

MODUL

Aturan Penulisan *Cascading Style Sheet* (CSS)

A. Tujuan

1. Menjelaskan konsep layout dalam tampilan halaman web
2. Menjelaskan berbagai metode layout tampilan halaman
3. Menjelaskan konsep *style* dalam halaman web
4. Menjelaskan penyisipan *style* dalam script program aplikasi web
5. Menjelaskan prosedur penulisan *style script* dalam program aplikasi web
6. Menjelaskan prosedur penyisipan script *style* kedalam html
7. Menerapkan *style* dalam program tampilan halaman web

B. Dasar Teori

Cascading Style Sheet (CSS) adalah suatu bahasa yang bekerja sama dengan dokumen HTML untuk mendefinisikan cara bagaimana suatu isi halaman web ditampilkan atau dipresentasikan. Presentasi ini meliputi *style* atau gaya teks, *link*, maupun tata letak (*layout*) halaman. Dengan adanya teknologi seperti ini, kita dapat memilah atau memisahkan antara kode untuk isi halaman *web* dan kode yang diperlukan khusus untuk menangani tampilan.

Saat ini masih sering dijumpai para pengembang web yang menggunakan *tag* `<table>` untuk mengatur *layout* atau tata letak dari halaman web. Ada juga yang menggunakan CSS, tetapi kodenya hanya diperoleh dari internet (hasil *download*) tanpa mengetahui lebih dalam akan arti dan makna dari kode-kode CSS tersebut. Hal-hal semacam ini disebabkan karena minimnya pengetahuan mereka tentang CSS sehingga tidak tahu harus memulai dari mana cara membuat halaman *web* yang memanfaatkan teknologi CSS.

Aturan Penulisan Kode CSS

Kode CSS tersusun atas **selector** dan **deklarasi**. *Selector* adalah *tag* HTML yang akan diberi atau dikenai CSS, sedangkan deklarasi adalah properti dan nilai yang akan ditentukan untuk *tag* bersangkutan. Sebagai contoh, perhatikan kode CSS berikut:



```
body {  
    background: black;  
}
```

Pada kode di atas, `body` disebut sebagai *selector* dan `background: black;` disebut sebagai deklarasi. `background` adalah properti dari *tag* `<body>` yang akan di-*set* nilainya, sedangkan `black` adalah nilai yang digunakan atau akan diisikan ke dalam atribut `background`. Sebuah *selector* dapat memiliki lebih dari satu deklarasi, dimana setiap deklarasi harus diakhiri oleh tanda titik koma atau *semicolon* (`;`).

Ada kalanya, dalam membuat kode CSS kita menjumpai satu atau beberapa *selector* yang memiliki deklarasi yang sama dengan *selector* lainnya. Untuk kasus semacam ini, deklarasi cukup ditulis sekali saja, akan tetapi daftar *selector*-nya saja yang harus disebutkan satu persatu dengan memisahkannya menggunakan tanda koma. Berikut ini contoh kode yang akan menunjukkan hal tersebut.

```
body, td, h2 {  
    background: black;  
    color: red;  
}
```

Pada contoh kode di atas, deklarasi akan berlaku untuk *tag* `<body>`, `<td>`, dan `<h2>`.

Penggabungan Kode CSS dengan Dokumen HTML

Proses penggabungan kode CSS dengan dokumen HTML dapat dilakukan melalui 3 (tiga) cara, yaitu:

- ***Embedded Styles***: Menuliskan kode CSS pada bagian *header* dokumen HTML
- ***Linked Styles***: Menuliskan kode CSS pada *file* terpisah dan dipanggil melalui *tag* `<link>` dari dokumen HTML
- ***Inline Styles***: Menuliskan kode CSS langsung pada *tag* yang akan diubah atau diatur *style*-nya

Cara 1: Embedded Styles

Pada cara ini, semua kode CSS ditulis di dalam *tag* `<style>` dan `</style>` dan ditempatkan pada bagian *header* dari dokumen HTML (di dalam *tag* `<head>`). Kode CSS yang ditulis dengan cara seperti ini hanya akan berlaku untuk satu dokumen bersangkutan saja. Artinya, jika Anda ingin menggunakan kode CSS tersebut dari dokumen HTML lain, maka Anda perlu menyalin kode CSS bersangkutan ke dalam dokumen lain yang dituju. Atribut TYPE pada *tag* `<style>` harus diisi dengan “text/css”.

```
<html>
<head>
<title>Embedded Styles</title>
<style type="text/css">
body {
    background: blue;
    color: pink;
}
</style>
</head>
<body>
...
</body>
</html>
```

Cara 2: Linked Styles

Cara ini merupakan cara yang paling banyak digunakan dan direkomendasikan. Pada cara ini, kode CSS akan ditulis secara terpisah pada *file* tersendiri (dengan ekstensi .css). File tersebut selanjutnya dapat dikaitkan atau dihubungkan dengan dokumen HTML melalui *tag* `<link>`. Dengan demikian, kode CSS dalam *file* .css tersebut dapat digunakan oleh beberapa dokumen HTML yang ingin menerapkan *style*. Melalui cara seperti ini, kode CSS akan mudah untuk dipelihara dan dimodifikasi.

