

# Kupas Tuntas CSS

**Hafid Mukhlisin**

hafidmukhlisin@gmail.com

<http://www.facebook.com/hafidm>

## ***Lisensi Dokumen:***

*Copyright © 2003-2013 IlmuKomputer.Com*

*Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.*

Melanjutkan artikel sebelumnya yang dimuat di web ilmu komputer berjudul Kupas Tuntas HTML. Maka kurang lengkap rasanya jika materi tentang CSS tidak disertakan. Ibarat HTML itu surat maka CSS adalah perangkonya.

## **Pendahuluan**

### **Mengapa Menggunakan CSS?**

Jika HTML digunakan untuk membuat struktur tampilan web. Maka CSS digunakan untuk mempercantik tampilan web. Nah lupakan CSS jika kita ingin membuat tampilan web kita biasa aja (baca: jelek ☺)

CSS focus terhadap make-up laman web. Sedangkan HTML focus terhadap konten web. Pemisahan ini membuat kita lebih mudah dan efisien dalam mengelola web kita.

### **Sejarah CSS**

<b>Tahun</b>	<b>Keterangan</b>
<b>1996</b>	CSS level 1 menjadi W3C Recommendation pada 17 December 1996.
<b>1998-99</b>	CSS level 2 menjadi W3C Recommendation pada 11 January 1999 (sumber lain : 12 Mei 1998). Level ini ditandai dengan ditambahkannya dukungan untuk media (printers), downloadable fonts, element-positioning dan tables.

**2001** Draft CSS level 3 dipublikasikan oleh W3C pada 23 Mei 2001 dan terus dikembangkan hingga sekarang. Ciri dari CSS Level 3 ini adalah dikembangkan secara modular.

### **Implementasi CSS Saat ini (2013)**

Banyak fitur yang dimasukkan sejak spesifikasi pertama di tahun 2000, meskipun demikian tidak banyak browser yang mendukung fitur tersebut hingga tahun 2006.

Saat ini browser-browser besar (Opera, Firefox, Chrome, IE dan Safari) secara parsial dan bertahap (baca: lambat) mulai menyempurnakan dukungannya pada CSS3

### **Apa itu CSS?**

CSS singkatan dari Cascading Style Sheets, yaitu sebuah bahasa berbasis text yang digunakan untuk memformat tampilan dari halaman web yang dibuat dengan HTML, misalnya dalam hal pewarnaan, ukuran, posisi dsb. Jika HTML adalah dasar dan batu bata yang membentuk struktur rumah, CSS adalah plester dan cat yang menghiasinya.

Dengan CSS, memungkinkan kita untuk memberikan efek-efek khusus pada isi halaman web secara lebih spesifik. Memungkinkan kita mengubah tampilan beberapa halaman web sekaligus dengan sedikit kode.

Sebenarnya, CSS bukan bahasa pemrograman seperti Javascript, bukan juga bahasa markup seperti HTML. Jadi jangan dibandingkan ☺

CSS merupakan W3C Recommendation.

### **Mulai Menggunakan CSS**

#### **Cara Berfikir**

Jika kita ingin memformat tampilan web kita dengan menggunakan CSS, maka cara berfikirnya:

1. Identifikasi elemen HTML yang akan di format. Bisa melalui nama tagnya (p, a, h1), bisa melalui nama class-nya, bisa melalui nama id nya dst.
2. Tentukan format elemen HTML yang diinginkan seperti warna, ukuran, font, border dst
3. Tentukan nilai dari format tersebut.

Misal: kita ingin semua konten dalam elemen <p> (paragraf) berwarna hijau. Maka:

1. Elemen htmlnya p  
p{ }
2. Formatnya warna  
p{ color: }

3. Nilai dari formatnya hijau  
`p{ color: green; }`

Jika kode tersebut diaplikasikan kepada dokumen HTML, maka semua konten atau teks dalam elemen paragraph akan berwarna hijau. Mudah bukan?

### Skema Dasar CSS

Berikut ini skema dasar dari perintah / kode CSS:

```
selector {  
  property1:value;  
  property2:value;  
  property3:value;  
}
```

Keterangan:

- ❖ Selector => elemen HTML yang akan dikenai aturan CSS, bisa berisi nama elemen, atau identitas lain seperti ID, nilai atribut class.
- ❖ Property => jenis format yang akan kita implementasikan untuk elemen HTML terpilih, misalnya warna text, warna background, jenis font, warna border dsb.
- ❖ Value => Nilai masing-masing property dari elemen HTML terpilih. Nilai tergantung dari properti yang digunakan, contoh properties yang berhubungan dengan colour dapat berisi hexadecimal colours seperti #336699, RGB seperti rgb(12,134,22) atau nama colour seperti red, green atau blue. Properties yang berhubungan dengan position maka nilainya dapat berisi margins, width, height

Berikut contoh sederhana:

```
p {  
  margin: 5px;  
  font-family: arial;  
  color: blue;  
}
```

Artinya: semua elemen P pada halaman HTML akan diformat tampilannya sebagai berikut:

- jarak antara elemen P dengan elemen lainnya adalah 5 pixel
- Jenis huruf atau font yang digunakan adalah Arial
- Warna huruf yang digunakan adalah biru

### Spasi Kosong Pada CSS

Sama dengan HTML spasi kosong akan diabaikan di CSS artinya nggak ada efek apapun kecuali kode CSSmu akan lebih mudah dibaca.

```
p {  
  margin: 5px;
```

```
font-family: arial;  
color: blue;  
}
```

Berfungsinya sama dengan:

```
p {margin: 5px;font-family: arial;color: blue;}
```

### Komentar pada CSS

Komentar dalam kode CSS digunakan agar kode CSS kita lebih mudah dibaca atau difahami baik oleh kita sendiri maupun oleh orang lain yang hendak memodifikasinya.

Untuk menulis komentar pada CSS adalah dengan cara meletakkannya diantara tanda `/*` dan `*/`.

```
/* These are basic element selectors */  
selector{  
    property1:value;  
    property2:value;  
    property3:value;  
}
```

atau bisa juga:

```
selector{  
    property1:value;  
    /*  
    property2:value;  
    property3:value;  
    */  
}
```

### Pengelompokan Selector

Kita juga dapat mengelompokkan selector-selector yang memiliki aturan CSS yang sama dalam satu blok kode. Contoh:

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

Selector `p` dan `h1` diatas memiliki aturan CSS yang sama, maka bisa digabung menjadi:

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

### Tipe Dasar Selector

Contoh kode HTML:

```
<p class="paragraf" id="contoh">Paragraph</p>
```

Ada 3 jenis selector dasar pada CSS yaitu: selector Element, Class dan ID

1) Element selector

```
p {}
```

artinya: semua elemen **p** akan dikenai aturan CSS

2) Class selector

```
.paragraf {}
```

semua elemen dengan class “paragraf” akan dikenai aturan CSS. Contoh: `<p class=”paragraf”> </p>` , `<a class=”paragraf”></a>`, `<div class=”paragraf”></div>`

3) ID selector

```
#contoh {}
```

semua elemen dengan id “contoh” akan dikenai aturan CSS. Contoh: `<p class=”contoh”> </p>` , `<a class=”contoh”></a>`, `<div class=”contoh”></div>`

Dalam prakteknya, untuk mendapatkan elemen HTML yang spesifik maka kita bisa mengkombinasi ketiganya:

- ✓ `p.warning {}` artinya semua elemen p dengan class warning
- ✓ `div#example {}` artinya semua elemen div dengan id example.

### Penyingkatan pada CSS

Agar lebih efisien kita juga bisa menyingkat kode CSS, sehingga kode CSS kita akan lebih simple dan efisien.

- ✓ Contoh pada properti border:

```
border-width: 2px;  
border-style: solid;  
border-color: black;
```

bisa disingkat menjadi

```
border: 2px solid black;
```

- ✓ Contoh lain pada properti margin:

```
div.foo {  
  margin-top: 1em;  
  margin-right: 1.5em;  
  margin-bottom: 2em;  
  margin-left: 2.5em;
```

```
}
```

Disingkat menjadi

```
div.foo {  
  margin: 1em 1.5em 2em 2.5em;  
}
```

✓ Contoh lain

**margin:3px 2px 4px;** artinya margin atas: 3px, kanan dan kiri: 2px, bawah: 4px

**margin:4px 2px;** artinya margin atas dan bawah: 4px, kiri dan kanan: 2px

**margin:1px;** artinya margin atas, kanan, bawah, dan kiri : 1px;

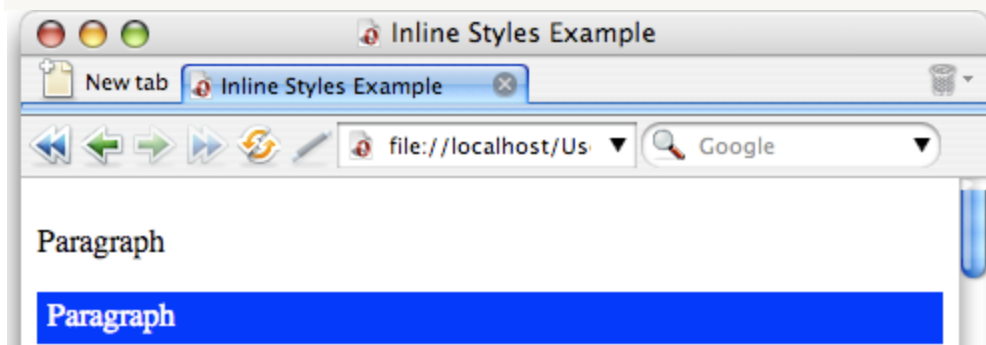
### Mengaplikasikan CSS pada HTML

Ada 3 cara mengaplikasikan aturan CSS ke dokumen HTML, yaitu : inline styles, embedded styles dan external style sheets.

#### *Inline styles*

Kita dapat mengaplikasikan styles kepada element tertentu menggunakan atribut style, contoh:

```
<p>Paragraph</p>  
<p style="background:blue; color:white; padding:5px;">Paragraph</p>
```



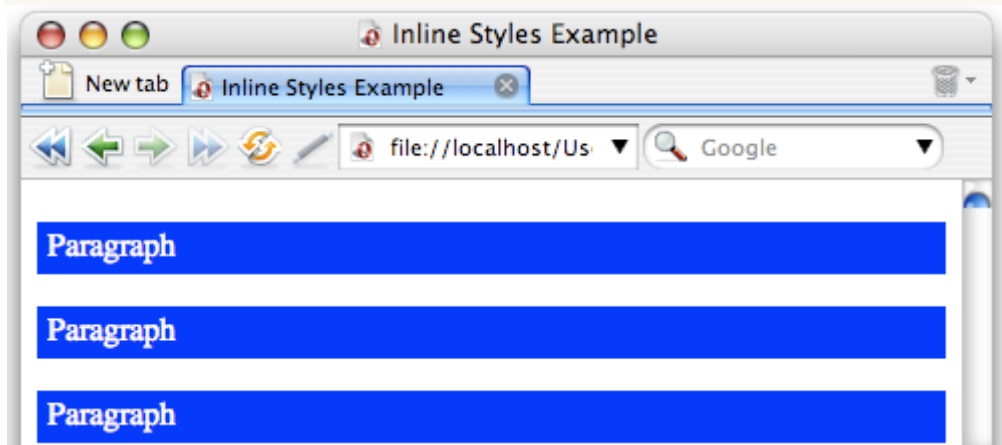
- ❖ Keuntungan menggunakan CSS inline style adalah memaksa browser untuk menggunakan aturan CSS tersebut adapun aturan CSS dibagian lain terkait elemen tersebut akan diabaikan.
- ❖ Kerugiannya adalah sulit maintenaceny. Karena style ada pada masing-masing elemen. Dan tentunya tidak efisien.

