatas masih terdapat satu kekurangan. Yakni apabila *user* menggunakan IE versi yang lebih lama, ukuran font akan lebih besar dari seharusnya apabila di *zoom-in* dan lebih kecil dari seharusnya apabila di *zoom-out*.

Menggabungkan Satuan Em dengan nilai Persen

Sebagai solusi untuk kedua masalah di atas, agar ukuran Font sesuai dengan pengaturan yang ditetapkan dan berlaku untuk seluruh *browser*, adalah dengan menyatakan ukuran Font *default* dalam satuan persen pada elemen `<body>`.

Contoh :

Masih menggunakan file HTML dan CSS pada contoh sebelumnya, tambahkan dan editlah baris kode CSS seperti pada contoh di bawah ini:

```
body {
    font-size:100%;
}
.artikel h1 {
    font-size:2.5em;
}
.artikel p {
    font-size: 0.875em;
}
```

Dengan penulisan CSS seperti di atas, akan dihasilkan sebuah halaman web dengan ukuran font normal adalah 100% (16px / 1em). Sedangkan untuk h1 pada Class “artikel” ukuran font-nya adalah 2.5em (40px) dan untuk paragraf pada Class “artikel” ukuran font-nya adalah 0.875em (14px).

Tampilan pada browser akan terlihat seperti pada gambar di bawah ini :



Font-family

Pada CSS, dikenal dua jenis nama *font family*, yakni :

- **Font Family**; yaitu sebuah nama jenis font yang spesifik, seperti : “Times New Roman” atau “Arial”.
- **Generic Family**; yaitu kelompok *font family* yang memiliki kemiripan tampilan, seperti “serif”, “sans-serif” atau “monospace”.

Perhatikan tabel di bawah ini !

Generic family	Font family	Keterangan
Serif	Times New Roman Georgia	Font “Serif” mempunyai bentuk ujung yang runcing pada beberapa bentuk karakternya
Sans-serif	Arial Verdana	“Sans” berarti “tidak dengan/tanpa”, artinya jenis huruf ini tidak mempunyai bentuk ujung runcing pada tiap karakternya.
Monospace	Courier New Lucida Console	Semua karakter jenis font ini memiliki ukuran lebar yang sama.



Perbedaan antara font Serif dan font Sans-serif.

Untuk mengatur jenis font agar sesuai dengan yang diinginkan, pada CSS digunakan Property “**font-family**”.

Sintaks-nya :

```
font-family: <family-name> | <generic-family>;
```

Keterangan :

- *Value* pada Property ini harus dinyatakan lebih dari satu jenis font (*font-family*), sebagai cadangan apabila font pertama tidak ditemukan, akan dicari font kedua, apabila tidak ditemukan juga akan dicari yang sejenis dengannya (*generic-family*)
- Jika nama *font-family* terdiri dari dua kata atau lebih, ditulis dengan menggunakan tanda petik (“*contoh nama font*”).

Contoh :

Edit Declaration untuk Selector “.artikel h1” dan “.artikel p”, sehingga menjadi seperti berikut :

```
.artikel h1 {  
    font-size:2.5em;  
    font-family:"Trebuchet MS", Arial, Helvetica, sans-serif;  
}  
.artikel p {  
    font-size: 0.875em;
```

FREE TO SHARE

```
font-family: "Palatino Linotype", "Book Antiqua", Palatino, serif;
}
```

Hasilnya, font pada elemen H1 akan ditampilkan dengan jenis font “Trebuchet MS” sedangkan pada elemen P dengan font “Palatino Linotype”. Perhatikan gambar di bawah ini:



Font-style

Property “**font-style**” umumnya digunakan untuk mengatur tampilan *italic text*. Property ini memiliki tiga *value*, yakni : *normal*, *italic* dan *oblique*.

Sintaks-nya :

```
font-style : normal | italic | oblique;
```

Keterangan :

- *Value* defaultnya berupa *normal*.
- *Italic*; menghasilkan teks miring
- *Oblique*; sama dengan *italic*, yaitu membuat tampilan teks miring.

Contoh :

Edit Declaration untuk Selector “.artikel h1 sup”, sehingga menjadi seperti berikut :

```
.artikel h1 sup {
  color:#FF0;
  text-decoration:blink;
  text-transform: lowercase;
  font-style:italic;
}
```

Hasilnya, teks pada elemen SUP akan diubah menjadi bentuk *italic* (miring), seperti terlihat pada gambar di bawah ini :



Font-variant

Property “font-variant” digunakan untuk membuat variasi pada teks, dimana *value* dari Property ini adalah “normal” dan “small-caps”.

Sintaks-nya :

```
font-variant: normal | small-caps;
```

Keterangan:

- *Value* defaultnya adalah *normal*.
- *Small-caps*; menghasilkan tampilan teks berupa “SMALL CAPS”.

Contoh :

Edit Declaration untuk Selector “.artikel p a”, sehingga menjadi seperti berikut :

```
.artikel p a {  
  color: #ffffff;  
  text-decoration: none;  
  font-variant: small-caps;  
}
```

Simpan file CSS, dan perhatikan hasilnya pada tampilan *browser*.

Font-weight

Property “font-weight” digunakan untuk menentukan nilai ketebalan font (*weightness*). Nilai dari Property ini adalah : “normal”, “bold”, “bolder”, “lighter” atau dengan menggunakan nilai antara 100, 200, s/d 900.

Sintaks-nya :

```
font-weight : normal | bold | bolder | lighter | <100 s/d 900>
```

Keterangan :

- *Value* defaultnya adalah *normal* (bisa menggunakan nilai 400)
- *Bold*; menghasilkan jenis font yang tebal (bisa menggunakan nilai 700)

FREE TO SHARE

- *Bolder*; menghasilkan jenis font yang lebih tebal dari *bold*
- *Lighter*; menghasilkan jenis font yang lebih tipis dari *normal*.

Contoh :

Edit Declaration untuk Selector “.artikel p a”, sehingga menjadi seperti berikut :

```
.artikel p a {
  color: #ffffff;
  text-decoration: none;
  font-variant: small-caps;
  font-weight: bold;
}
```

Simpan file CSS, dan perhatikan hasilnya pada tampilan *browser*.

Meringkas Penggunaan Property Font

Pengaturan jenis font dengan Property yang banyak pada sebuah elemen, bisa diringkas dengan menggunakan Property “**font**”. Cara penggunaannya adalah dengan menuliskan Property-Property yang dibutuhkan sesuai dengan urutannya. Sintaks-nya :

```
font : <font-style> <font-variant> <font-weight> <font-size/line-height> <font-family>;
```

Keterangan :

- Property “font-size” dan “font-family” harus selalu disertakan, jika salah satunya hilang akan digunakan nilai *default* dari masing-masing Property.
- Penggunaan Property “line-height” bisa dibaca di bagian yang berhubungan.

D. List Styling

Pada dokumen HTML, terdapat dua jenis list yang sering digunakan, yakni :

- Unordered list, dengan Tag atau biasa disebut dengan *bullets*.
- Ordered list, dengan Tag atau biasa disebut dengan *numbering*.

Dengan CSS, kita bisa mengatur tampilan pada masing-masing jenis list. Untuk Unordered List, kita bisa mengatur bentuk *bullets*-nya apakah berbentuk *circle*, *disc* atau *square*. Sedangkan untuk Ordered List, kita bisa mengatur bentuk *numbering*-nya apakah menggunakan angka, huruf, huruf romawi, dan lain sebagainya.

Property CSS yang digunakan untuk melakukan pemformatan pada list adalah : *list-style-type*, *list-style-position*, dan *list-style-image*.

List-style-type

Property “list-style-type” digunakan untuk mengatur bentuk dari *bullets* atau *numbering* yang digunakan.

