



Panduan lengkap
MENGGUASAI

PEMROGRAMAN

CSS

KOMANG WISWAKARMA

Sanksi Pelanggaran Pasal 72

Undang-undang Nomor 19 Tahun 2002

Tentang Hak Cipta

1. Barangsiapa dengan sengaja melanggar dan tanpa hak melakukan perbuatan sebagaimana dimaksud dalam Pasal 2 Ayat (1) atau Pasal 49 Ayat (1) dan Ayat (2) dipidana dengan pidana penjara masing-masing paling singkat 1 (satu) bulan dan/atau denda paling sedikit Rp 1.000.000,00 (satu juta rupiah), atau pidana penjara paling lama 7 (tujuh) tahun dan/atau denda paling banyak Rp 5.000.000.000,00 (lima miliar rupiah).
2. Barangsiapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu ciptaan atau barang hasil pelanggaran hak cipta atau hak terkait sebagai dimaksud Ayat (1) dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp 500.000.000,00 (lima ratus juta rupiah).

Panduan Lengkap Menguasai Pemrograman CSS

KOMANG WISWAKARMA



www.bukulokomedia.com

Panduan Lengkap Menguasai Pemrograman CSS

Oleh: Komang Wiswakarma

Copyright © Lokomedia, 2010

Hak Cipta dilindungi oleh Undang-Undang

Diterbitkan pertama kali oleh :

Penerbit Lokomedia

Jl. Jambon, Perum. Pesona Alam Hijau 2 No. B4, Kricak
Yogyakarta 55242.

email : redaksi@bukulokomedia.com

website : <http://bukulokomedia.com>

ISBN : 978-979-1758-65-9

Cetakan Pertama : Maret 2010

Dilarang memperbanyak, mencetak ataupun menerbitkan sebagian maupun seluruh isi buku ini tanpa izin tertulis dari penerbit.

Ucapan Terima Kasih

Pertama-tama, saya panjatkan puji syukur kehadapan Ida Sang Hyang Widhi atas segala karunia-Nya sehingga terbitnya buku ini.

Ucapan terima kasih sebesar-besarnya saya berikan kepada seluruh keluarga saya, kedua orangtua, Ni Nyoman Asriani dan Putu Prapta yang selalu mendukung, kedua kakak, Putu Anantawijaya dan Kadek Widyaswari yang selalu memberikan nasihat dan memberikan saran-saran bermanfaat, Dwimas Sukma Agung yang menjadi penyemangat. Serta seluruh warga IT President University dan warga B-1 Dormitory.

Akhir kata, saya ucapkan terimakasih kepada semua yang telah membantu dalam terbitnya buku ini.

Kata Pengantar

Pemrograman web menggunakan CSS (*Cascading Style Sheet*) merupakan salah satu hal yang cukup penting untuk diterapkan pada pembuatan web saat ini. Seiring dengan kebutuhan web yang semakin meningkat dan jalur transaksi web pada internet yang semakin membutuhkan koneksi yang cepat, CSS menyediakan fitur untuk melakukan pemrograman web yang mudah diterapkan dan dapat menyebabkan perubahan yang signifikan dalam dunia web.

CSS menyediakan kemudahan bagi pembuat maupun pengembang web untuk menciptakan sebuah halaman web yang mudah dan cepat diakses serta dengan design halaman web yang luar biasa.

CSS adalah suatu bahasa pemrograman yang diciptakan untuk memisahkan keberadaan konten dan desainnya, sehingga design halaman tersebut dapat digunakan berkali-kali pada halaman yang berbeda, sehingga kita tidak perlu mendeklarasikan design tersebut berulang-ulang kali pada halaman HTML. Dengan berbagai kemudahan tersebut, CSS menjadi salah satu bahasa pemrograman yang paling disarankan dalam pembuatan web.

Pada buku ini akan dijelaskan lebih mendetail tentang struktur dan komponen dari CSS serta ditambahkan pula trik-trik yang dapat digunakan untuk mempercantik web. Buku ini akan membuat pengetahuan Anda tentang pemrograman CSS bertambah serta Anda dapat langsung mempraktekannya pada web yang akan Anda buat.

Akhir kata, penulis juga menyadari masih banyak terdapat kekurangan-kekurangan dalam buku ini. Untuk itu, Anda dapat mengirimkan kritik, saran atau pertanyaan ke alamat email penulis wiswasite@gmail.com atau mengunjungi website penulis di www.wiswakarma.com.

Source code dari buku ini dapat didownload secara gratis di <http://bukulokomedia.com/css-komang.rar>

Gianyar, Bali, Desember 2009

Komang Wiswakarma

Daftar Isi

BAB 1. Pengenalan CSS	2
1.1. Definisi CSS	2
1.2. Asal Mula CSS	3
1.3. World Wide Web Consortium (W3C)	3
1.4. Mengapa Kita Perlu Menggunakan CSS?	4
1.5. Apa itu Style Sheet?.....	4
1.6. CSS VS HTML	5
BAB 2. CSS dan Style Sheet	8
2.1. Bagaimana Style Sheet Bekerja.....	8
2.2. Cara Kerja CSS	9
2.3. Bagian-Bagian Style Sheet	10
2.4. Syntax Style Sheet	12
2.5. Linking dan Embedding	13
2.5.1. Meng-embed Style Sheet di Dokumen HTML.....	14
2.5.2. Menghubungkan Style Sheet di Dokumen HTML.....	14
2.6. Statement	15
2.7. Menuliskan Komentar pada Style Sheet	16
2.8. Menuliskan Komentar pada Dokumen HTML.....	16
2.9. Aturan CSS.....	17
BAB 3. Komponen CSS Selektor	20
3.1. Pengenalan Komponen Selektor	20
3.2. Apa yang Dilakukan Selektor	20