#### *Embedded styles*

Deklarasi CSS diletakkan pada elemen HEAD dengan menggunakan Elemen STYLE

```
<html>  
<head>  
<style type="text/css" media="screen">
```

```
p {  
    color:white;  
    background:blue;  
    padding:5px;  
}  
</style>  
<head>  
<body>  
<p>Paragraph</p>  
<p>Paragraph</p>  
<p>Paragraph</p>  
</body>  
</html>
```



- ❖ Keuntungannya, lebih mudah maintenance karena aturan CSS terpusat di satu tempat.
- ❖ Kekurangannya, kode HTML akan tampak lebih kompleks karena juga berisi kode-kode HTML, tentu saja hal ini akan mengurangi performance saat loading web..

### ***External style sheets***

Deklarasi CSS dapat diletakkan pada file CSS tersendiri diluar file HTML, kemudian file CSS hubungkan dengan file HTML dengan menggunakan elemen LINK.

Sebagai contoh, kita perlu membuat dulu file CSS yang berisi kode2 CSS.

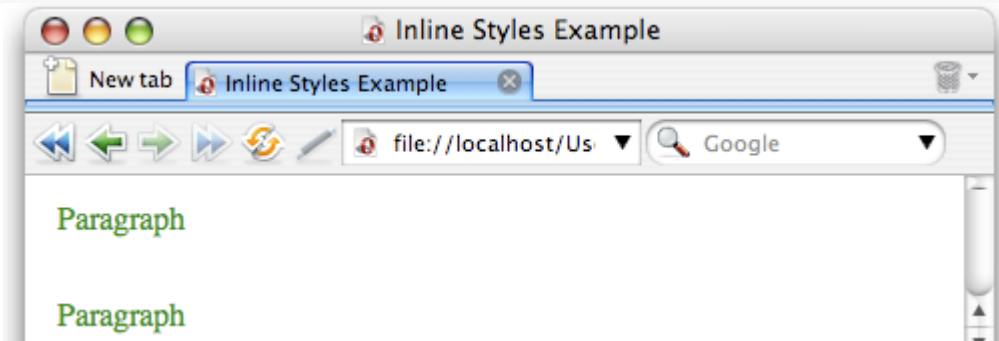
```
p {  
    color:green;  
    padding:5px;  
}
```

Lalu kita simpan, misalnya dengan nama style.css

Kemudian kita membuat halaman HTML yang mana pada elemen HEAD kita tambahkan kode berikut:

```
<html>  
<head>
```

```
<link rel="stylesheet" href="style.css" type="text/css"
media="screen">
<head>
<body>
<p>Paragraph</p>
<p>Paragraph</p>
</body>
</html>
```



Selain menggunakan elemen LINK, ada cara lain untuk mendeklarasikan external Style Sheet yaitu dengan menggunakan property @importing yang diletakkan pada elemen STYLE.

```
<style type="text/css" media="screen">
  @import url("style.css");

  ...other import statements or CSS styles could go here...
</style>
```

## CSS Dasar

### CSS Warna

Warna dalam CSS dapat direpresentasikan dengan beberapa metode:

- ✓ Hexadecimal colors
- ✓ RGB colors
- ✓ Predefined/Cross-browser color names
- ✓ dll (RGBA colors, HSL colors, dan HSLA colors)

#### Hexadecimal Colors

Hexadecimal color dituliskan dengan: #RRGGBB, RR (red), GG (green) dan BB (blue) merupakan bilangan hexadecimal (0 sampai FF) yang mengacu pada warna spesifik.

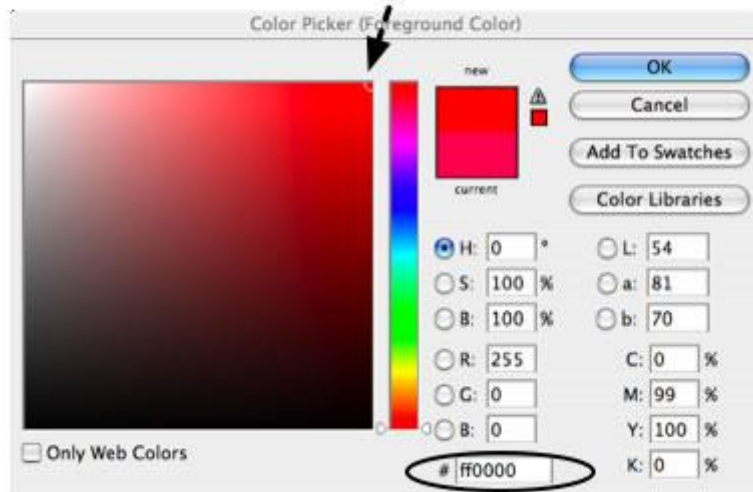


Contoh untuk membuat warna biru maka kita set nilai komponen BB/Blue dengan maksimal yaitu FFF, sedangkan nilai komponen lainnya kita set 0. Sehingga #0000ff akan dirender sebagai blue.

```
p{  
    background-color:#ff0000;  
}
```

Gunakan color picker

Tools like Photoshop will allow you to choose a color, such as red, and will give you the hex value.



## RGB Colors

RGB color dituliskan dengan: `rgb(red, green, blue)`. Masing-masing parameter (red, green, and blue) mendefinisikan ketajaman dari warna dan bernilai integer antara 0 and 255 atau persentase (dari 0% hingga 100%).

Contoh untuk membuat warna biru maka kita set nilai komponen Blue dengan maksimal yaitu 255, sedangkan nilai komponen lainnya kita set 0. Sehingga `rgb(0,0,255)` akan dirender sebagai blue. Atau bisa juga dengan menggunakan persen: `rgb(0%,0%,100%)`.

```
p{  
    background-color:rgb(0,0,255);  
}
```

## Predefined/Cross-browser Color Names

147 nama warna didefinisikan pada spesifikasi HTML dan CSS color (17 standard colors plus 130 more). 17 standard colors tersebut adalah: aqua, black, blue, fuchsia, gray, grey, green, lime, maroon, navy, olive, purple, red, silver, teal, white, dan yellow.

```
p{
    background-color:olive;
}
```



Lebih lengkapnya silahkan lihat di cheatsheet CSS Color ☺

## Selector Tingkat Lanjutan

Disamping selector element, class and id, ada cara lain selector lain yang bisa menunjuk suatu elemen HTML dengan kriteria yang lebih spesifik lagi.

### Universal selectors - \*

Menunjuk setiap element HTML pada halaman tersebut.

```
* {
    border: 1px solid #000000;
}
```

### Attribut selectors

```
img[alt] {
    border: 1px solid #000000;
}
```

Semua elemen IMAGE yang memiliki atribut ALT akan dikenai aturan CSS tsb.

### Selecting by attribute value

Contoh untuk menunjuk  kita bisa menggunakan

```
img[src="alert.gif"] {
```

```
border: 1px solid #000000;  
}
```

Sayang sekali tidak support oleh < IE6 ☺

### *Selecting based on substrings within the attribute value*

Menunjuk sesuatu yang nilai atributnya diawali dengan teks tertentu, misal pada kasus yang sama dengan sebelumnya

```
img[src^="alert"] {  
border: 1px solid #000000;  
}
```

Menunjuk sesuatu yang nilai atributnya diakhiri dengan teks tertentu, misal pada kasus yang sama dengan sebelumnya

```
img[src$="gif"] {  
border: 1px solid #000000;  
}
```

Terakhir, menunjuk sesuatu yang nilai atributnya terdapat teks tertentu, misal pada kasus yang sama dengan sebelumnya

```
img[src*="ert"] {  
border: 1px solid #000000;  
}
```

### *Selecting based on delimited items within the attribute value*

Contohnya jika kita memiliki elemen HTML dengan class yang lebih dari satu:

```
<p class="featured archive english"></p>
```

Maka kita bisa menggunakan salah satu cara berikut untuk menunjuk ke elemen HTML diatas:

```
p[class~="featured"]  
p[class~="archive"]  
p[class~="english"]
```

Contohnya yang lain adalah misalnya kita memiliki elemen HTML dengan nama class yang panjang dan berdelimiter -:

```
<p id="english-daily-feature"></p>
```

Maka kita bisa menggunakan kode:

```
p[id="english"]
```

Namun **tidak** bisa menggunakan kode:

```
p[id|"daily"] atau p[id|"feature"]
```

Tidak disupport oleh <IE 8

### ***Child Selector***

Untuk menunjuk ke anak dari elemen tertentu. Contoh: kita ingin memformat “Teks 2” saja dengan warna hijau.

```
<div>
  <p><span>Teks 1</span></p>
  <span>Teks 2</span>
</div>
```

Maka gunakan

```
div > span {
  color: green;
}
```

Artinya anak dari elemen div yaitu elemen span akan diformat warnanya menjadi hijau. Adapun “Teks 2” bukan merupakan anak dari div melainkan cucu. Nah lo ☺

Tidak support < IE 6

### ***Descendent selector***

Hampir sama dengan child selector, hanya saja descendent akan menunjuk semua child dibawahnya meski tidak langsung. Jadi fungsinya untuk menunjuk ke anak cucu (keturunan ☺) dari elemen tertentu. Contoh: kita ingin memformat semua konten dalam span dengan warna hijau.

```
<div>
  <p><span>Teks 1</span></p>
  <span>Teks 2</span>
</div>
```

Maka gunakan

```
div span {
  color: green;
}
```

Note that child selectors are not supported in IE6 or below.

### ***Adjacent sibling selector***

Menunjuk pada elemen tertentu yang terletak TEPAT setelah elemen tertentu pada level hirarki elemen yang sama.

```
<h2>My heading</h2>
<p>My first paragraph</p>
<p>My second paragraph</p>
<p>My third paragraph</p>
<p>My fourth paragraph</p>
<p>My fifth paragraph</p>
```

Untuk menunjuk ke elemen P yang berisi "My first paragraph" kita bisa gunakan:

```
h2 + p {
  ...
}
```

Tidak support < IE 6

### **General Sibling Selectors**

Sama dengan Adjacent Sibling namun selector menunjuk pada semua elemen tertentu yang terletak setelah elemen tertentu pada level hirarki elemen yang sama

```
<h2>My heading</h2>
<p>My first paragraph</p>
<p>My second paragraph</p>
<p>My third paragraph</p>
<p>My fourth paragraph</p>
<p>My fifth paragraph</p>
```

Untuk menunjuk ke semua elemen P setelah elemen H2, kita bisa gunakan:

```
h2 + p {
  ...
}
```

Tidak support < IE 8

### **Pseudo-class selectors**

CSS pseudo-class adalah keyword yang ditambahkan ke dalam selectors spesifik tergantung dari elemen HTMLnya. Contoh **:hover** akan diaplikasikan formatnya ketika user *hover* atau mouse over elemen HTML.

```
selector:pseudo-class {
  property: value;
}
```

List of pseudo-classes

- ❖ :-ms-input-placeholder
- ❖ :active
- ❖ :checked
- ❖ :disabled
- ❖ :empty
- ❖ :enabled
- ❖ :first-child
- ❖ :first-of-type
- ❖ :focus
- ❖ :hover
- ❖ :indeterminate
- ❖ :invalid
- ❖ :lang
- ❖ :lang(c)
- ❖ :last-child
- ❖ :last-of-type
- ❖ :link
- ❖ :not
- ❖ :nth-child(n)
- ❖ :nth-last-child(n)
- ❖ :nth-last-of-type(n)
- ❖ :nth-of-type(n)
- ❖ :only-child
- ❖ :only-of-type
- ❖ :optional
- ❖ :required
- ❖ :root
- ❖ :target
- ❖ :valid
- ❖ :visited
- ❖ Dynamic pseudo-classes
- ❖ Structural pseudo-classes
- ❖ ui element states pseudo-classes

### ***Link and user action pseudo-classes***

#### **Link Pseudo Class**

- :link —normal atau default state dari links
- :visited — select links yang telah dikunjungi pada browser

#### **User Action Pseudo-Classes**

- :focus — selects links yang sedang fokus (cursor keyboard)
- :hover — selects links yang sedang di cover oleh pointer mouse
- :active — selects links yang sedang diklik.