FREE TO SHARE

Sintaks-nya :

```
list-style-type : <marker-type> | none ;
```

Keterangan :

- Property ini bisa digunakan pada elemen , ataupun .
- Nilai default untuk adalah “disc” dan nilai default untuk adalah “decimal”.
- *Value (marker type)* yang biasa digunakan untuk *bullets dan numbering* adalah sebagai berikut :

Value	Keterangan
circle	<i>Bullets</i> berbentuk lingkaran
disc	<i>Bullets</i> berbentuk bulatan hitam
square	<i>Bullets</i> berbentuk kotak
decimal	<i>Numbering</i> berupa angka desimal (1, 2, 3 dst)
decimal-leading-zero	<i>Numbering</i> berupa angka desimal dengan nilai 0 di depan (01, 02, 03, dst)
lower-alpha	<i>Numbering</i> berupa huruf kecil (a, b, c, d dst)
lower-roman	<i>Numbering</i> berupa angka romawi kecil (i, ii, iii, iv dst)
upper-alpha	<i>Numbering</i> berupa huruf besar (A, B, C, D dst)
upper-roman	<i>Numbering</i> berupa angka romawi besar (I, II, III, IV dst)

- Selain jenis *bullets* dan *numbering* tersebut, masih terdapat beberapa jenis lagi seperti : *armenian, georgian, hiragana, katakana, dsb.*

Contoh :

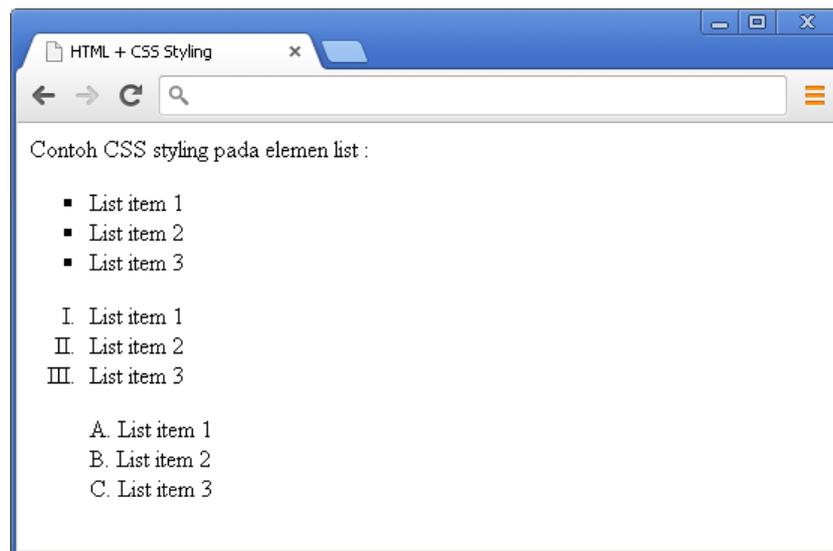
File [list.html]

```
<body>
Contoh CSS styling pada elemen list :
<ul class="square">
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ul>
<ol class="roman">
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>
<ol class="alpha">
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>
</body>
```

File [list-style.css]

```
ul.square {
    list-style-type:square;
}
ol.roman {
    list-style-type:upper-roman;
}
ol.alpha {
    list-style-type:upper-alpha;
}
```

Perhatikan tampilan yang dihasilkan seperti pada gambar di bawah ini :



List-style-position

Property “**list-style-position**” digunakan untuk mengatur posisi dari *bullets* atau *numbering* pada elemen list. Nilai *default*-nya adalah *outside*, namun bisa diganti menjadi *inside*.

Sintaks-nya :

```
list-style-position: outside | inside;
```

Contoh :

File [list.html]

```
<body>
List-style-position : outside | inside
<ol class="roman">
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>
<ol class="alpha">
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>
</body>
```

File [list-style.css]

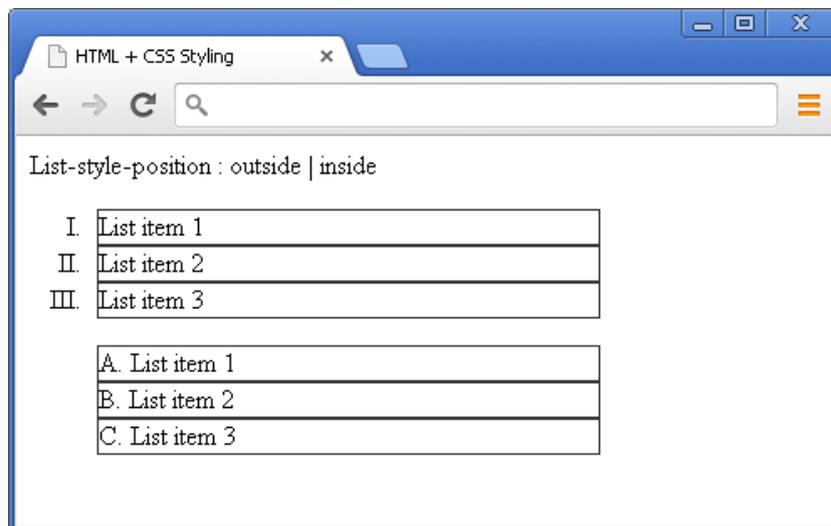
FREE TO SHARE

```

ol {
  width:300px;
}
li {
  border: 1px solid #333;
}
ol.roman {
  list-style-type:upper-roman;
}
ol.alpha {
  list-style-type:upper-alpha;
  list-style-position:inside;
}

```

Perhatikan tampilan yang dihasilkan seperti gambar di bawah ini :



List-style-image

Property “list-style-image” digunakan untuk menggantikan bentuk-bentuk bullets pada elemen list menjadi bentuk bullets khusus yang kita buat dalam bentuk file gambar. Biasanya hal ini kita lakukan pada saat membuat link menu atau memang ingin menampilkan list dalam bentuk yang tidak biasa.

Sintaks-nya :

```
list-style-image : url('url-file-gambar');
```

Keterangan :

- Value default dari Property ini adalah *none*.
- Gambar bisa menggunakan jenis file PNG, GIF ataupun JPG.
- Sebisa mungkin ukuran file gambar sesuai dengan ukuran font yang diterapkan pada elemen list.

Contoh :

FREE TO SHARE

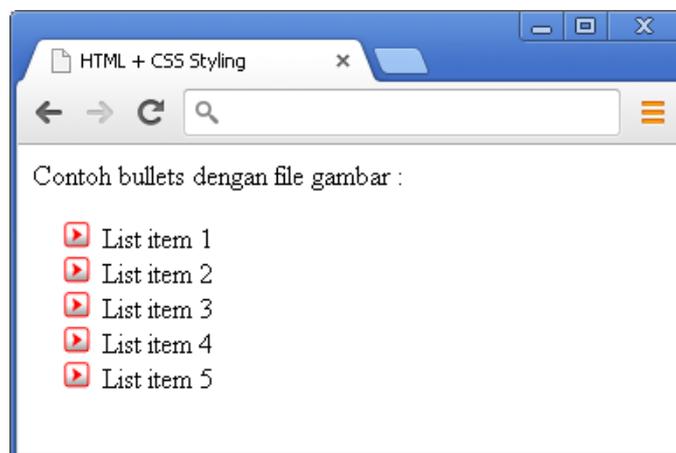
File [List-image.html]

```
<ul>
<li>List item 1</li>
<li>List item 2</li>
<li>List item 3</li>
<li>List item 4</li>
<li>List item 5</li>
</ul>
```

File [list-style.css]

```
ul {
    list-style-image:url('list-arrow.png');
}
```

Dengan struktur file HTML dan CSS seperti di atas, akan dihasilkan tampilan halaman web seperti pada gambar di bawah ini :



Keterangan : file gambar bullets (list-arrow.png) berada dalam satu folder yang sama dengan file CSS.