Tag `<link>` yang digunakan untuk memanggil *file* .css diletakkan pada bagian *header* dokumen HTML, dengan bentuk umum penggunaan seperti berikut:

```
<link rel="stylesheet" href="nama_file_css" type="text/css" />
```

Contoh:

File style.css

```
body {  
    background: blue;  
    color: white;  
}
```

File sample.html

```
<html>  
<head>  
<title>Linked Styles</title>  
<link rel="stylesheet" href="style.css" type="text/css">  
</head>  
<body>  
...  
</body>  
</html>
```

Dengan menggunakan cara seperti di atas, ada beberapa keuntungan yang diperoleh, yaitu:

- Kode (baik CSS maupun HTML) tampak lebih rapi dan mudah untuk dibaca atau dipahami.
- Lebih mudah dikembangkan atau dimodifikasi.
- Kode CSS dapat digunakan oleh dokumen-dokumen HTML lain yang ingin menerapkan *style* tersebut.
- Perubahan pada *file.css* akan berpengaruh terhadap semua dokumen HTML yang menerapkannya. Artinya perubahan *style* hanya dilakukan sekali pada *file* .css-nya saja.

Cara 3: Inline Styles

Pada cara ini, kode CSS ditulis atau dimasukkan ke dalam atribut STYLE pada *tag* yang akan digunakan. Cara ini merupakan cara yang tidak direkomendasikan karena kode akan tampak rumit dan susah untuk dibaca. Sebagai contoh, jika Anda memiliki beberapa *tag* <h2> dalam dokumen HTML yang Anda buat, maka Anda harus menuliskan *style* pada setiap *tag* yang ada. Jika suatu saat Anda ingin memodifikasi *style* tersebut, maka Anda harus mengubah pada setiap *tag* yang ada.

Berikut ini contoh penggunaan *inline styles* di dalam *tag* HTML:

```
<h2 style="color: red; border-bottom: 1px solid red;">
Heading 2 Pertama</h2>
<h2>Heading 2 Kedua</h2>
```

Pada contoh di atas, *tag* <h2> pertama menggunakan *style* dan *tag* <h2> kedua tidak. Jika Anda ingin *tag* <h2> kedua juga menerapkan *style* yang sama, maka Anda perlu menuliskannya seperti berikut:

```
<h2 style="color: red; border-bottom: 1px solid red;">
Heading 2 pertama</h2>
<h2 style="color: red; border-bottom: 1px solid red;">
Heading 2 kedua</h2>
```

Dengan melihat secara sepintas dari contoh di atas kita bisa menyimpulkan bahwa cara ini merupakan cara yang tidak praktis, yang akan menyebabkan kode menjadi 'gemuk' dan susah untuk dibaca maupun dipelihara.

C. Latihan

Embedded Styles

Latihan 6.1 (css-embedded.html)

```
<html>
<head>
  <title>CSS</title>
  <style type="text/css">
body {
  background: black;
  color: white;
}

h2 {
  color: yellow;
  border-bottom: 1px solid white;
}
</style>
</head>
<body>
<h2>Demo Embedded Styles</h2>

<p>Halaman ini adalah halaman yang sudah diatur style-nya oleh kode CSS </p>

</body>
</html>
```

Linked Styles

Latihan 6.2 (style.css)

```
body {
  background: black;
  color: white;
}
```

(sample.html)

```
<html>
<head>
  <title>CSS</title>
  <link rel="stylesheet" href="style.css" type="text/css" />
</head>
<body>
<h2>Demo Memberi Link Style</h2>

<p>Ini adalah body</p>

</body>
</html>
```

Inline Styles

Latihan 6.3 (css-inline.html)

```
<html>
<head>
  <title>CSS</title>
</head>
<body>
<h2>Demo Inline Styles</h2>

<p style="color: red; text-decoration: underline;">
Contoh paragraf yang menggunakan style.</p>

<p>Contoh paragraf yang tidak menggunakan style.</p>
</html>
```

D. Tugas Kelompok

- 1) Buatlah sebuah halaman *website* tentang tiket penjualan dan penjadwalan !
Misal. Tiket Penjualan Pertandingan Sepak Bola beserta jadwal pertandingannya.
- 2) Diharapkan tiap kelompok berbeda judulnya dengan kelompok lain.
- 3) Buatlah tampilannya menggunakan CSS dengan penggabungan **Embedded Styles**, **Linked Styles**, dan **Inline Styles** !
- 4) Pada tampilan desain, boleh menggunakan icon-icon sebagai pendukung dari tampilannya.