Contoh

```
a:link {
  color: blue;
}

a:visited {
  color: gray;
}

a:hover, a:focus {
  text-decoration: none;
}

a:active {
  font-weight: bold;
}
```

Untuk input form ketika cursor mengarah ke input tersebut:

```
input:focus {
  border: 2px solid black;
  background color: lightgray;
}
```

### *The negation (not) pseudo-class*

Contoh:

```
<section id="abstract"> ... </section>
<section id="experiment"> ... </section>
<section id="tests"> ... </section>
<section id="results"> ... </section>
<section id="conclusion"> ... </section>
<section id="references"> ... </section>
```

Kita dapat menyelect semua elemen section kecuali yang memiliki ID “paragraph4” kita bisa menggunakan:

```
#abstract, #experiment, #tests, #results, #conclusion {
  ...
}
```

Namun dengan negation selector, sbb:

```
section:not(#references) {
  ...
}
```

Negation selector is not supported by IE8 and below.

### ***The language (lang) pseudo-class***

```
<p lang="en-US">A paragraph of American text, gee whiz!</p>
```

Untuk menunjuk elemen P yang languagenya “en-US” caranya

```
p:lang(en-US) {  
    ...  
}
```

### ***The target pseudo-class***

Menunjuk sebuah elemen jika elemen itu menjadi target dari current page URL, contoh:

```
<a href="#target">Click me</a>
```

```
<div id="target">Woot!</div>
```

Untuk style elemen DIV hanya ketika link diklik bisa menggunakan:

```
div:target {  
    ...  
}
```

### ***UI element state pseudo-classes***

Pada HTML 5 Form, elemen INPUT memiliki atribut required, yang mana akan menampilkan warning ketika elemen tersebut tidak dibiarkan blank ketika disubmit.

```
<input type="text" required>
```

Kita dapat style elemen INPUT hanya ketika data yang dimasukkan kedalamnya valid atau tidak valid

```
input:valid {}
```

atau

```
input:invalid {}
```

Selain itu kita juga bisa style elemen berdasarkan atribut enabled (default) atau disabled

```
input:enabled {}
```

and this style:

```
input:disabled {}
```



Akhirnya, kita pun juga bisa style checkbox hanya ketika di checked

```
input[type="checkbox"]:checked { }
```

### ***Structural pseudo-classes***

Structural pseudo-classes adalah advanced selectors yang mengizinkan kita untuk menunjuk ke elemen HTML yang spesifik berdasarkan posisinya dalam dokumen hirarki. Selector ini diperkenalkan pada CSS level 3, lihat kembali adjacent sibling selector.

❖ **:root**

Menunjuk ke elemen root atau induk dari dokumen, biasanya elemen <html>.

```
html:root{ ... }
```

❖ **<p>:nth-child(n)**

Menunjuk ke anak dari elemen HTML tertentu secara berulang, cocok untuk menunjuk item dari list atau cell dari tabel:

```
<ul>
  <li>First</li>
  <li>Second</li>
  <li>Third</li>
  <li>Fourth</li>
  <li>Fifth</li>
  <li>Sixth</li>
  <li>Seventh</li>
  <li>Eighth</li>
  <li>Ninth</li>
  <li>Tenth</li>
</ul>
```

n di set sesuai pattern yang kita inginkan.

- Untuk menunjuk ke semua item dalam list gunakan:

```
li:nth-child(n)
```

- Untuk menunjuk ke item ganjil atau genap dalam list gunakan:

```
li:nth-child(odd)
li:nth-child(even)
```

- Or you can use this rule to accomplish the same result:

```
li:nth-child(2n+1)
li:nth-child(2n+0)
```

Aturannya sebagai berikut:

- o `li:nth-child(5)`: select the fifth adjacent list item.
- o `li:nth-child(4n+1)`: select every fourth list item, and then add 1 to each result. So numbers 5 and 9.
- o `li:nth-child(3n-2)`: select every third list item, and subtract 2 from each result. So numbers 1, 4 and 7.

Kemudian kita juga bisa menggunakan `:first-child` dan `:last-child` selectors —menunjuk ke elemen pertama atau terakhir dari elemen tertentu:

```
li:first-child { ... }

li:last-child { ... }
```

## Pseudo-elements

Pseudo elemen menunjuk sebagian dari elemen

### *:first-letter*

Select huruf pertama:

```
p:first-letter {
  font-weight: bold;
  font-size: 300%;
  background-color: red;
}
```

Membuat efek drop cap.

### *:first-line*

Select baris pertama

```
p:first-line {
  font-weight: bold;
}
```

## *Generated content using :before and :after*

Menambahkan konten tertentu kepada elemen tertentu

```
<p class='pertama'>Paragraf1</p>
<p id='kedua'>Paragraf1</p>
```

Tambahkan style CSS:

```
p.pertama:before{ content: " mulai "; font-weight:bold; }  
p#kedua:after{ content: " akhir "; font-weight:bold; }
```

### ***CSS3 pseudo-element double colon syntax***

CSS 3 memiliki cara baru untuk menulis pseudo elements yaitu dengan menggunakan double titik dua. Namun saat ini masihizinkan kita menggunakan single titik dua. Untuk kedepan gunakan ini

```
a::after { ... }
```

Tapi sekarang gunakan single titik dua saja karena terkait dukungan browser yang belum sempurna

## **Inheritance dan cascade**

### **Inheritance**

Inheritance = Aturan CSS untuk suatu elemen HTML akan mempengaruhi semua child elemen. Namun tidak semua properti CSS inheritance contoh margin dan width tidak inheritance. Properti background juga tidak inheritance namun default nilainya adalah transparent, artinya jika parent diset backgroundnya maka child juga akan terpengaruh.

Misal untuk kode HTML berikut

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="UTF-8">  
    <title>Inheritance</title>  
  </head>  
  <body>  
    <h1>Heading</h1>  
    <p>A paragraph of text.</p>  
  </body>  
</html>
```

Kita kenai aturan CSS sbb:

```
html {  
  font: 75% Verdana, sans-serif;  
}
```

Dengan kode diatas maka semua elemen dibawah elemen HTML akan dikenai CSS tersebut (font default most browser = 16 pixel, 75% x 16 pixel = 12 pixel), Yap konten dalam elemen <h1> dan <p> akan memiliki font dengan size 12px dan font Verdana atau jika tidak maka sans-serif.

Misal ada aturan CSS tambahan:

```
h1 {  
    font-size: 300%;  
}
```

Maka ukuran dari font dalam elemen h1 adalah 300% of 12px = 36px. Kenapa jadi hitung-hitungan? ☺

Jika kita memahami inheritance ini maka kode CSS kita akan semakin efisien.

### ***Forcing inheritance***

Adakalanya browser bandel.. ☺ maksudnya adalah setiap browser memiliki nilai default untuk elemen tertentu, contohnya elemen anchor atau <a>. Tanpa kita atur melalui CSS maka dengan sendirinya elemen <a> memiliki warna atau format sendiri, karena itu pemaksaan terhadap inheritance perlu dilakukan-

Contoh

```
<ul>  
    <li><a href="/">Home</a></li>  
    <li><a href="/news/">News</a></li>  
    <li><a href="/products/">Products</a></li>  
    <li><a href="/services/">Services</a></li>  
    <li><a href="/about/">About Us</a></li>  
</ul>
```

Kita ingin agar warna dari teks dibawah elemen <ul> berwarna hijau

```
ul {  
    background: green;  
}
```

Ternyata nggak cukup, perlu ditambah:

```
ul li a {  
    color: inherit;  
}
```

## Cascade

Karena CSS mengenal inheritance pada beberapa propertinya, maka seringkali akan terjadi konflik pengaturan (conflict of interest kah? ☺), yaitu satu elemen HTML di format oleh beberapa deklarasi CSS. Lantas mana yang akan dieksekusi atau diterapkan?

Ada 3 konsep utama dalam cascade:

1. Importance
2. Specificity
3. Source order

Keterangan:

Importance is the most ... er ... important. Jika dua deklarasi sama-sama diset importance, maka specificity menentukan. Jika specificity sama, maka source order yang menentukan

### *Importance*

Urutan selector dari yang paling tinggi atau paling diperhitungkan: Pseudo code, ID, Class, dan Elemen. Misalnya kita memiliki elemen `<h1 id="heading1" class="heading">My Text</h1>`, dengan aturan CSS:

```
h1 {  
    color:red;  
}  
#heading1{  
    color:yellow;  
}  
.heading{  
    color:green;  
}
```

Apa warna "My Text" ??, lalu coba edit .heading menjadi:

```
.heading{  
    color:green !important;  
}
```

### *Specificity*

Specificity maksudnya seberapa spesifik selector itu menunjuk ke elemen tertentu. Sebagai contoh selector `* {}` yaitu menunjuk ke semua elemen pada dokumen HTML merupakan selector dengan low specificity. Sedangkan selector `#nav` yang hanya menunjuk ke elemen

dengan id nav memiliki specifity yang tinggi. Artinya ketika terdapat dua deklarasi CSS maka mana yang didahulukan? Jawabannya adalah tergantung nilai dari specifity. Nilai dari specifity dapat dihitung sehingga kita nanti bisa tahu mana aturan CSS yang diterapkan ketika 2 atau lebih format CSS “bertabrakan”.

Specificity mempunyai 4 komponen; sebut saja dengan komponen a, b, c dan d. Komponen “a” yang paling tinggi sedangkan “d” yang terendah

- ✓ Komponen “a” deklarasi CSS yang diletakkan di atribut style (inline styling), selain itu nilainya 0.

```
<p style="background:blue; color:white; padding:5px;">Paragraph</p>
```

- ✓ Komponen “b” adalah jumlah selector id yang merujuk ke elemen (diawali dengan tanda #)
- ✓ Komponen “c” adalah jumlah attribute selectors—termasuk class selectors — dan pseudo-classes.
- ✓ Komponen “d” adalah jumlah elemen dan pseudo-elements selector.

Contoh :

Selector	a	b	c	d	Specificity
<b>h1</b>				1	0,0,0,1
<b>.foo</b>			1		0,0,0,1
<b>#bar</b>		1			0,1,0,0
<b>html&gt;head+body ul#nav *.home a:link</b>		1	2	5	0,1,2,5

#### Penjelasan no 4.

Ada satu selector ID (#nav), sehingga b = 1. Ada satu attribute selector (.home) dan satu pseudo-class (:link), sehingga c = 2. Ada 5 tipe elemen selector (<html>, <head>, <body>, <ul> and <a>), sehingga d = 5. Sehingga hasilnya 0,1,2,5.

Note: Combinators (like >, + and white space) do not affect a selector’s specificity. The universal selector (\*) has no input on specificity, either.

Note #2: There is a huge difference in specificity between an id selector and an attribute selector that happens to refer to an id attribute. Although they match the same element, they have very different specificities. The specificity of #nav is 0,1,0,0 while the specificity of [id="nav"] is only 0,0,1,0.

Mari kita lihat contoh berikut:.

1. Buat dokumen HTML lalu masukkan beberapa paragraph.

```
<body>
  <h1>Heading</h1>
  <p>A paragraph of text.</p>
  <!-- Add this -->
  <p>A second paragraph of text.</p>
</body>
```

2. Tambahkan kode CSS untuk memformat teks pada elemen paragraf

```
p {
  color: blue;
}
```

3. Simpan dan lihat via browser. Maka dua paragraph akan berwarna biru.
4. Edit kode HTML, tambahkan **id** pada paragraph pertama.