Meringkas Penggunaan Property List

Pada contoh sebelumnya, kita telah menggunakan tiga Property pada list yakni "*list-style-type*", "*list-style-position*" dan "*list-style-image*". Untuk menggunakan ketiga Property tersebut secara bersamaan, kita bisa meringkas penulisannya dengan memanfaatkan Property "**list-style**".

Sintaks-nya :

```
list-style: <list-style-type> <list-style-position> <list-style-image>;
```

Keterangan :

- Value pada Property ini dituliskan secara berurutan seperti terlihat pada sintaks di atas
- Apabila salah satu bagian hilang, akan digunakan nilai default dari masing-masing Property.

Contoh :

FREE TO SHARE

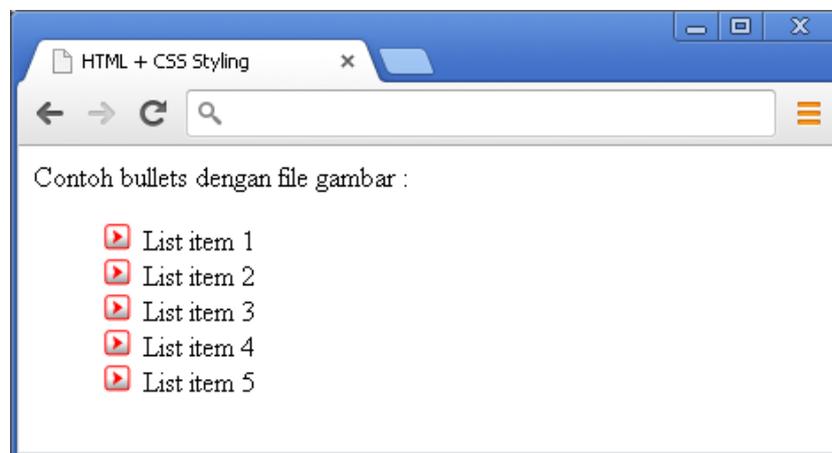
File [List-image.html]

```
<ul>
<li>List item 1</li>
<li>List item 2</li>
<li>List item 3</li>
<li>List item 4</li>
<li>List item 5</li>
</ul>
```

File [list-style.css]

```
ul {
  list-style-image: square inside url('list-arrow.png');
}
```

Hasilnya akan terlihat seperti pada gambar berikut ini: (*bandingkan dengan contoh sebelumnya*)



Keterangan :

- Jika file “list-arrow.png” tidak ditemukan, akan digunakan *bullet* dengan bentuk “square”
- Jika value berupa posisi “inside” tidak dinyatakan, akan digunakan nilai default-nya yakni “outside”.

E. Link Styling

Sebuah link (*hyperlink*) pada halaman web bisa diatur tampilannya dengan menggunakan CSS. Property yang digunakan bisa berupa Property apa saja yang biasa untuk mengatur tampilan elemen HTML, seperti : *background-color*, *color*, *font-size*, dll.

Sebuah link bisa diatur tampilannya sesuai dengan keadaan/status dari link itu sendiri. Status dari link adalah :

- *a:link*, yaitu keadaan normal dimana sebuah elemen HTML menjadi sebuah link.
- *a:visited*, yakni keadaan dimana sebuah link sudah dikunjungi/dilihat tautannya.
- *a:hover*, yakni keadaan dimana pointer mouse melewati sebuah link.
- *a:active*, yakni keadaan dimana link itu sedang di-klik.

Contoh :

FREE TO SHARE

File [link.html]

```
<body>
<ul>
<li><a href="index.html"> Home </a></li>
<li><a href="about.html"> About Us </a></li>
<li><a href="contact.html"> Contact Us</a></li>
</ul>
</body>
```

File [link-style.css]

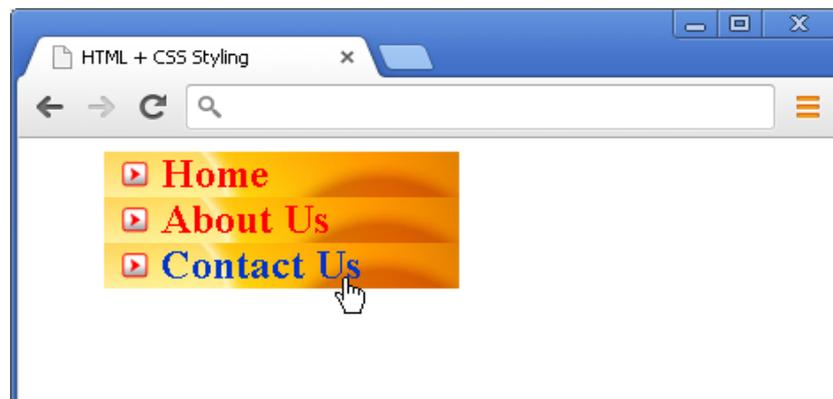
```
ul li {
    list-style:square inside url('list-arrow.png');
    background: url('wallpaper.jpg') no-repeat;
    text-indent:10px;
}
a:link, a:visited {
    color:#ff0000;
    text-decoration:none;
    font-size:24px;
    font-weight:bold;
}
a:hover {
    color:#03c;
}
a:active {
    color: #000;
}
```

Keterangan :

Ketika melakukan pemformatan untuk beberapa keadaan dari sebuah *hyperlink*, maka aturan penulisannya adalah;

- a:hover harus dituliskan setelah a:link, dan a:visited
- a:active harus dituliskan setelah a:hover

Contoh diatas akan menghasilkan tampilan seperti gambar berikut : (*arahkan mouse ke link untuk melihat efek yang dihasilkan*)



Note : Untuk mendapatkan sebuah tampilan menu navigasi yang menarik dan "keren" dengan CSS bisa dibaca pada bagian lain tutorial ini.

FREE TO SHARE

F. Table Styling

Elemen *Table* (tabel) pada halaman web, bisa diformat sedemikian rupa sehingga menghasilkan tampilan yang cantik dan menarik. Pemformatan yang dilakukan bisa berupa penambahan background, mengatur bentuk garis (*border*), text alignment dan lain sebagainya.

Secara umum sebuah elemen tabel pada dokumen HTML terdiri dari beberapa unsur Tag HTML, yakni `<table>`, `<tr>`, `<td>`, `<th>` dan `<caption>`. Masing-masing Tag HTML tersebut mendefinisikan bagian tertentu dari sebuah elemen tabel, sehingga Selector yang digunakan harus disesuaikan dengan bagian mana yang akan dipilih untuk diatur tampilannya.

Pada bagian ini akan dibahas beberapa pemformatan pada tabel yang biasa dilakukan untuk menghasilkan tampilan tabel yang menarik.

Table Border

Dalam pembuatan sebuah tabel tentu tidak terlepas dari bagaimana bentuk dan tampilan dari tabel tersebut, salah satunya adalah bentuk dari garis tabel (*border*). Untuk mengatur bentuk dari *border* kita bisa menggunakan Property "**border**".

Sintaks-nya :

```
border: <width> <style> <color>;
```

Keterangan :

- Selector yang digunakan bisa *table*, *th* ataupun *td*
- Dalam Property ini harus dinyatakan dengan tiga value, yakni : *width*, *style* dan *color*.
- Value berupa *width*, menentukan ketebalan dari border. Nilainya adalah "*thin*", "*medium*" dan "*thick*", atau bisa juga menggunakan nilai dalam satuan pixel (1px, 2px, 3px, dst.)
- Value berupa *style*, menentukan jenis dari border. Nilainya adalah "*hidden*", "*dotted*", "*dashed*", "*solid*", "*double*", "*groove*", "*inset*", "*outset*" dan "*none*".
- Value berupa *color*, menentukan warna dari border. Bisa menggunakan nilai hexadesimal, RGB atau color-name.

