

## MODUL PWPB

# PENGENALAN CASCADING STYLE SHEETS (CSS)

---

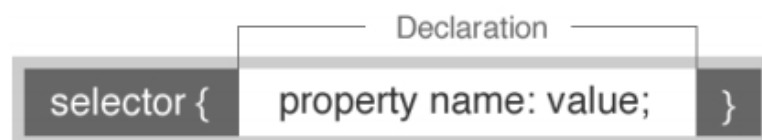
**Cascading Style Sheet (CSS)** merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemrograman.

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.

### A. 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 :

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

## 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 :

1. Selector yang digunakan adalah **“p”**, yakni sebuah tag HTML yang digunakan untuk mendefinisikan sebuah paragraf.
2. Declaration pada contoh diatas dinyatakan dengan **“color:blue;”** dan **“text-align:center;”**
3. 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*.

## B. 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*).

## 1. 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;}
```

## 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*).

**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 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:

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-align: justify;
}
.merah{
    color: red;
}
.biru{
    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;
}
```

#### 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 :

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; }
```

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*.

#### 5. Pengelompokan Selector

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*.

##### 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.

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.

## 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>
</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*.

## 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*.