```
<body>
  <h1>Heading</h1>
  <!-- Add the id of "special" to this paragraph -->
  <p id="special">A paragraph of text.</p>
  <p>A second paragraph of text.</p>
</body>
```

5. Tambahkan CSS sebagai berikut:

```
#special {
  background-color: red;
  color: yellow;
}
```

6. Simpan dan lihat via browser. Maka paragraph pertama akan berwarna kuning dengan latar belakang merah, sedangkan paragraph kedua tetap sama yaitu berwarna biru.

### **Source order**

Jika dua deklarasi CSS menunjuk ke elemen yang sama, dimana memiliki importance & specifity yang sama maka deklarasi CSS yang terakhirlah yang akan diterapkan.

Contoh, tambahkan pada kode CSS diatas.

```
p {
  background-color: yellow;
  color: black;
}
```

Maka paragraph pertama akan berwarna kuning dengan latar belakang merah, sedangkan paragraph kedua akan berwarna hitam dengan latar belakang kuning.

## Bekerja dengan Huruf & Teks

### Dasar Pemformatan Teks

Untuk mengawali materi ini, sangat penting untuk membaca tentang typography di web [http://docs.webplatform.org/wiki/concepts/web\\_typography](http://docs.webplatform.org/wiki/concepts/web_typography) sehingga bisa menjadi gambaran awal bagi kita untuk melakukan styling text di web.

### Format Font

CSS memiliki beberapa properties yang dimulai dengan font-, yang mengizinkan kita mengontrol banyak fitur teks.

### Memilih Jenis Huruf

font-family mengizinkan kita untuk menentukan jenis font yang digunakan. Gunakan kode HTML berikut

<http://people.opera.com/cmills/css-text-styling/zappa.html>, dan tambahkan aturan CSS font berikut.

```
body { font-family: Arial; }
```

Kode ini akan memformat teks dari semua elemen dalam body menjadi teks dengan jenis font Arial. Yang mana huruf ini biasanya terdapat pada semua system operasi.

### Jenis Huruf Yang AMan

Ada 11 font default yang "insyaAllah" ada pada setiap sistem operasi:

- ✓ Sans-serif: fonts without serifs: Verdana, Arial, Trebuchet MS
- ✓ Serif: fonts with serifs: Times new roman, Georgia
- ✓ Monospaced: fonts in which every glyph takes up the same space, like in computer code: Andale mono, Courier new
- ✓ Cursive: fonts that have a decorative, often handwritten-looking style: Comic Sans
- ✓ Fantasy: fonts that have a bold, often ornamental or quirky style, which are meant to be used for headings, not body copy: Impact

### Deklarasi Huruf

Deklarasi lebih dari satu jenis font. Kadangkala kita menginginkan jenis font tertentu yang ada dikomputer kita diluar web safe fonts diatas, namun font tersebut belum tentu ada dikomputer lain sehingga sebagai cadangan atau pengganti kita juga bisa mendeklarasikan di kode CSSnya.

```
font-family: 'helvetica neue', arial, verdana, sans-serif;
```



Jika Helvetica neue tidak ditemukan maka browser akan menggunakan jenis font arial, dst.

### Gambar Pengganti

Kadangkala kita ingin menampilkan font yang unit yang tidak ada dalam dunia nyata.. maka kita bisa membuatnya via photoshop atau image editor lainnya. Contoh:



Simpan image diatas dengan nama zappa.png, lalu buat kode htmlnya

```
<h1>FRANK ZAPPA</h1>
```

Lalu kode CSSnya

```
h1 {  
  background-image: url(zappa.png);  
  height: 55px;  
  text-indent: -9000px;  
}
```

Jadi intinya elemen h1 akan format backgroundnya menjadi image diatas, lalu teks yang ada di dalam elemen h1 “dilempar” sejauh 9000 pixel ke kiri. ☺

Checkout <http://people.opera.com/cmills/css-text-styling/zappa3.html>

### Fitur Web fonts

CSS3 memperkenalkan Web fonts, adalah fitur yang mengizinkan kita menggunakan font secara embedded meskipun di sistem user belum diinstal. Gunakan **@font-face** :

Misalnya kita memiliki file font dengan nama myfont.ttf, taruf dalam satu directory dengan file CSS kita.

```
@font-face {  
  font-family: 'My font';  
  src: url('myfont.ttf') format('truetype');  
}
```

Yap itu kode dasarnya, sebenarnya ada banyak opsi namun saat ini keep simple dulu ☺

Setelah dideklarasikan kita bisa menggunakan font diatas:

```
p{  
  font-family: 'My font';  
}
```

Mudah bukan?

Kode diatas memang simple namun tidak semua browser support dengan format font tersebut. Sehingga untuk membuat kode kita cross browser maka kodenya nggak simple lagi ☹.

Ikuti step berikut

1. Font yang digunakan disini atau pada Frank Zappa heading adalah Romantiques, yang bisa kita peroleh di <http://www.fontspace.com/category/circus>. Download dan unzip.
2. Kunjungi [fontquirrel.com](http://fontquirrel.com) lalu pilih [@font-face generator](#).
3. Klik "add fonts", lalu pilih filr font Romantique yang kita inginkan lalu check disclaimer agreement.
4. Klik tombol "Download your kit". Tunggu beberapa saat, lalu pada dialog simpan simpan contoh file tersebut.
5. File-file hasil unzip berupa font dengan berbagai format serta file **stylesheet.css**.
6. Kopi semua file font (**.eot**, **.svg**, **.ttf** and **.woff**) ke lokasi yang sama dengan file CSS.
7. Buka file **stylesheet.css** dan kopi kode CSS pada file CSS kitasperti ini:

```
@font-face {  
  font-family: 'RomantiquesRegular';  
  src: url('romantiques-webfont.eot');  
  src: url('romantiques-webfont.eot?#iefix') format('embedded-opentype'),  
        url('romantiques-webfont.woff') format('woff'),  
        url('romantiques-webfont.ttf') format('truetype'),  
        url('romantiques-webfont.svg#RomantiquesRegular')  
  format('svg');  
  font-weight: normal;  
  font-style: normal;  
}
```

#### Keterangan

- ✓ . Nilai src mengacu pada lokasi file font. IE uses the .eot version; most modern browsers will use the .woff, which is smaller in file size than the others; older browsers other than IE that do not support .woff will use the .ttf or .svg files.
  - ✓ the font-weight and font-style specify attributes such as the weight of the font, and if it is italic, but it is not necessary to worry about those for now
8. Edit CSS untuk h1, ganti dengan kode berikut:

```
h1, h2 {  
  font-family: RomantiquesRegular;  
}
```

Simpan dan lihat via browser. Cara ini lebih flexible dibandingkan dengan image replacement.

<http://people.opera.com/cmills/css-text-styling/zappa4.html>

## Pengaturan Satuan dan Ukuran dengan font-size

```
font-size: 62.5%;  
font-size: 12px;  
font-size: 1.6em;  
font-size: xx-small;
```

Ada 4 unit size dalam CSS:

- ✓ Pixel: ukuran titik pada layar. Contoh: font-size:10px; (Lack in <=IE6)
- ✓ Keyword: xx-small, x-small, small, medium, large, x-large, and xx-large. Contoh ukuran font untuk body = medium, top level heading = xx-large, second level heading = x-large, dst
- ✓ Percentage: misal font-size: 62.5%. Default browser = 16px, 62.5% = 10px.
- ✓ Em: hampir sama dengan persen, misal 100%=1em, 150%=1.5 em

## Perubahan Deatil Huruf dengan font-weight, font-style, and font-variant

- ✓ **font-weight** allows you to set the boldness of text in selected elements.
- ✓ **font-style** allows you to set a element's text to be oblique or italic.
- ✓ **font-variant**allows you to set and element's text to be small-caps, also known as copperplate letters.

### Menggunakan font-weight

Untuk membuat font lebih tebal gunakan font-weight, dengan nilai sbb:

- **bold** dan **bolder** provide two levels of extra boldness.
- **lighter** provides a level of less boldness.
- **100, 200 ...** all the way up to **900** provide incremental levels of boldness.
- **normal** is the default non-bold setting.

Kenyataannya tidak semua nilai setting tampak perubahannya di web terutama jika font sizenya kecil banget.

Sebagai contoh aturan CSS berikut ini meng-highlight baris pertama paragraph pada masing-masing heading

```
h1 + p:first-line, h2 + p:first-line {  
  font-weight: bold;  
}
```

Note: **bold** is equivalent to **700**.

### Format font-style

font-style dapat bernilai italic, oblique, and normal. Normal adalah default, sedangkan italic memerintahkan browser untuk menggunakan italic version dari font (jika ada),

and oblique memerintahkan browser untuk menggunakan font normal, tapi but slant it. Jika font tidak support dengan italic maka ketika kita set italic maka font tersebut akan diset ke oblique..

Berikut ini kodenya

```
font-style: italic;
```

### ***Bekerja dengan font-variant***

font-variant memiliki 2 nilai yaitu, normal yang merupakan default, dan small-caps, yang menggunakan karakter capital namun ukurannya lebih kecil dari huruf pertama:

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

### **Penyingkatan Kode Huruf**

Urutannya

- ✓ Bold, italic, and small-caps (small capital letters)
- ✓ The size
- ✓ The line height
- ✓ The font typeface

```
p {font: italic 75%/125% "Comic Sans MS", cursive;}
```

Sama dengan :

```
font-style:italic;  
font-size:75%;  
line-height:125%;  
font-family: "Comic Sans MS", cursive;
```

### **Format Teks**

#### ***Text direction – direction : rtl / ltr***

Mendefinisikan direction dari text, misal untuk menulis text arab

```
p{ direction:rtl; }
```

#### ***text-align : left / right / center / justify***

Mendefinisikan perataan atau alignment horizontal dari text:

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

#### ***vertical-align: baseline/sub/super/top/text-top/middle/bottom/text-bottom/ length/%***

Mendefinisikan perataan atau alignment vertical dari text:

```
td{ vertical-align:middle; }
```

***text-indent: length (px,em) / percentage (%)***

Mendefinisikan indentasi atau tabulasi

```
p{ text-indent:10em; }
```

***text-transform: none/capitalize/uppercase/lowercase***

Mengontrol transformasi huruf

```
p{ text-transform:capitalize; }
```

***text-decoration: none/underline/overline/line-through/blink***

Memberikan dekorasi pada suatu text:

```
p{ text-decoration:blink; }
```

***Line-height: normal/length/%***

Mendefinisikan tinggi area text:

```
p{ line-height:0.8em; }
```

***Letter-spacing:normal/length***

Mengontrol spasi antar karakter

```
p{ letter-spacing:0.8em; }
```

***Word-spacing: normal/length***

Mengontrol spasi antar kata.

```
p{ word-spacing:0.8em; }
```

***White-space:normal/pre/nowrap***

Mengontrol spasi kosong ditampilkan apa adanya kah?

```
p{ white-space:pre; }
```

***FYI: CSS 3 Text Shadow***

Membuat efek shadow pada teks

Text-shadow: shadow\_right shadow\_bottom shadow\_Intensity;

```
h3{ text-shadow: red 0.2em 0.3em 0.2em }
```

## Grafik

### Menggunakan Gambar Latar Belakang dengan CSS

Menggunakan image sebagai background hanya bisa dilakukan via CSS sebagaimana yang dijelaskan pada materi HTML sebelumnya.

CSS Background sendiri memiliki beberapa property

**Background-color: color value|transparent| inherit**

Mengatur warna background dari suatu elemen HTML.

```
background-color:red;
```

**Background-image: URL | none | inherit**

Mengatur image background dari suatu elemen HTML.

```
background-image:url(alert.png);
```

**Background-repeat: repeat | repeat-x | repeat-y | no-repeat**

Mengatur perulangan image background dari suatu elemen HTML.