Contoh :

File [table.html]

```
<table>
<caption>Daftar Alamat dan Nomor Telepon Perusahaan </caption>
<tr>
  <th>Nama Perusahaan</th>
  <th>Alamat</th>
  <th>Nomor Telp</th>
</tr>
<tr>
  <td>PT. ABC Indonesia</td>
  <td>Jakarta Pusat</td>
  <td>021 - 123123</td>
</tr>
<tr>
  <td>PT. Nusantara Abadi</td>
  <td>Bekasi Barat</td>
```

FREE TO SHARE

```

        <td>021 - 234567</td>
    </tr>
    <tr>
        <td>PT. Citra Perkasa</td>
        <td>Tangerang Selatan</td>
        <td>021 - 567890</td>
    </tr>
</table>

```

File [table-style.css]

```

table, td{
    border:1px solid red;}
th {
    background-color: #C00000;
    border:3px outset red;
    color:#ffffff;}

```

Bentuk file HTML dan CSS diatas, akan menghasilkan tampilan halaman web seperti pada gambar di bawah ini:

Daftar Alamat dan Nomor Telepon Perusahaan

Nama Perusahaan	Alamat	Nomor Telp
PT. ABC Indonesia	Jakarta Pusat	021 - 123123
PT. Nusantara Abadi	Bekasi Barat	021 - 234567
PT. Citra Perkasa	Tangerang Selatan	021 - 567890

Border Collapse

Dari contoh pada gambar di atas, terlihat bahwa tabel tersebut memiliki border dengan bentuk ganda (*double*), padahal kita sudah menentukan bahwa *table* dan *td* memiliki Property border : *1px solid red*. Hal ini disebabkan karena masing-masing sel pada tabel tersebut memiliki jarak, karena memang secara *default* sel pada tabel dalam HTML akan selalu diberi jarak 1px di tiap sisinya. Untuk menghilangkan jarak tersebut, kita bisa menggunakan Property "**border-collapse**".

Sintaks-nya :

```
border: <border-width> <border-style> <border-color>;
```

Keterangan :

- Property ini bisa diletakkan pada Selector *table*
- Value berupa "*collapse*" menghasilkan sel tanpa jarak, sedangkan value berupa "*separate*" menghasilkan tabel tanpa jarak pada tiap selnya.

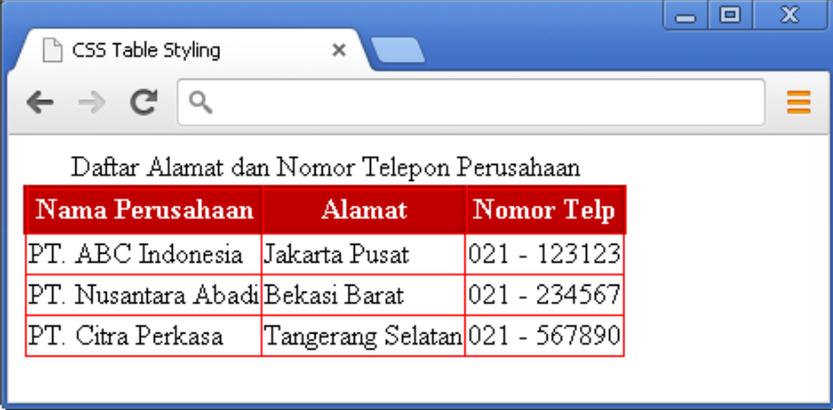
FREE TO SHARE

Contoh :

Edit file CSS pada contoh sebelumnya menjadi :

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

Hasilnya akan terlihat seperti pada gambar di bawah ini :



The screenshot shows a web browser window titled "CSS Table Styling". The page content is titled "Daftar Alamat dan Nomor Telepon Perusahaan". Below the title is a table with three columns: "Nama Perusahaan", "Alamat", and "Nomor Telp". The table contains three rows of data.

Nama Perusahaan	Alamat	Nomor Telp
PT. ABC Indonesia	Jakarta Pusat	021 - 123123
PT. Nusantara Abadi	Bekasi Barat	021 - 234567
PT. Citra Perkasa	Tangerang Selatan	021 - 567890

Table Width dan Table Height

Sebuah tabel dalam halaman web, bisa kita atur ukuran(size)-nya. Dalam hal ini, ukuran bisa berarti lebar dan tinggi dari tabel (*table*) ataupun lebar dan tinggi dari masing-masing sel (*th dan td*). Property yang digunakan adalah Property "**width**" dan "**height**".

Sintaks-nya :

```
width: length | percent;
height: length | percent;
```

Keterangan :

- Value untuk Property *width* dan *height* bisa menggunakan nilai *length* ataupun *persen*.
- Nilai *length* bisa menggunakan nilai berupa angka dalam satuan *px*, *em* ataupun *cm*.
- Nilai berupa *persen* (%) menyatakan ukuran berdasarkan persentasi dari area yang melingkupinya, sehingga didapatkan nilai yang fleksibel.

Contoh :

Pada file tabel.html, edit bagian elemen *table heading* sehingga menjadi seperti berikut :

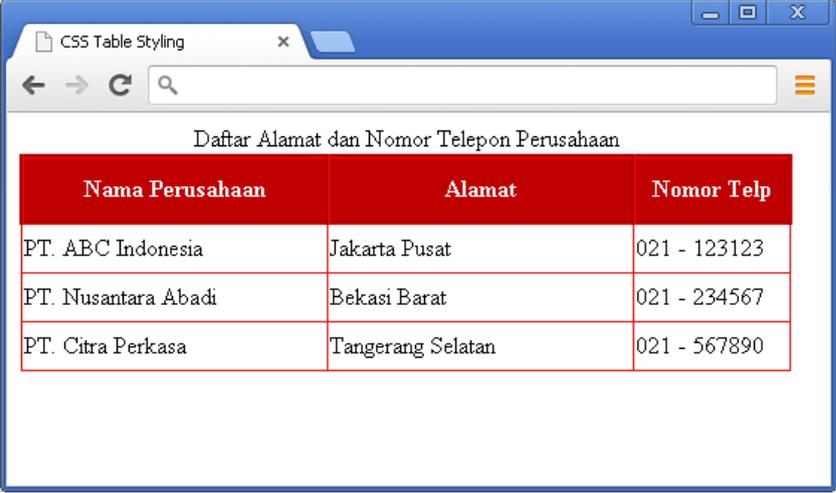
```
<tr>
  <th class="kolom1">Nama Perusahaan</th>
  <th class="kolom2">Alamat</th>
  <th class="kolom3">Nomor Telp</th>
</tr>
```

FREE TO SHARE

Kemudian pada file table-style.css, edit dan tambahkan baris CSS sehingga menjadi seperti berikut ini :

```
td {
    height: 30px;
}
th {
    background-color: #C00000;
    border:3px outset red;
    color:#ffffff;
    height:40px;
}
.kolom1, .kolom2 {
    width:200px;
}
.kolom3 {
    width:100px;
}
```

Hasilnya, tampilan tabel akan terlihat seperti gambar berikut ini :



Nama Perusahaan	Alamat	Nomor Telp
PT. ABC Indonesia	Jakarta Pusat	021 - 123123
PT. Nusantara Abadi	Bekasi Barat	021 - 234567
PT. Citra Perkasa	Tangerang Selatan	021 - 567890

Table Padding

Untuk membuat tampilan data pada tabel lebih nyaman untuk dibaca, maka perlu diberikan jarak dari garis tepi sel dengan teks yang ada di dalam sel tersebut. Property yang digunakan adalah “padding”.

