



# CSS

Cascading Style Sheet

PEMROGRAMAN WEB

## MODUL 2021

Mujiyanto, M.KOM

Be a Web Developer Like a Boss

## Daftar Isi

Daftar Isi.....	0
A. Pengenalan Css .....	2
B. Sintaks CSS .....	2
1. Aturan penulisannya : .....	3
C. Bentuk Komentar pada CSS (CSS Comments).....	4
D. Jenis- Jenis Selector pada CSS .....	4
1. Selector berupa Tag HTML.....	4
2. Selector berupa atribut ID .....	5
3. Selector berupa atribut CLASS .....	5
4. Selector untuk Elemen Turunan .....	6
E. Pengelompokan Selector (Grouping) .....	7
F. Mengintegrasikan CSS pada Dokumen HTML.....	7
1. External Style Sheet .....	7
2. Internal Style Sheet.....	8
3. Inline Style Sheet.....	9
G. CSS Box Model.....	9
H. Apa itu CSS Box Model ? .....	9
I. Menetapkan Ukuran Sebuah Elemen ( Width dan Height ) .....	11
J. Margin .....	13
K. Meringkas Penggunaan Property Margin .....	14
L. Border .....	16
M. Padding.....	17

## A. Pengenalan Css

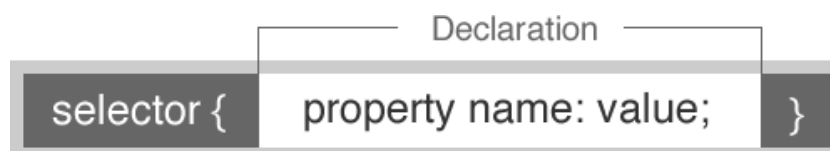
Cascading Style Sheet (CSS) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemrograman. Sama halnya styles dalam aplikasi word processing seperti Microsoft Word yang dapat mengatur beberapa style, misalnya 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>

## B. Sintaks 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.

### 1. 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.

#### (a) Contoh

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

Berikut adalah contoh penulisan baris kode CSS : Atau, agar mudah dibaca bisa ditulis sebagai berikut :

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

#### (b) 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-**

(c) `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*.

### C. 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.

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

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

### D. 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*).

#### 1. Selector berupa Tag HTML

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

(a) Contoh :

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

## 2. 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*).

(a) Contoh :

```
<p>artikel pada paragraf</>
<p>artikel pada paragraf</>
```

File [**index.html**]

**Style.css**

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

#penting {
```

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 di atasnya yang tidak memiliki atribut ID.

## 3. 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 :

```
<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>
```

File [**index.html**]

**Style.css**

```
color: blue;
```

```
<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>
```

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

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;
```

#### 4. 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 :

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

### File [index.html]

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

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*.

### E. 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.

### F. 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*.

#### 1. 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.

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

Sebagai contoh, baris kode CSS dinyatakan sebagai berikut :

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.

## 2. 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>
```

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*.

### 3. 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*.

## G. 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.

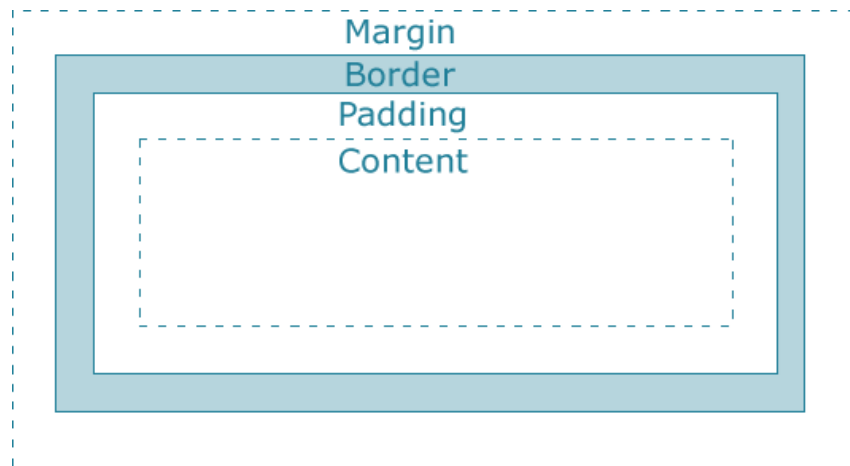
*(i) Note : Contoh-contoh penerapan CSS pada bagian ini menggunakan metode Internal Style*

## H. 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!

### I. 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”**.

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

Sintaks-nya :

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)*.

```
<!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>
```

(a) Contoh :

Hasilnya akan terlihat seperti pada gambar berikut ini



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.

## J. 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 |
```

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.

#### K. 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

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

✓ Property “margin” dengan 4 value

➤ Margin atas = 10px

- Margin kanan = 20px
  - Margin bawah = 30px
  - Margin kiri = 40px
- (dibaca searah jarum jam)*

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

✓ Property "margin" dengan 3 value

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

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

✓ Property "margin" dengan 2 value

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

```
margin: 10px;
```

✓ Property "margin" dengan 1 value

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

Contoh :

```
.container {
    background:
    #06F;
    width:400px;
    height:300px;
    margin:auto;
}

.artikel {
    background: #CCF;
    margin:10px;
}
```

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

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 `<img>` sebesar 20px.
- Jarak antara "artikel" dengan "img" adalah sebesar 30px, dihitung dari margin "artikel" 10px ditambah margin "img" 20px.

#### L. **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.

### M. 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

```
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.