```
background-image:url(alert.png);
```

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

**Background-attachment: scroll | fixed | inherit**

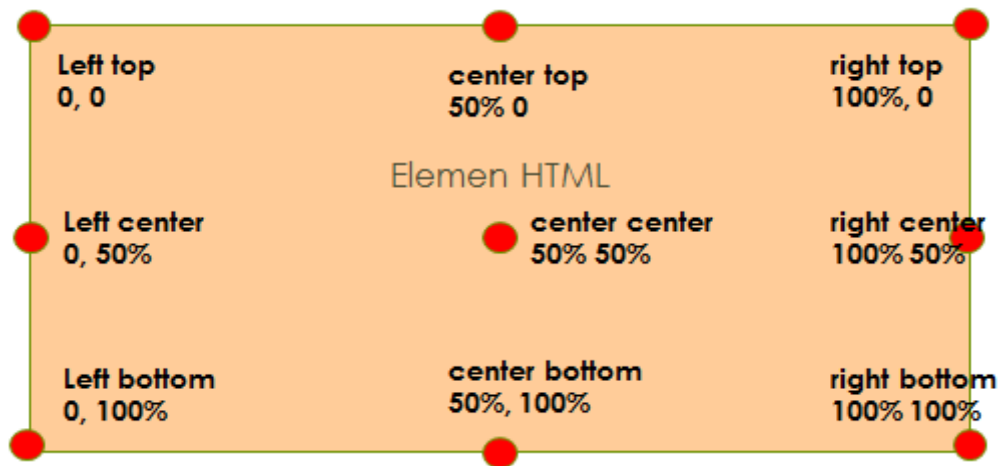
Mengatur apakah sebuah image background bisa discroll atau tidak.

```
background-attachment:scroll;
```

**Background-position: length (umumnya pixels) | percentage (dari lebar elemen) | the keywords (top, right, bottom, left, dan center)**

Mengatur posisi image background secara horizontal dan vertical

```
background-position:right bottom;
```



### Penyingkatan Pada Kode Background

1. color
2. image
3. repeat
4. attachment (sangat jarang digunakan)
5. horizontal-position
6. vertical-position

background:green url(logo.gif) no-repeat scroll left top;

Selegkapnya [http://docs.webplatform.org/wiki/tutorials/using\\_css\\_background\\_images](http://docs.webplatform.org/wiki/tutorials/using_css_background_images)

### Membuat Efek Gradien pada CSS

Preview



```
background: #1e5799; /* Old browsers */
```

```
background: -moz-linear-gradient(top, #1e5799 0%, #2989d8 50%,  
#207cca 51%, #7db9e8 100%); /* FF3.6+ */
```

```
background: -webkit-gradient(linear, left top, left bottom,  
color-stop(0%,#1e5799), color-stop(50%,#2989d8), color-  
stop(51%,#207cca), color-stop(100%,#7db9e8)); /* Chrome,Safari4+  
*/
```

```
background: -webkit-linear-gradient(top, #1e5799 0%,#2989d8
50%,#207cca 51%,#7db9e8 100%); /* Chrome10+,Safari5.1+ */

background: -o-linear-gradient(top, #1e5799 0%,#2989d8
50%,#207cca 51%,#7db9e8 100%); /* Opera 11.10+ */

background: -ms-linear-gradient(top, #1e5799 0%,#2989d8
50%,#207cca 51%,#7db9e8 100%); /* IE10+ */

background: linear-gradient(to bottom, #1e5799 0%,#2989d8
50%,#207cca 51%,#7db9e8 100%); /* W3C */

filter: progid:DXImageTransform.Microsoft.gradient(
startColorstr='#1e5799', endColorstr='#7db9e8',GradientType=0 );
/* IE6-8 */
```

checkout <http://www.colorzilla.com/gradient-editor/>

## Membuat Border menggunakan CSS border-radius

box radius untuk membuat efek rounded elemen

This box should have a rounded corners for Firefox, Safari/Chrome, Opera and IE9.

border-radius: top-left top-right bottom-right bottom-left;

Contoh:

```
div{
    border-radius:5px 10px 15px 20px;
}
```

Gunakan -moz-border-radius untuk old Mozilla Browser (<3.5)

<http://border-radius.com/>

<http://www.css3.info/preview/rounded-border/>



## Menambahkan Efek Bayangan pada CSS box-shadow

box shadow (untuk membuat efek bayangan dan)

Box-shadow: color width\_shadow\_right width\_shadow\_bottom width\_blur

```
<div>Tes Border Radius</div>
```

```
div{
    height:100px;
    width:100px;
    border:1px solid black;
    box-shadow: black 0.5em 0.5em 0.3em;
}
```

## Memahami efek CSS filter

[http://docs.webplatform.org/wiki/tutorials/css\\_filters](http://docs.webplatform.org/wiki/tutorials/css_filters)

## Menambahkan Efek Transparan

Ada dua cara untuk membuat efek transparent di CSS

**Properti Opacity, nilainya 0.0 – 1.0**

```
<div>TEXT TRANSPARENT</div>
```

```
div{
    background-color:blue;
    opacity:0.5;
    filter:alpha(opacity=40); /* For IE8 and earlier */
}
```

## Alpha Color

Menggunakan background warna RGBA (HSLA). (Lihat di CSS Color cheatsheet)

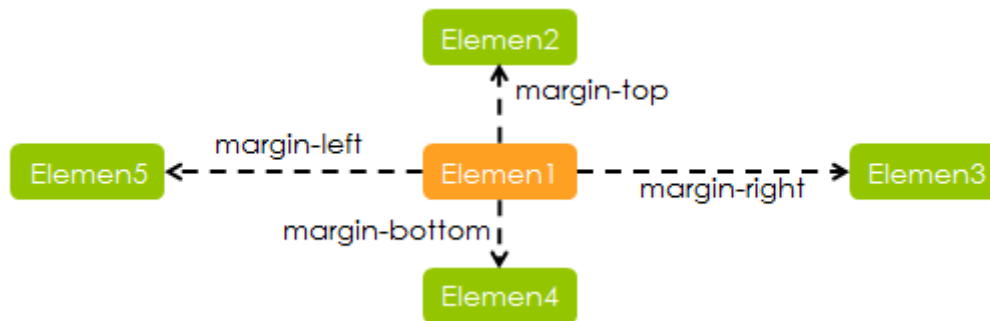
```
<div>TEXT TRANSPARENT</div>
```

```
div{
    background-color:rgba(0,0,255,0.5);
}
```

}

## Layout Web menggunakan CSS

### Margin



Nilai propertinya: px, em (16px), %

Contoh :

```
div{  
margin-top:2em;  
margin-right:3em;  
margin-bottom:4em;  
margin-left:5em;  
}  
div{  
margin:2em 3em 4em 5em;  
}
```

Shorthand tips:

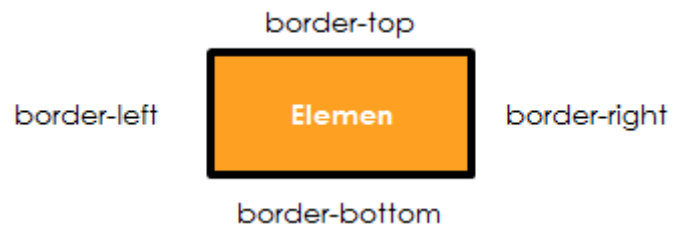
1 = top, right, bottom, left

2 = top & bottom, right & left

3 = top, right & left, bottom

- Nilai negatif => overlap antar elemen.
- Nilai auto u/ block level elemen akan membuat elemen tersebut align center.
- Kecuali elemen IMAGE, margin tidak bisa diaplikasikan untuk inline elemen
- Margin collapse antar 2 elemen

## Border

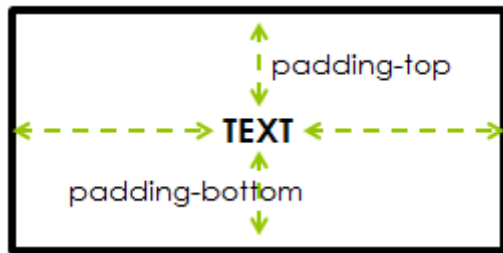


propertinya: border-width, border-style, border-color

Contoh :

```
border-width:5px; /* width all side 5px */
border-top-width:5px; /* width top 5px */
border-width:5px 10px;
/* width top&down=5px, left&right=10px */
border:5px solid red;
/* width style color */
```

## Padding



propertinya: padding, padding-top, padding-right, padding-bottom, padding-left, contoh

```
padding-top:5em;
```

```
padding:5em 10em;
```

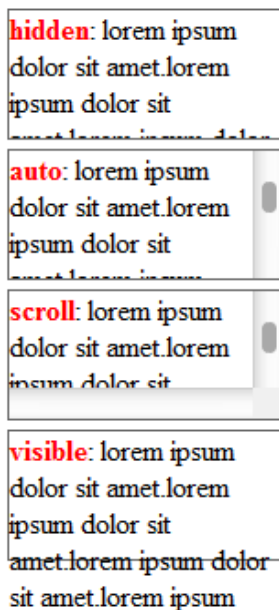
Contoh :

Nilai auto = tidak berguna. Nilai negative = invalid. Padding tidak collapsed.

Padding bisa diaplikasikan untuk inline maupun block level elemen

## Width and Height

### Dasar-dasar Width & Height



Kecuali IMAGE, width dan height tidak bisa diaplikasikan pada inline elemen

```
div{  
width:10em; /* 160px */  
height:5em;  
}
```

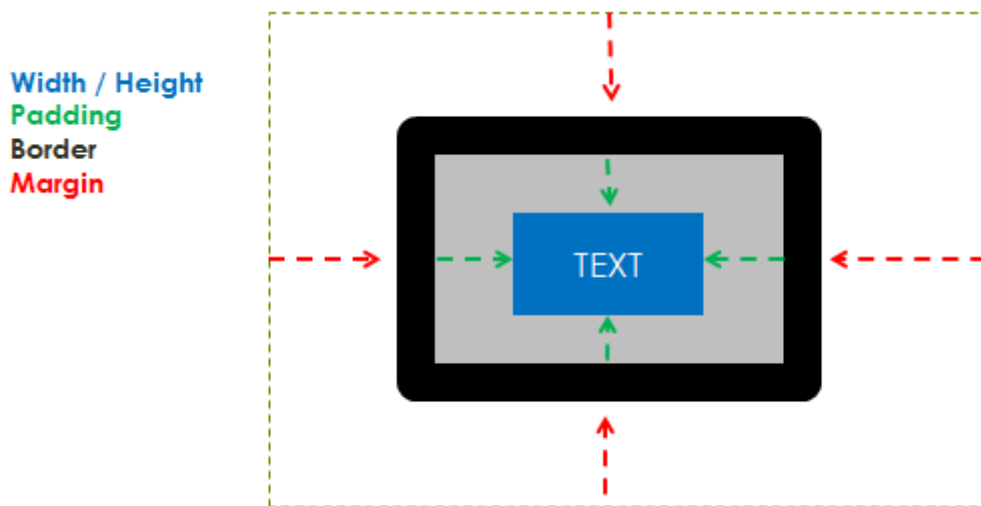
## Overlapping Content

Gunakan properti : overflow dengan 4 valid nilai — visible (default), hidden, auto, and scroll

```
div{  
width:10em; /* 160px */  
height:5em;  
overflow:scroll; /*  
}
```

**Min-height, max-height, min-width, max-width** (lack in <= IE6)

## Actual size



Actual size = width/height + padding + border-size + margin

Jika width/height tidak di set maka properti child akan berpengaruh juga

## Elemen Display

CSS memiliki properti display yang berfungsi untuk menampilkan suatu elemen dengan aturan tertentu. Display memiliki 3 valid nilai yaitu: inline, block, none

`display:inline;`

Menampilkan elemen menjadi inline level elemen, meskipun default elemen tersebut adalah block level maka kode ini akan memaksanya menjadi inline □

`display:block;`

Kebalikan dari inline, menampilkan elemen menjadi block level elemen:

`display:none;`

Perintah ini akan menyembunyikan tampilan elemen dan menghilangkan block-nya

CSS juga memiliki properti visibility yang fungsinya menampilkan atau menyembunikan suatu elemen. Nilainya: visible, hidden. contoh