Sintak-nya :

```
padding : length;
```

Keterangan :

- Nilai *length* bisa menggunakan nilai berupa angka dalam satuan *px*, *em* ataupun *cm*.

Contoh :

Edit file CSS pada contoh sebelumnya menjadi :

FREE TO SHARE

```
table, td{
  border:1px solid red;
  border-collapse: collapse;
  padding: 5px;
}
```

Hasilnya akan terlihat seperti pada gambar berikut :



The screenshot shows a web browser window titled "CSS Table Styling". The page content is titled "Daftar Alamat dan Nomor Telepon Perusahaan". Below the title is a table with three columns: "Nama Perusahaan", "Alamat", and "Nomor Telp". The table has a red border and red header cells. The data rows are as follows:

Nama Perusahaan	Alamat	Nomor Telp
PT. ABC Indonesia	Jakarta Pusat	021 - 123123
PT. Nusantara Abadi	Bekasi Barat	021 - 234567
PT. Citra Perkasa	Tangerang Selatan	021 - 567890

Note : Pembahasan lebih lengkap mengenai Property-property yang bisa diterapkan pada elemen table, bisa dibaca pada bagian "CSS Box Model".

Bagian 3 - CSS BOX MODEL

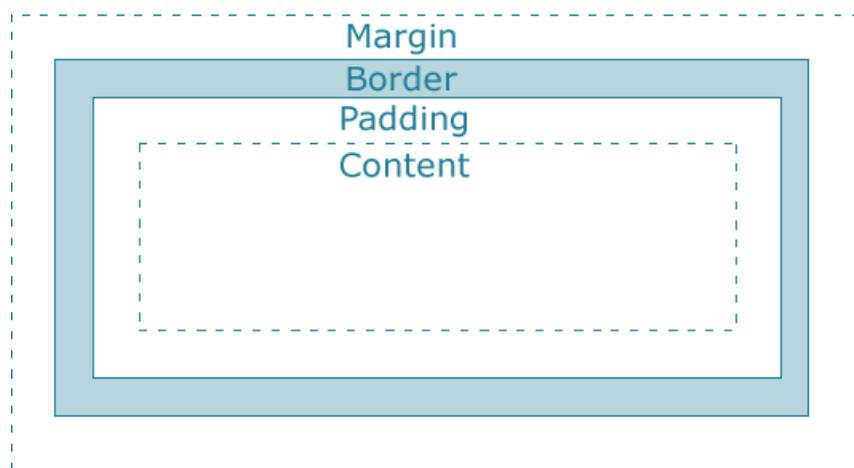
Pada bagian ini, kita akan membahas sebuah bentuk *styling* dalam CSS yang disebut dengan istilah “Box Model”. Apa itu CSS Box Model ? Bagaimana prinsip kerja dan cara implementasinya dalam desain web? Ikuti pembahasannya pada bagian ini dengan seksama dan jangan lupa untuk langsung mencobanya.

Note : Contoh-contoh penerapan CSS pada bagian ini menggunakan metode Internal Style Sheet

A. Apa itu CSS Box Model ?

Pada dasarnya semua elemen HTML bisa dianggap sebagai sebuah “box”. Istilah “Box Model” dalam CSS erat kaitannya ketika kita berbicara tentang desain dan layout website. CSS Box Model pada dasarnya adalah sebuah area yang melingkupi sebuah elemen HTML, yang didalamnya terdiri dari : *margin*, *border*, *padding* dan *content* itu sendiri.

Berikut ini adalah ilustrasi yang bisa menggambarkan apa itu “Box Model” dalam CSS.



Keterangan :

- **Margin**, adalah area kosong yang berada di bagian paling luar dari sebuah elemen HTML. Nilai yang dinyatakan dalam bagian ini, menyatakan jarak atau batas dari elemen HTML tersebut dengan elemen HTML lainnya. **Margin** tidak memiliki nilai background warna, tetapi selalu bersifat *transparent*.
- **Border**, adalah sebuah garis pembatas yang mengelilingi sebuah elemen HTML namun berada didalam wilayah **Margin**.
- **Padding**, adalah sebuah area kosong yang berada diantara Border dan Content dari elemen HTML. Berbeda dengan Margin yang tidak memiliki warna, unsur warna dari Padding didapatkan dari nilai background element HTML yang bersangkutan.
- **Content**, adalah bagian utama dari elemen HTML yang berisi teks atau gambar yang akan ditampilkan.

Perlu diketahui bahwa posisi dari Margin, Border dan Padding berada diluar dari Content, sehingga nilai yang diberikan untuk ketiganya akan berpengaruh menambah luas area dari elemen HTML. Simak baik-baik pembahasan berikut ini!

Menetapkan Ukuran Sebuah Elemen (Width dan Height)

Sebuah elemen HTML dalam halaman web bisa kita atur nilai lebar dan tingginya. Property yang digunakan untuk menetapkan ukuran dari sebuah elemen HTML ini adalah Property “width” dan “height”.

Sintaks-nya :

```
width: auto | length | percent ;  
height: auto | length | percent ;
```

Keterangan :

- Value *default* dari Property “width” dan “height” adalah *auto*, yang berarti browser akan melakukan perhitungan secara otomatis untuk menentukan nilainya.
- Value berupa *length*, bisa menggunakan nilai desimal dengan satuan *px*, *em* atau *cm*, namun yang paling umum digunakan adalah dengan satuan *pixel (px)*.
- Value berupa *percent*, akan diterjemahkan oleh browser dengan menghitung persentasi dari nilai lebar atau tinggi elemen induknya.

Contoh :

```
<!DOCTYPE html>  
<html">  
<head>  
<title>CSS Box Model</title>  
<style>  
.container {  
    background: #06F;  
    width:400px;  
    height:300px;  
}  
.artikel {  
    background: #CCF;  
}  
img {  
    width:340px;  
    height:180px;  
}  
</style>  
</head>  
  
<body>  
<div class="container">  
      
    <div class="artikel"> Ini adalah artikel singkat yang berisi  
    penjelasan tentang gambar di atas </div>  
</div>  
</body>  
</html>
```

Hasilnya akan terlihat seperti pada gambar berikut ini :

FREE TO SHARE



Penjelasan :

- Elemen div dengan atribut *Class="container"* ditentukan dengan nilai *width:400px*; nilai *height:300px*; dan *background:#06F*; sehingga menghasilkan sebuah tampilan area kotak berukuran 400x300px dengan latar belakang warna biru.
- Elemen div dengan atribut *Class="artikel"* ditentukan dengan nilai *background:#CCF*; namun tidak diberikan deklarasi untuk *width* dan *height*, sehingga nilai keduanya adalah *auto*. Dalam hal ini nilai *width* mengikuti nilai *width* dari elemen induk (*Container*), sedangkan untuk nilai *height* mengikuti kebutuhan *content*-nya (teks didalamnya).
- Elemen *img* ditentukan dengan nilai *width:340px*; dan nilai *height:180px*; sehingga menghasilkan tampilan gambar dengan ukuran lebar 340x180px.

B. Margin

Jarak sebuah elemen HTML dengan elemen HTML lain yang bersinggungan disebut dengan "margin". Tiap elemen HTML memiliki empat sisi *margin* yaitu margin atas, margin kanan, margin bawah dan margin kiri.

Luas area dari masing-masing margin bisa diatur dengan menggunakan Property berikut :

- "**margin-top**" - untuk margin atas,
- "**margin-right**" - untuk margin kanan,
- "**margin-bottom**" - untuk margin bawah, dan
- "**margin-left**" - untuk margin kiri.

Sintaks-nya :

```
margin-top: auto | length | percent;  
margin-right: auto | length | percent;  
margin-bottom: auto | length | percent;  
margin-left: auto | length | percent;
```

FREE TO SHARE

Keterangan :

- Value *default* untuk masing-masing Property adalah “0”
- Value yang bisa digunakan untuk mengatur nilai margin bisa berupa nilai auto, length dan percent.
- Value berupa auto, akan menghasilkan nilai margin yang diperhitungkan secara otomatis oleh browser mengikuti keadaan disekeliling elemen.
- Value berupa length, bisa menggunakan satuan *px*, *em* atau *cm*. Namun yang umum digunakan adalah menggunakan satuan *px* (*pixel*).
- Value berupa percent, akan menghasilkan nilai margin yang dihitung dari persentasi area yang berada di sekelilingnya.

Meringkas Penggunaan Property Margin

Jika pada penjelasan sebelumnya kita menentukan nilai margin dengan menggunakan Property untuk masing-masing sisi elemen, berikut ini kita akan menggunakan satu Property untuk mengatur ke-empat sisi dari elemen HTML. Property yang digunakan adalah “margin”.

Sintaks-nya :

```
margin:<margin-top> <margin-right> <margin-bottom> <margin-left>;
```

Keterangan :

- Property “margin” digunakan untuk menentukan nilai margin keempat sisi elemen HTML
- Value dari Property ini bisa terdiri dari satu sampai empat nilai, yang aturannya adalah sebagai berikut :

✓ **Property “margin” dengan 4 value**

```
margin: 10px 20px 30px 40px;
```

- Margin atas = 10px
- Margin kanan = 20px
- Margin bawah = 30px
- Margin kiri = 40px

(dibaca searah jarum jam)

✓ **Property “margin” dengan 3 value**

```
margin: 10px 20px 30px;
```

- Margin atas = 10px
- Margin kanan dan kiri = 20px
- Margin bawah = 30px

✓ **Property “margin” dengan 2 value**

```
margin: 10px 20px;
```

- Margin atas dan bawah = 10px;
- Margin kanan dan kiri = 20px

✓ **Property “margin” dengan 1 value**

```
margin: 10px;
```

- Margin keempat sisinya (atas-kanan-bawah-kiri) = 10px

FREE TO SHARE

Contoh :

Editlah bagian CSS pada contoh sebelumnya sehingga menjadi seperti berikut ini :

```
.container {
  background: #06F;
  width:400px;
  height:300px;
  margin:auto;
}
.artikel {
  background: #CCF;
  margin:10px;
}
img {
  width:340px;
  height:180px;
  margin:20px;
}
```

Hasilnya akan terlihat seperti gambar berikut ini :



Penjelasan :

- Margin pada "container" bernilai *auto*, sehingga menghasilkan nilai margin untuk keempat sisi elemen dengan nilai yang sama. Pada kasus ini menghasilkan tampilan "container" yang berada di tengah-tengah jendela browser.
- Margin pada "artikel" bernilai 10px, sehingga menghasilkan sebuah area kosong (jarak) di sekeliling elemen tersebut dengan sebesar 10px.
- Margin pada "img" bernilai 20px, sehingga menghasilkan sebuah area kosong (jarak) sekeliling sebesar 20px.
- Jarak antara "artikel" dengan "img" adalah sebesar 30px, dihitung dari margin "artikel" 10px ditambah margin "img" 20px.

C. Border

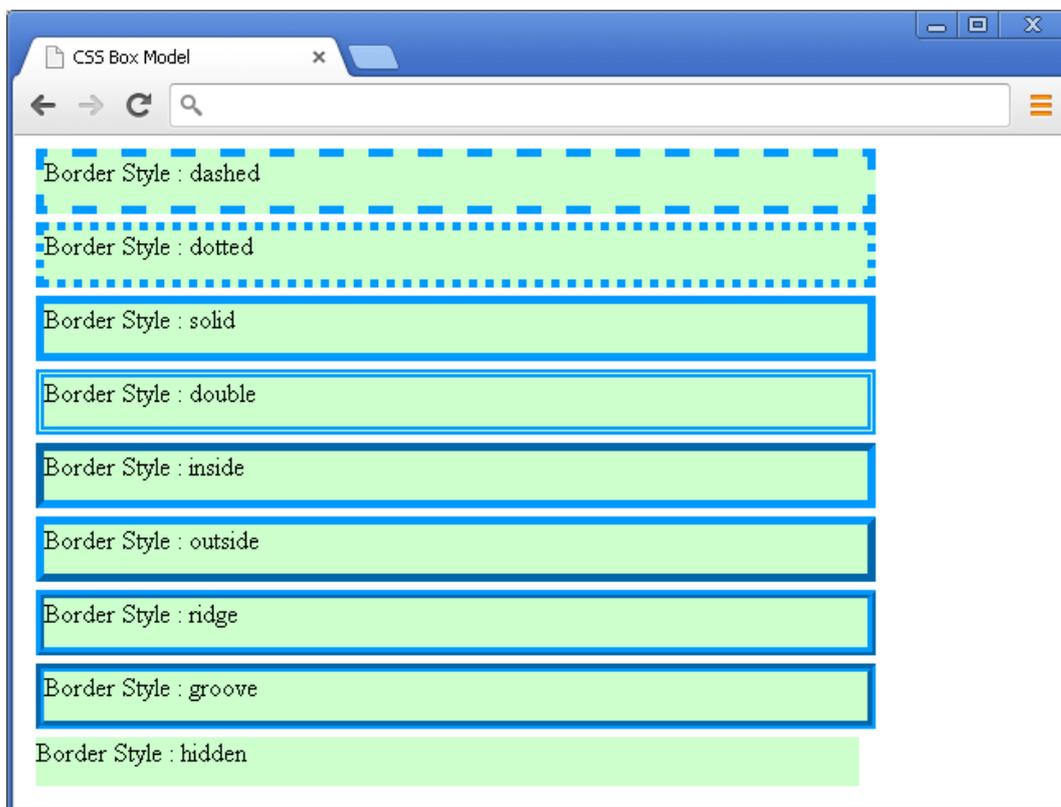
Border adalah sebuah garis pembatas yang mengelilingi sebuah elemen HTML, posisinya diluar Content namun berada di dalam Margin. Karena posisinya berada diluar bagian Content, maka keberadaanya berpengaruh terhadap kebutuhan ruang yang digunakan untuk menampilkan sebuah elemen HTML.

Sebuah elemen HTML memiliki Border untuk keempat sisinya, dimana bentuk, ketebalan dan warnanya bisa kita atur sesuai dengan kebutuhan desain.

Border-style

Property **“border-style”** digunakan untuk menentukan bentuk dari sebuah border. Value dari Property ini bisa menggunakan : *“dotted”*, *“dashed”*, *“solid”*, *“double”*, *“inset”*, *“outset”*, *“ridge”*, *“groove”*, *“hidden”* dan *“none”*.

Berikut adalah contoh bentuk (*style*) dari sebuah border :



Sintaks-nya :

```
border-style: <border-top-style> < border-left-style > < border-  
bottom-style > < border-right-style >;
```

Keterangan :

- Property **“border-style”** digunakan untuk menentukan bentuk (*style*) dari border pada keempat sisi elemen HTML.