```
visibility:hidden;
```

Perintah ini akan menyembunyikan tampilan elemen namun tidak menghilangkan bloknya

### **Rata Tengah pada Elemen Block Level**

Untuk mengatur agar block level elemen berada pada posisi tengah secara horizontal maka gunakan properti margin dengan nilai auto;

```
div{  
width:200px;  
margin:0 auto;  
}
```

### **Float & Clear**

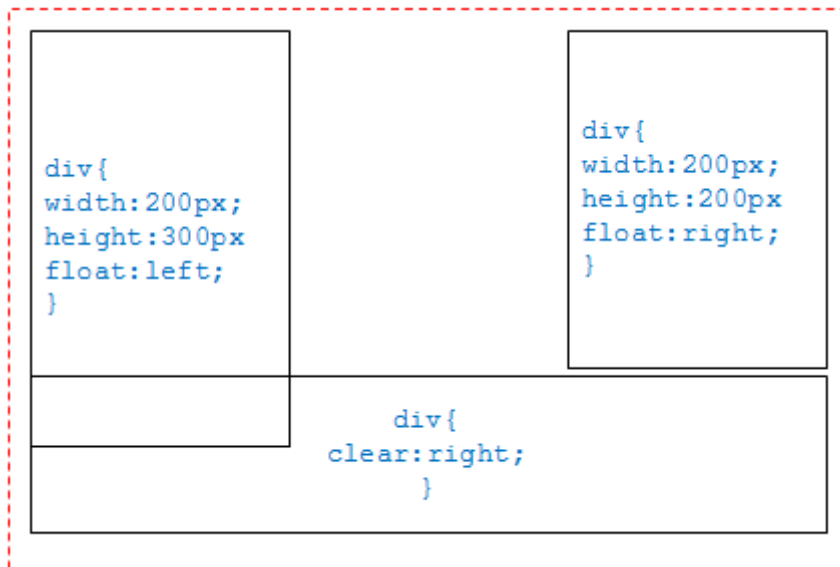
CSS memiliki properti float berfungsi untuk memaksa posisi elemen ke align tertentu. Nilai propertinya: left, right, none.

```
div{  
width:200px;  
float:left;  
}
```

```
div{  
width:200px;  
float:right;  
}
```

Untuk menormalkan float, CSS juga memiliki fungsi clear dengan nilai: left, right, none dan both.

Spesifiknya, float juga bisa digunakan untuk mengatur posisi image dalam suatu artikel, apakah pojok kiri atau kanan (meski ada vertical-align)



Gunakan perintah clear:both.

## Positioning

### Static Position

Posisi default elemen HTML, sesuai dengan dokumen flow elemen. Disini kita mengenal istilah block level element dan inline level element.

### Relative Position

Sama dengan static hanya saja properti align yaitu top, dan left bisa berpengaruh terhadap posisi awalnya. Jadi yang dimaksud dengan relatif disini adalah relatif dengan posisi awalnya (top dan leftnya).

Contoh:

```
<p>Lorem ipsum dolor sit amet consectetur adipiscing elit.
Curabitur feugiat feugiat purus. Aenean eu metus. Nulla
facilisi. <span>DFP Prakom Terampil</span> Pellentesque quis
justo vel massa suscipit sagittis. Class aptent taciti sociosqu
ad litora torquent per conubia nostra, per incepto</p>
```

```
p{
    width:20em;
}
span{
    position:relative;
    left:1em;
```

```
top:1em;  
background:lime;  
}
```

Lorem ipsum dolor sit amet consectetur adipiscing  
elit. Curabitur feugiat feugiat purus. Aenean eu metus.  
Nulla facilisi. Pellentesque  
quis justo vel massa. **DFP Prakom Terampil** Class aptent  
taciti sociosqu ad litora torquent per conubia nostra,  
per incepto

### **Absolute Position**

Posisi ini tidak mengikuti dokumen flow. Berbeda dengan posisi relative yang hanya bisa mengatur left dan top saja, dengan posisi absolute semua align yaitu top, right, left, bottom bisa diatur. Jadi absolute disini adalah posisinya absolute terhadap browser (atau elemen dengan posisi relative) sesuai dengan posisi yang ditentukan.

Dengan mengeset menjadi absolute, membuat elemen seolah berada pada beda layer, dimana absolute terletak pada layar atas. Mengenai hal ini, terdapat juga properti Z-index bagi position absolute yaitu untuk mengatur posisi "layer" apakah didepan atau dibelakang.

Contoh:

```
<p>Lorem ipsum dolor sit amet consectetur adipiscing elit.  
Curabitur feugiat feugiat purus. Aenean eu metus. Nulla  
facilisi. <span>DFP Prakom Terampil</span>Pellentesque quis  
justo vel massa suscipit sagittis. Class aptent taciti sociosqu  
ad litora torquent per conubia</p>  
  
p{  
    width:20em;  
}  
  
span{  
    position:absolute;  
    left:1em;  
    top:1em;  
    background:lime;  
}
```



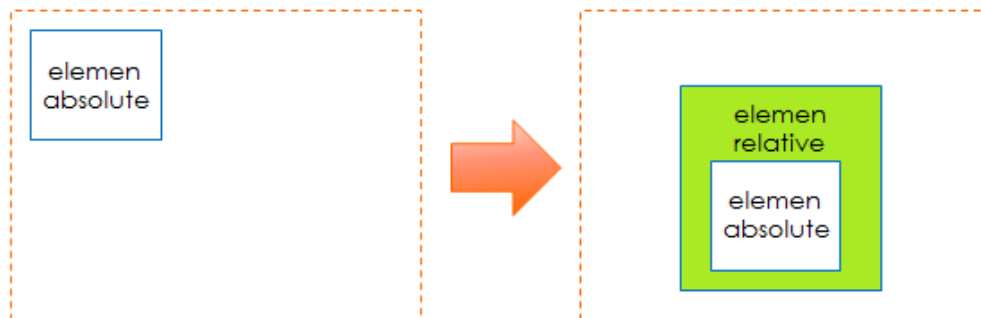
LDFP Prakom Terampil  
met consetetuer adipiscing  
elit. Curabitur feugiat feugiat purus. Aenean eu metus.  
Nulla facilisi. Pellentesque quis justo vel massa  
suscipit sagittis. Class aptent taciti sociosqu ad litora  
torquent per conubia nostra, per incepto

#### Tips:

Posisi absolute dapat digunakan untuk menyembunyikan elemen yaitu dengan mengatur nilai posisi left atau top menjadi nilai negatif

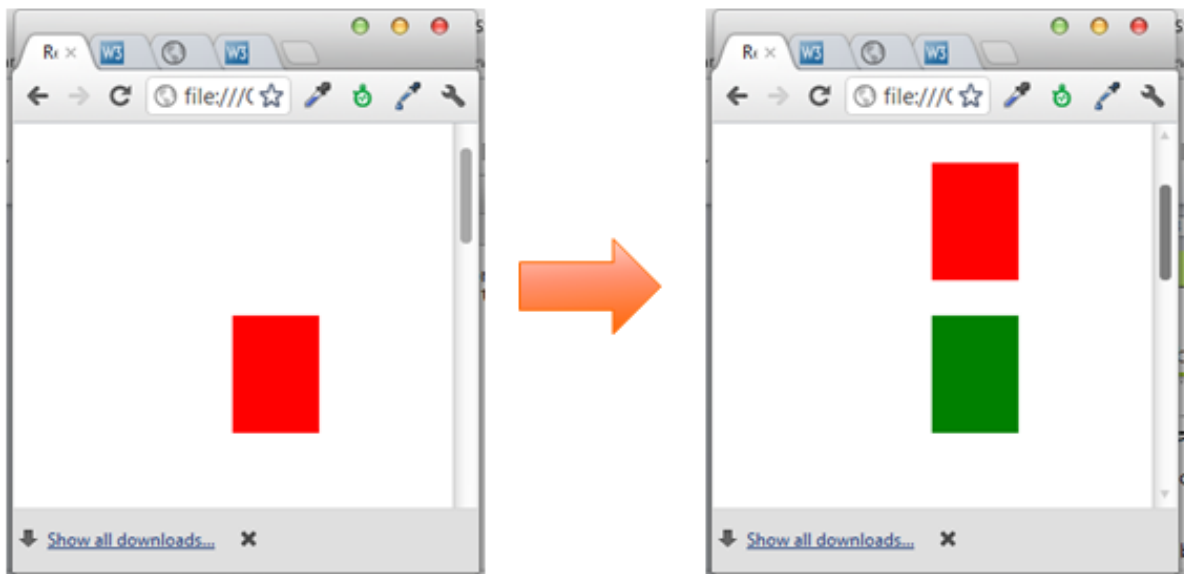
```
div{ position:absolute; top:-1000px;}
```

Gunakan position relative agar elemen dengan posisi absolute, posisinya tidak absolute terhadap browser namun absolute terhadap elemen tertentu



#### Fixed Position

Posisi ini hampir sama dengan absolute, hanya saja posisi fixed tidak statis seperti absolute, namun posisinya senantiasa fixed atau tetap terhadap browser, meskipun browser discroll sekalipun.



```
<style type="text/css">
body{
    height:900px;
}
#fixed{
    width:20%; height:5em;  background:green;
    position:fixed;
    left:50%;
    top:50%;
}
#absolute{
    width:20%; height:5em;  background:red;  position:absolute;
    left:50%;
    top:50%;
}
</style>
</head>
<body>
<div id="fixed"></div>
<div id="absolute"></div>
</body>
```

## CSS untuk perangkat tertentu

### CSS spesifik menggunakan Media Types

CSS untuk spesifik device, umumnya:

1. Screen : Komputer
2. Print : Printer
3. Handled : mobile
4. All : All media (the default)

Contoh deklarasi:

```
<link rel="stylesheet" href="style.css" media="print" type="text/css">
<style type="text/css">
  @import url("style.css") print;
  @media print{
    bla..bla..bla..
  }
</style>
```

### *Membedakan tampilan web dengan tampilan ketika di print*

```
<link rel="stylesheet" href="style.css" media="screen" type="text/css">
<link rel="stylesheet" href="styleforprinting.css" media="print" type="text/css">
```

### *Mencegah printing dokumen:*

```
@media print {
  Body {display: none;}
}
```

### CSS Spesifik menggunakan media queries

Media query merupakan extend dari media type terutama untuk menangani gadget yang mulai beragam. Notebook, Tablet, Smartphone dsb.

Ada 3 cara menentukan media queries:

1. Menggunakan elemen **link**:

```
<link media="screen"
```

```
    href="/path/to/global.css" type="text/css" rel="stylesheet"/>
<link media="only screen and (max-width: 320px)"
    href="/path/to/touch.css" type="text/css" rel="stylesheet"/>
<link media="only screen and (max-width: 1024px)"
    href="/path/to/tablet.css" type="text/css" rel="stylesheet"/>
<link media="handheld"
    href="/path/to/mobile.css" type="text/css" rel="stylesheet"/>
```

## 2. Menggunakan klausa **@import** pada CSS

```
@import url("/path/to/touch.css") only screen and (max-width: 320px);
```

## 3. Menggunakan klausa **@media** pada CSS

```
aside.sidebar { float: right; }
@media only screen and (max-width: 320px) {
    aside.sidebar { display: none }
}
```

## CSS untuk IE ☺

Kadang2 IE perlu perlakuan khusus, oleh karena itu khusus IE mengenali kode pendeteksian IE sbb:

IE tertentu