FREE TO SHARE

- Property ini tidak memiliki nilai default, sehingga valuenya harus ditentukan.
- Pada Property ini bisa diberikan 1 sampai 4 buah value, yang masing-masingnya menentukan bentuk border untuk tiap sisi elemen HTML. Aturannya adalah sebagai berikut :

✓ **Border-style dengan 4 value**

```
border-style: dashed solid dotted solid;
```

- Border-top-style = dashed
- Border-right-style = solid
- Border-bottom-style = dotted
- Border-left-style = solid

(dibaca searah jarum jam)

✓ **Border-style dengan 3 value**

```
border-style: dashed solid dotted;
```

- Border-top-style = dashed
- Border-right-style dan border-left-style = solid
- Border-bottom-style = dotted

✓ **Border-style dengan 2 value**

```
border-style: dashed solid;
```

- Border-top-style dan border-bottom-style = dashed
- Border-right-style dan border-left-style = solid

✓ **Border-style dengan 1 value**

```
border-style: dashed;
```

- Border-style untuk keempat sisi = dashed

Selain menggunakan cara di atas, kita juga bisa menentukan bentuk border secara spesifik untuk masing-masing sisi border dengan Property berikut :

- **Border-top-style** : menentukan style / bentuk border bagian sisi atas
- **Border-right-style** : menentukan style / bentuk border bagian sisi kanan
- **Border-bottom-style** : menentukan style / bentuk border bagian sisi bawah
- **Border-left-style** : menentukan style / bentuk border bagian sisi kiri

Border-width

Property “**border-width**” digunakan untuk mengatur ketebalan dari border. Ketebalan sebuah border bisa dinyatakan dengan nilai “*thin*”, “*medium*” dan “*thick*”, tapi bisa juga dinyatakan dalam satuan pixel.

Sintaks-nya :

```
border-width : thin | medium | thick | <length>;
```

FREE TO SHARE

Keterangan :

- Property “border-width” digunakan untuk menentukan ketebalan border untuk keempat sisi elemen HTML.
- Value *default* untuk Property ini adalah *medium*.
- Value berupa “*thin*” menghasilkan sebuah border yang tipis, Value berupa “*thick*” menghasilkan sebuah border yang tebal, dan Value berupa “*medium*” menghasilkan sebuah border dengan ketebalan antara *thin* dan *thick*.
- Selain menggunakan ketiga Value tersebut, bisa juga menggunakan nilai *length* dengan satuan *pixel* seperti 1px, 5px, 10px, dst.
- Property ini bisa diberikan 1 sampai 4 buah value, yang masing-masingnya mendefinisikan ketebalan dari border untuk masing-masing sisi. Aturan penggunaannya sama dengan penggunaan Property “border-style”.

Selain menggunakan Property “border-width” untuk menentukan ketebalan keempat sisi border, kita juga bisa menentukan ketebalan border secara spesifik untuk masing-masing sisi border dengan Property berikut :

- **Border-top-width** : menentukan ketebalan border bagian sisi atas
- **Border-right-width** : menentukan ketebalan border bagian sisi kanan
- **Border-left-width** : menentukan ketebalan border bagian sisi bawah
- **Border-bottom-width** : menentukan ketebalan border bagian sisi kiri

Border-color

Property “border-color” digunakan untuk menentukan warna dari border pada elemen HTML. Seperti pada penentuan warna pada Property lain dalam CSS, Property ini juga bisa diberikan value dengan nilai warna *Hexadecimal*, *RGB* ataupun *color name*.

Sintaks-nya :

```
border-color : color-value | transparent;
```

Keterangan :

- Property “border-color” digunakan untuk menentukan warna border sebuah elemen pada keempat sisinya.
- Property ini tidak bisa berdiri sendiri, harus diawali dengan penentuan ketebalan border dengan menggunakan Property “border-width”.
- Value untuk Property ini bisa menggunakan nilai warna *Hexadecimal*, *RGB* ataupun *color name*, dimana nilai defaultnya adalah hitam (*black*).
- Selain menggunakan *color-value*, bisa juga digunakan nilai *transparent* yang menghasilkan border tidak terlihat.
- Property ini bisa diberikan 1 sampai 4 buah value, yang masing-masingnya mendefinisikan warna dari border untuk masing-masing sisi. Aturan penggunaannya sama dengan penggunaan Property “border-style” dan “border-width”.

Selain menggunakan Property “border-color”, kita juga bisa mendefinisikan warna border untuk masing-masing sisi border dengan Property :

- **Border-top-color** : menentukan warna border bagian sisi atas
- **Border-right-color** : menentukan warna border bagian sisi kanan
- **Border-left-color** : menentukan warna border bagian sisi bawah
- **Border-bottom-color** : menentukan warna border bagian sisi kiri

Meringkas Penggunaan Property Border

Secara umum sebuah border dalam CSS terdiri dari tiga unsur, yakni :

- ✓ Bentuk border – yang dinyatakan dengan Property “border-style”,
- ✓ Ketebalan border – yang dinyatakan dengan Property “border-width”, dan
- ✓ Warna border – yang dinyatakan dengan Property “border-color”.

Ketiga unsur tersebut harus dinyatakan untuk mendefinisikan sebuah border yang akan diterapkan pada elemen HTML. Namun, seringkali kita hanya memformat tampilan elemen HTML dengan border yang cukup sederhana. Oleh karenanya, CSS menyediakan cara meringkas penulisan baris kode CSS untuk memformat sebuah border dengan satu Property yaitu “border”.

Sintaks-nya :

```
border: <border-width> <border-style> <border-color>;
```

Keterangan :

- Property “border” digunakan untuk memformat border dengan mendefinisikan ketiga unsur border secara bersamaan, yakni bentuk, ketebalan dan warna border.
- Dengan menggunakan Property ini, maka nilai yang didefinisikan adalah nilai masing-masing unsur border pada keempat sisinya.
- Pada Property ini, value untuk *border-style* harus dinyatakan secara spesifik. Sedangkan apabila *border-width* dan *border-color* tidak dinyatakan, akan digunakan nilai default dari masing-masing Property tersebut.

Contoh :

Editlah bagian CSS pada contoh sebelumnya sehingga menjadi seperti berikut ini :

```
.container {
  background: #06F;
  width:400px;
  height:300px;
  margin:auto;
  border:10px ridge #F30;
}
.artikel {
  background: #CCF;
  margin:0 20px;
  border-style:dashed dotted;
  border-width:2px;
  border-color:#FFF;
}
```

FREE TO SHARE

```
img {
  width:340px;
  height:180px;
  margin:20px;
  border-style:solid;
  border-width:5px 10px;
  border-color:#000000;
}
```

Hasilnya akan terlihat seperti pada gambar di bawah ini :



Penjelasan :

- Border pada “img” dinyatakan dengan { **border-style:solid; border-width:5px 10px; border-color:#000000;** } sehingga menghasilkan border dengan bentuk solid berwarna hitam, dengan ketebalan sisi atas-bawah 5px, dan kanan-kiri 10px.
- Dengan deklarasi CSS pada “img” tersebut, menyebabkan bertambahnya luas area pada elemen *image* yang sebelumnya 340x180 *pixel* menjadi 400x230 *pixel*. Perhitungannya adalah sebagai berikut :
 Lebar elemen; **340** (*width*) + **40** (*margin-right + margin-left*) + **20** (*border-right-width + border-left-width*) = **400px**.
 Tinggi elemen; **180** (*height*) + **40** (*margin-top + margin-bottom*) + **10** (*border-top-width + border-bottom-width*) = **230px**.
- Border pada “container” dinyatakan dengan { **border:10px ridge #F30;** }
- Border pada “artikel” dinyatakan dengan { **border-style:dashed dotted; border-width:2px; border-color:#FFF;** }

FREE TO SHARE

D. Padding

Padding adalah area kosong yang mengelilingi sebuah elemen HTML dan memberikan jarak antara Content dengan Border. Sama halnya dengan Margin dan Border, Padding yang letaknya berada di bagian luar elemen HTML akan berpengaruh terhadap kebutuhan ruang untuk penempatan sebuah elemen HTML.