```
<!--[if IE 6]><![endif]-->
```

Lebih kecil dari IE tertentu

```
<!--[if lt IE 6]><![endif]-->
```

Lebih kecil sama dengan IE tertentu

```
<!--[if lte IE 6]><![endif]-->
```

body {

```
    color: red; /* all browsers, of course */
```

```
    color : green\9; /* IE8 and below */
```

```
    *color : yellow; /* IE7 and below */
```

```
    _color : orange; /* IE6 */
```

```
}
```

## Implementasi CSS pada Dokumen Web

### Format Teks (Web typography)

[http://docs.webplatform.org/wiki/concepts/web\\_typography](http://docs.webplatform.org/wiki/concepts/web_typography)

#### Line length

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas porttitor, augue at rhoncus commodo, nibh nulla feugiat odio, bibendum ornare dolor arcu id nibh. Quisque nibh risus, dignissim nec, ultricies eu, bibendum fringilla, ligula.

Curabitur vehicula. Mauris laoreet. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Maecenas lacus. Donec erat. Nunc eleifend magna a massa.

#### Drop caps

Gunakan first-letter pseudo-element.

**L**orem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas porttitor, augue at rhoncus commodo, nibh nulla feugiat odio, bibendum ornare dolor arcu id nibh. Quisque nibh risus, dignissim nec, ultricies eu, bibendum fringilla, ligula.

#### Small caps

**ABANDON HOPE  
ALL YE WHO ENTER HERE**

## Hanging punctuation

A good typographical effect can be used if your sentence starts with quote marks. Using the `text-indent` CSS property combined with a negative value—either a value in ems (-10em), points (-10pt), pixels (-10px) or percent (-10%)—allows you to shunt the quote mark out into the left, maintaining the left vertical line of your block of text,


**“Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Maecenas porttitor, augue at rhoncus commodo, nibh  
nulla feugiat odio, bibendum ornare dolor arcu id nibh.”**

**“Quisque nibh risus, dignissim nec, ultricies eu, bibendum  
fringilla, ligula.”**

## Format Latar Belakang Gambar

### Membuat Tampilan Pesan

Buatlah tampilan pesan error sebagai berikut

 **Alert!** The text of our alert message goes here.

Copy the following HTML and paste it into the HTML document:

```
<p class="alert">  
  <strong>Alert!</strong> The text of our alert message goes here.  
</p>
```

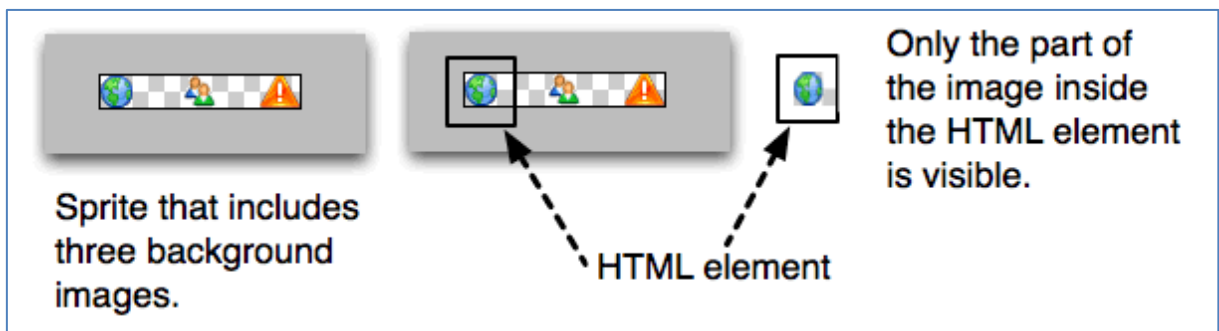
```
.alert{  
  background-color: #FFFFCC;  
  background-image: url(alert.png);  
  background-repeat: no-repeat;  
  background-attachment: scroll;  
  
  /* Add the background position */  
  background-position: 10px 10px;  
}
```

Gunakan image ini ⚠

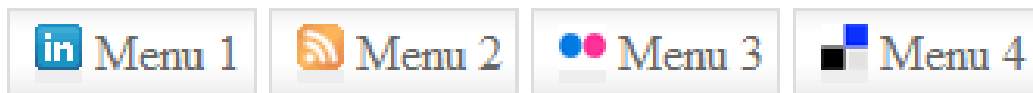
Gunakan shorthand lebih singkat ☺

```
.alert{  
    /* change to the shorthand form */  
    background: #FFFFCC url(alert.png) no-repeat scroll 10px 10px;  
}
```

### Teknik Sprite Image



Buatlah tampilan menu sebagai berikut dengan menggunakan teknik sprite image



### Resource

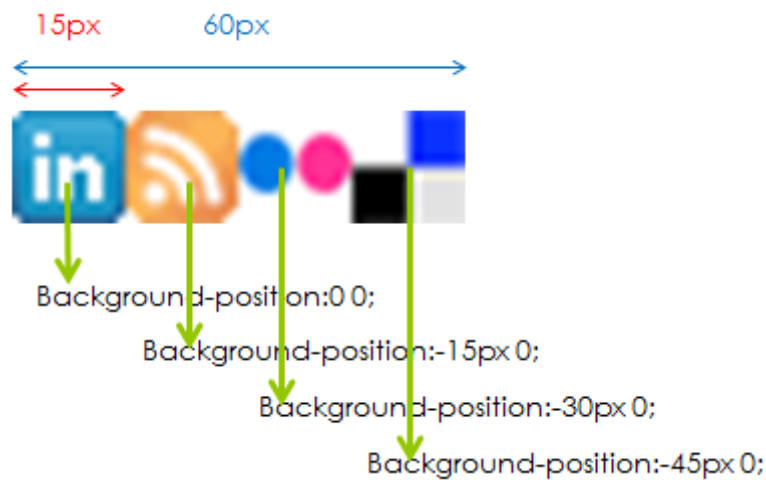


sprite\_logo.gif



sprite\_bg.jpg

### Hitung-hitungan



### **Struktur HTML**

```
<div id="menu">
<a href="#"><span class="logo1">&nbsp;</span>Menu 1</a>
<a href="#"><span class="logo2">&nbsp;</span>Menu 2</a>
<a href="#"><span class="logo3">&nbsp;</span>Menu 3</a>
<a href="#"><span class="logo4">&nbsp;</span>Menu 4</a>
</div>
```

### **Skenario**

- ✓ Image sprite\_logo.gif akan ditampilkan pada elemen SPAN
- ✓ Image sprite\_bg.jpg akan menjadi background dari elemen A (link) dengan repeatation  
background = repeat horizontal = repeat-x

### **Kode CSS**

```
div#menu a{
    border:1px solid #ddd;
    padding:0.3em;
    background:url(sprite_bg.jpg) repeat-x left center;
    text-decoration:none;
    color:#666;
}
div#menu a span.logo1{
    padding-left:0.7em;
```



```
margin:0 0.3em;
background:#eee url(sprite_logo.gif) no-repeat 0 0;
}
div#menu a span.logo2{
padding-left:0.7em;
margin:0 0.3em;
background:#eee url(sprite_logo.gif) no-repeat -15px 0;
}
div#menu a span.logo3{
padding-left:0.7em;
margin:0 0.3em;
background:#eee url(sprite_logo.gif) no-repeat -30px 0;
}
div#menu a span.logo4{
padding-left:0.7em;
margin:0 0.3em;
background:#eee url(sprite_logo.gif) no-repeat -45px 0;
}
```

## **Format lists dan links**

### **List**

#### ***Basic bullets and numbers***

Menggunakan atribut `list-style-type` dengan nilai: none, square, circle, disk, decimal, decimal-leading-zero, lower-roman, upper-roman, upper-alpha, lower-alpha, lower-greek dsb;

CSS Code:

```
ul li {
list-style-type: square;
}
```

HTML Code:

```
<ul>
<li>List item</li>
<li>List item</li>
<li>List item</li>
</ul>
```

- List item
- List item
- List item

## Unordered lists

### Disc

- First item
- Second item
- Third item

### Square

- First item
- Second item
- Third item

### Circle

- First item
- Second item
- Third item

### None - no bullets

First item  
Second item  
Third item

## Ordered lists

### Decimal

1. First item
2. Second item
3. Third item

### Decimal with leading zeros

01. First item
02. Second item
03. Third item

### Lowercase ascii letters


- a. First item
- b. Second item
- c. Third item

### Lowercase roman numerals

- i. First item
- ii. Second item
- iii. Third item

### *Custom bullets using images*

Menggunakan atribut `list-style-image` dengan nilai: none, url image;

Membuat menu file rss berita, gunakan image (rss\_icon.gif) 



HTML Code:

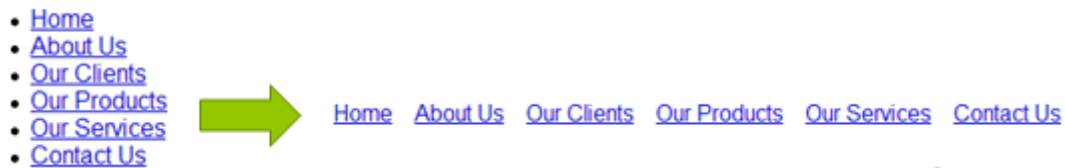
```
<ul class="rss">
  <li><a href="http://example.com/rss.xml">News</a></li>
  <li><a href="http://example.com/rss.xml">Sport</a></li>
  <li><a href="http://example.com/rss.xml">Weather</a></li>
  <li><a href="http://example.com/rss.xml">Business</a></li>
</ul>
```

CSS Code:

```
ul li {
  list-style-image:url(rss_icon.gif) ; /* /
}
```

### ***Horizontal List***

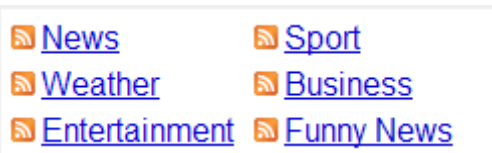
Default list adalah vertical karena ul dan li adalah block level elemen, untuk membuatnya menjadi horizontal gunakan properti display:inline.



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

Cara lain adalah dengan menggunakan float:left dan set width li;

```
li {
  float:left;
  width:150px;
}
```



```
.rss {
  margin: 5px 5px 0 5px;
  padding: 0;
```

```
width: 100%;  
}  
  
.rss li {  
  display: inline-block;  
  width: 40%;  
  margin: 0 2% 0 0;  
  list-style-type: none;  
  background: #fff url("icon-rssfeed.gif") 0 3px no-repeat;  
  padding: 0 0 5px 15px;  
  display: -moz-inline-box; /* old mozilla */  
}  
  
<!--[if lte IE 7]>  
  <style type="text/css">  
    .rss li {  
      float: left;  
    }  
  </style>  
<![endif]-->
```

## Link

### *Memahami link states*

Link atau anchor memiliki 5 macam status:

1. :unvisited — The default state of a link when it has not been activated or visited previously.
2. :visited — The state of a link the user has already visited.
3. :focus — Applies while the link has focus—for example while a keyboard user's cursor is on that link.
4. hover — Applies while a user is "hovering" over the link with a pointer like a mouse, but has not yet clicked the link.
5. active — Applies while the user activates the link—literally during the time they are clicking it. In some browsers this style also applies when the link has been opened in another window or tab.

```
a {}  
a:link {}  
a:visited {}  
a:focus {}  
a:hover {}
```

```
a:active {}
```

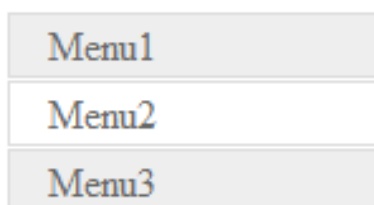
Secara default browser sudah memiliki aturan CSS untuk menangani status link.