Padding tidak memiliki unsur warna, namun keberadaannya terpengaruhi oleh unsur warna dari latar belakang sebuah elemen yang dinaunginya.

Tiap elemen HTML memiliki empat sisi Padding yaitu padding atas, padding kanan, padding bawah dan padding kiri. Masing-masing besaran luas padding bisa ditentukan dengan menggunakan Property :

- **“padding-top”** - untuk padding atas,
- **“padding-right”** - untuk padding kanan,
- **“padding-bottom”** - untuk padding bawah, dan
- **“padding-left”** - untuk padding kiri.

Sintaks-nya :

```
padding-top: length | percent;  
padding-right: length | percent;  
padding-bottom: length | percent;  
padding-left: length | percent;
```

Keterangan :

- Value *default* untuk masing-masing Property adalah **“0”**
- Value yang bisa digunakan untuk mengatur nilai margin bisa berupa nilai *length* dan *percent*.
- Value berupa *length*, bisa menggunakan satuan *px*, *em* atau *cm*. Namun yang umum digunakan adalah menggunakan satuan *px (pixel)*.
- Value berupa *percent*, akan menghasilkan nilai margin yang dihitung dari persentasi area yang berada di sekelilingnya.

Meringkas Penggunaan Property Padding

Agar penulisan nilai padding sebuah elemen HTML lebih ringkas, bisa digunakan Property **“padding”**.

Sintaks-nya :

```
padding:< padding-top> < padding-right> <padding-bottom>  
<padding-left>;
```

Keterangan :

- Property **“padding”** digunakan untuk menentukan nilai Padding dari keempat sisi elemen HTML
- Value dari Property ini bisa terdiri dari satu sampai empat nilai, yang penggunaannya sama dengan Property Margin dan Property Border.

FREE TO SHARE

Contoh :

Editlah bagian CSS untuk Selector “.artikel” pada contoh sebelumnya sehingga menjadi seperti berikut ini :

```
.artikel {  
    background: #CCF;  
    margin:0 20px;  
    border-style:dashed dotted;  
    border-width:2px;  
    border-color:#FFF;  
    padding:5px 15px;  
}
```

Hasilnya akan terlihat seperti pada gambar berikut ini :



Penjelasan :

Pada gambar di atas, terlihat bahwa diantara Content (teks) dan Border terdapat jarak (Padding). Padding bagian atas dan bawah sebesar 5px, dan Padding bagian kanan dan kiri sebesar 15px.

Dalam mendesain sebuah halaman web, penggunaan nilai padding sangat berpengaruh dalam hal kenyamanan user saat membaca artikel yang kita sajikan. Sebuah artikel yang diberikan jarak dengan garis tepinya kan lebih nyaman dibaca daripada artikel yang posisinya menempel dengan garis tepinya.

Pertimbangkan besaran nilai Padding dengan konsep desain web yang diterapkan, sehingga akan menghasilkan sebuah tampilan halaman web yang cantik dan nyaman untuk dibaca.

E. Outline

Outline adalah sebuah garis yang mengelilingi sebuah elemen HTML, letaknya berada diluar Border. Seperti halnya Border, Outline terdiri dari 3 unsur dasar yakni ketebalan, bentuk dan warna. Masing-masing unsur tersebut didefinisikan dengan Property :

- **outline-width** : untuk menentukan ketebalan outline,
- **outline-style** : untuk menentukan bentuk dari outline,
- **outline-color** : untuk menentukan warna dari outline.

Penulisan deklarasi untuk membentuk sebuah Outline pada elemen HTML juga bisa diringkas menggunakan sebuah Property, yaitu Property “**outline**”.

Sintaks-nya :

```
outline : <outline-width> <outline-style> <outline-color>;
```

Penggunaan dan value dari masing-masing Property tersebut sama dengan penggunaan dan value dari Property yang digunakan untuk mengatur Border.

Outline vs Border

Secara umum Outline dan Border memiliki kesamaan, yaitu membentuk garis tepi yang mengelilingi sebuah elemen HTML. Value yang digunakan untuk mendefinisikan masing-masing Property-nya pun menggunakan nilai yang sama. Perbedaannya dari keduanya (Outline dan Border) adalah pada posisi atau letaknya dan efek yang dipengaruhinya.

Border mengelilingi sebuah elemen dan berada diluar Content, diantara Margin dan Padding, dan keberadaannya membutuhkan sebuah ruang khusus sehingga berpengaruh terhadap kebutuhan ruang untuk penempatan sebuah elemen HTML.

Berbeda dengan Border, Outline mengelilingi sebuah elemen dan berada di luar Border namun keberadaannya tidak membutuhkan sebuah ruang khusus, sehingga tidak berpengaruh terhadap kebutuhan ruang untuk penempatan sebuah elemen HTML.

Perhatikan contoh berikut ini !

```
<!DOCTYPE html>
<html>
<head>
<title>CSS Box Model</title>
<style>
.gbr1 {
  width:200px;
  padding:10px;
  border:5px solid #000000;
  outline:5px solid #00FF00;
}
.gbr2 {
  width:150px;
  padding:10px;
  border:5px solid #F00;
  outline:5px dotted #0033CC;
}
```

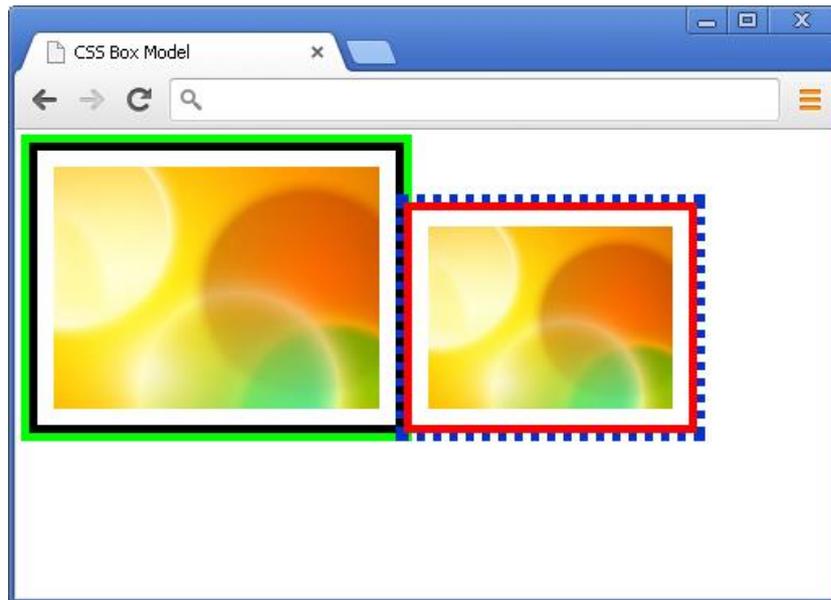
FREE TO SHARE

```
</style>
</head>

<body>

</body>
</html>
```

Hasilnya akan terlihat seperti pada gambar di bawah ini :



Penjelasan :

Nilai Border-width ditentukan sebesar 5px, sedangkan nilai Outline-width ditentukan 5px juga. Border dari masing-masing elemen gambar membutuhkan ruang khusus, sehingga keduanya hanya bersinggungan saja. Berbeda dengan Outlin yang tidak membutuhkan ruang khusus, terlihat bahwa Outline dari masing-masing elemen gambar saling memasuki area dari elemen yang bersinggungan.

Ringkasnya, keberadaan sebuah Outline pada elemen HTML tidak mempengaruhi kebutuhan area yang akan digunakan untuk menempatkan sebuah elemen HTML tersebut.