```
body {  
    background: #fff;  
    color: #000;  
    font-size: 2em;  
}  
  
a {  
    text-decoration: underline;  
}  
  
a:link {  
    color: #0000CC;  
}  
  
a:visited {  
    color: #6D006D;  
}  
  
a:focus {  
    color: #CC0000;  
}  
  
a:hover {  
    color: #CC0000;  
}  
  
a:active {  
    color: #CC0000;  
    font-style: italic;  
}
```

link, visited, focus, hover, active

## Link & List

### *Interactive Vertical List*



### Kode HTML

```
<ul>
<li><a href="#">Menu1</a></li>
<li><a href="#">Menu2</a></li>
<li><a href="#">Menu3</a></li>
</ul>
```

### Kode CSS

```
ul{
    width:150px;      /* set lebar menu */
}
ul li{
    list-style-type:none;      /* hilangkan bullet stylenya */
    border:1px solid #ddd;
    margin:1px;
    padding:2px 0 2px 15px;
    background-color:#eee;
}
ul li:hover{ /* dieksekusi ketika mouse over ke area LI
    background-color:#fff;
    cursor:pointer;
}
ul li a{
    text-decoration:none;
    color:#666;
}
ul li a:hover{/* dieksekusi ketika mouse over ke area A
    text-decoration:underline;
    color:#333;
}
```

### Format tables

Tabel memiliki 3 bagian utama yaitu

1. Table headings

2. Table data cells
3. Table captions

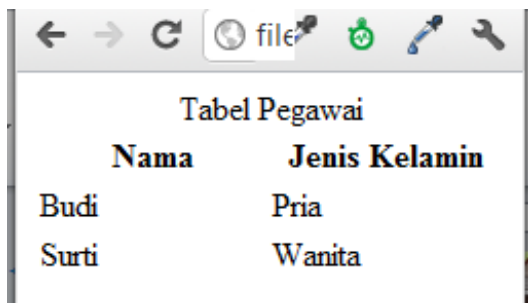
Fokuskan pada style ketiganya.



Tabel Pegawai  
**Nama Jenis Kelamin**  
Budi Pria  
Surti Wanita

#### Table and cell width (default:auto)

```
table {  
    width: 100%;  
}  
th, td {  
    width: 25%;  
}
```



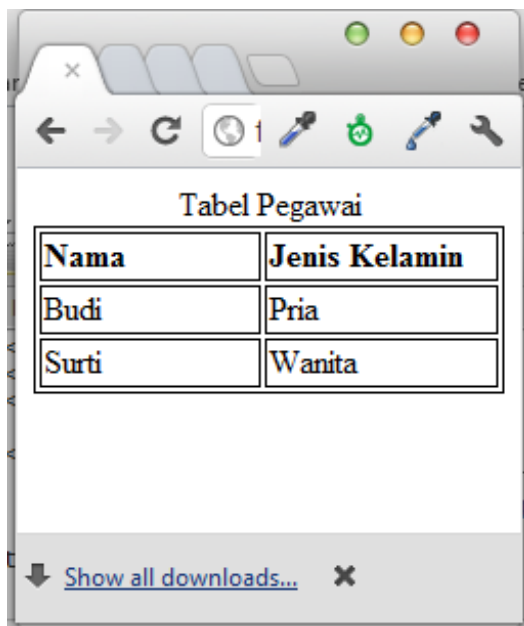
#### Text alignment (default center untuk caption atau th)

```
th, td {  
    width: 25%;  
    text-align:left;  
    vertical-alignment:top;  
}
```

#### Border

```
th, td {  
    ...
```

```
...
border: 1px solid #000;
}
table {
width: 100%;
border: 1px solid #000;
border-spacing: 0;
border-collapse: collapse;
}
```



### Padding

```
th, td {
....
....
padding: 0.3em;
}
caption {
padding: 0.3em;
}
```

### Caption position & styling (old IE tidak support)

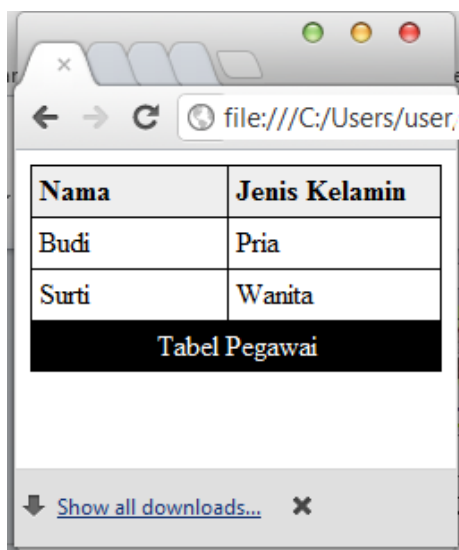
```
table {
```



```
...  
caption-side: bottom;  
color: #fff;  
background: #000;  
}
```

### Style header

```
th {  
    background-color: #eee;
```



The screenshot shows a web browser window with a file:// URL. It displays a table with two columns: 'Nama' and 'Jenis Kelamin'. The table has two data rows: 'Budi' (Pria) and 'Surti' (Wanita). Below the table is a caption 'Tabel Pegawai' on a black background. The browser's download bar at the bottom shows 'Show all downloads...'.

Nama	Jenis Kelamin
Budi	Pria
Surti	Wanita

Tabel Pegawai

### Membuat warna row dalam tabel selang-seling

```
tr:nth-child(even) {  
    background: #CCC;  
}  
tr:nth-child(odd) {  
    background: #FFF;  
}
```

### Format forms

Contoh structure HTML Form

```
<form id="form-login" name="form-login">  
<fieldset>  
<legend>Form Login </legend><ul>
```

```
<li>
<label for="username">Username:</label>
<input type="text" name="username" id="username">
</li>
<li>
<label for="password">Password:</label>
<input type="password" name="password" id="password">
</li>
<li>
<input type="submit" value="Login">
</li>
</ul>
</form>
```

**Contoh Form**

Form Login

- Username:
- Password:
- 

### ***Styling Header & Wrapper***

```
h3 {
  margin: 0 0 1.2em 0;
  border-bottom: .05em solid black;
  font-size: 1.429em;
  line-height: 1.15em;
}
form {
  width: 35.929em;
  margin: 0;
}
ul {
  margin: 0;
  padding: 0;
```

```
list-style-type: none;  
}
```

### ***Styling List and fieldset***

```
li {  
    clear: both;  
    height: 1.714em;  
    margin: 0;  
}  
  
fieldset {  
    height: 1.429em;  
    margin: 0 0 -.143em 0;  
    border: 0;  
}
```

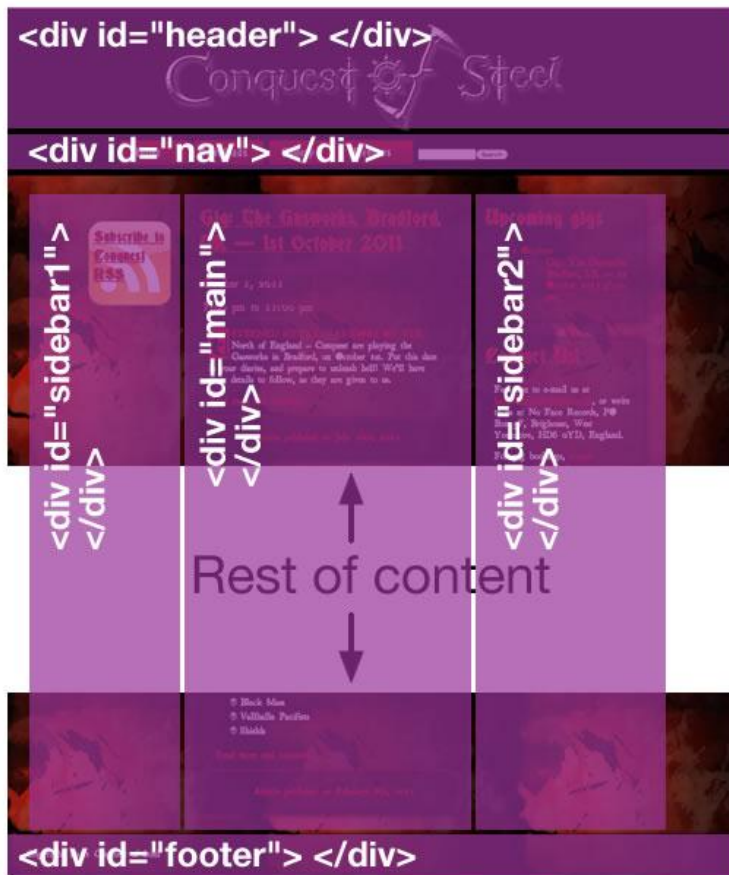
### ***Make 2 coloumn***

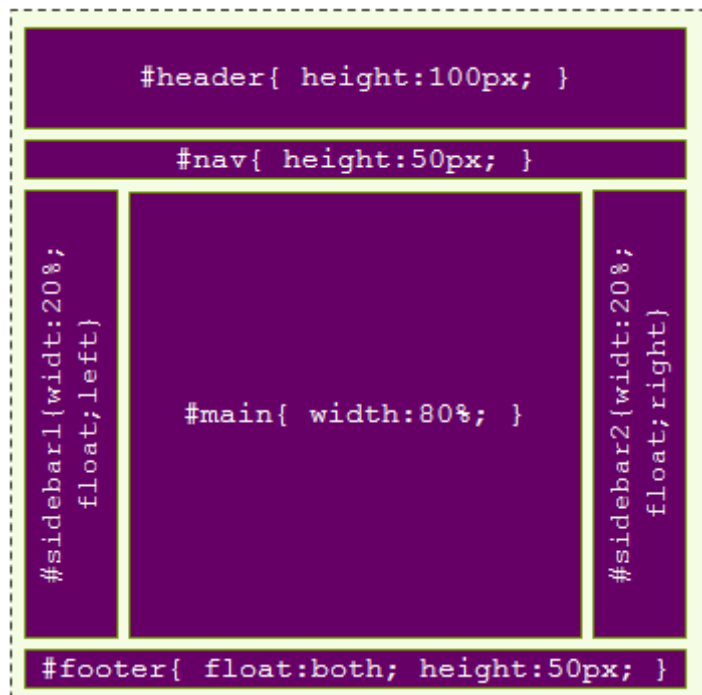
```
label {  
    display: block;  
    float: left;  
    clear: left;  
    width: 5em;  
    overflow: auto;  
    padding-right: 1.714em;  
    text-align: right;  
}  
  
input {  
    height: 1.143em;  
    border: 1px solid #ddd;  
    padding: .071em;  
    line-height: 1em;  
}  
  
dst
```

## Format Layout Halaman

Pada materi Kupas Tuntas HTML, telah kita pelajari tentang konsep layout yang umum digunakan saat ini, yaitu dengan menggunakan elemen DIV.

Kecuali dengan menggunakan elemen TABEL, HTML sendiri tidak bisa membuat layout dokumen sebagaimana disamping, oleh karena itu CSS lah yang akan melakukannya.





## Referensi

<http://w3.org/>

<http://www.webplatform.org>

<http://www.w3schools.com/>

## Biografi Penulis

**Hafid Mukhlisin.** Menyelesaikan S1 jurusan Teknik Informatika di Universitas Teknologi Yogyakarta pada tahun 2008. Mulai tahun 2009 hingga sekarang, penulis berkarir sebagai Pranata Komputer Ahli di Kementerian Keuangan.

Kompetensi inti pada bidang web programming, dan aplikasi Microsoft Office. Berpengalaman dalam pembangunan website atau aplikasi berbasis web pada berbagai instansi pemerintahan dan swasta. Saat ini, aktif melakukan riset dan mengajar di instansi Kementerian Keuangan dalam bidang bidang terkait Teknologi Informasi dan Komputer.