3.3. Tipe-Tipe Selektor.....	20
3.3.1. Selektor Type (Tulisan).....	20
3.3.2. Selektor Class.....	22
3.3.3. Selektor ID	24
3.3.4. Selektor Descendant (Turunan).....	26
3.3.5. Selektor Link Pseudo Class.....	31
3.3.6. Selektor Pseudo Elemen.....	32
3.3.7. Selektor Dynamic Pseudo Class.....	36
3.3.8. Selektor Language	36
3.3.9. Selektor Child.....	37
3.3.10. Selektor First-Child.....	39
3.3.11. Selektor Adjacent (Berdekatan).....	40
BAB 4. Komponen CSS Properti.....	44
4.1. Properti Text Style (Gaya Tulisan).....	46
4.1.1. color.....	46
4.1.2. font-weight.....	47
4.1.3. font-family.....	48
4.1.4. font-size.....	50
4.1.5. font-variant.....	52
4.1.6. font-style.....	53
4.1.7. text-decoration.....	53
4.1.8. text-transform.....	55
4.2. Properti Text Layout (Tampilan Tulisan).....	56
4.2.1. letter-spacing.....	56
4.2.2. word-spacing.....	57
4.2.3. line-height.....	58

4.2.4. vertical-align.....	59
4.2.5. text-indent.....	61
4.2.6. text-align.....	62
4.2.7. direction.....	64
4.2.8. unicode-bidi.....	65
4.3. Properti Background (Tampilan Latar Belakang).....	65
4.3.1. background-color.....	66
4.3.2. background-image.....	67
4.3.3. background-attachment.....	68
4.3.4. background-repeat.....	69
4.3.5. background-position.....	70
4.3.6. background.....	72
4.4. Properti Border (Garis).....	73
4.4.1. border-width, border-top-width, border-left-width, border-right-width, border-bottom-width.....	74
4.4.2. border-color.....	76
4.4.3. border-style.....	77
4.4.4. border, border-top, border-left, border-bottom, border-right.....	79
4.5. Properti Margin (Ruang/Jarak Pembatas).....	80
4.5.1. margin, margin-top, margin-left, margin-bottom, margin-right.....	80
4.6. Properti Padding.....	82
4.6.1. padding, padding-top, padding-left, padding-bottom, padding-right.....	82
4.7. Properti Page Layout (Tampilan Halaman).....	83
4.7.1. position.....	87
4.7.2. top.....	89

4.7.3. left.....	90
4.7.4. bottom.....	92
4.7.5. right.....	93
4.7.6. width.....	95
4.7.7. min-width.....	96
4.7.8. max-width.....	96
4.7.9. height.....	97
4.7.10. min-height.....	98
4.7.11. max-height.....	99
4.7.12. z-index.....	100
4.7.13. visibility.....	101
4.7.14. overflow.....	102
4.7.15. float.....	103
4.7.16. clear.....	104
4.7.17. clip.....	105
4.8. Properti Tipe Elemen.....	106
4.8.1. display.....	107
4.8.2. white-space.....	109
4.8.3. list-style-type.....	110
4.8.4. list-style-image.....	113
4.8.5. list-style-position.....	113
4.8.6. list-style.....	114
4.8.7. border-collapse.....	115
4.8.8. border-spacing.....	115
4.8.9. caption-side.....	116
4.8.10. empty-cells.....	117
4.8.11. table-layout.....	118

4.9. Properti User Interface (Tampilan Pengguna)	119
4.9.1. cursor	119
4.9.2. outline	120
4.9.3. outline-color	121
4.9.4. outline-style	121
4.9.5. outline-width	122
BAB 5. Komponen CSS Nilai Properti	126
5.1. Nilai Panjang	126
5.2. Nilai Persentase	127
5.3. Nilai Warna (Properti Color)	128
5.3.1. Nama Warna	128
5.3.2. Warna Hexadecimal RGB	130
5.3.3. Warna Decimal RGB	131
5.4. Nilai URL	131
5.5. Nilai Kata Kunci	132
5.6. Nilai Bentuk (Shape)	132
BAB 6. CSS Tahap Lanjut	134
6.1. Cascade dan Inheritance	134
6.1.1. Cascade	134
6.1.2. Inheritance (Pewarisan Nilai)	137
6.2. Konten Ter-generate Otomatis	138
6.2.1. String text	139
6.2.2. Counter dan Counters	139
6.2.3. Kata Kunci yang Menyatakan Elemen	140
6.2.4. attr function ()	141

6.2.5. quote	142
6.2.6. Properti counter dan counters	143
6.2.6.1. counter	143
6.2.6.2. counters	145
6.2.6.3. counter-reset	146
6.2.6.4. counter-increment	147
6.3. Media	148
6.3.1. Aturan @media	148
6.3.2. Tipe Media	149
6.3.3. Import Media Spesifik dengan Aturan @import	149
6.4. Printing	150
6.4.1. Aturan @page	150
6.4.2. Properti Page Spesifik	151
6.5. CSS pada Mobile Phone	156
6.5.1. Selektor CSS pada Mobile Phone	157
6.5.2. Properti CSS pada Mobile Phone	157
6.6. Validasi File HTML dan CSS	159
6.3.1. Informasi Dasar Validasi HTML dan CSS	159
6.3.2. Mengapa Perlu Memvalidasi HTML dan CSS?	159
6.3.3. Mengapa Kita Tidak Perlu Memvalidasi HTML dan CSS?	160
6.3.1. Seberapa Sering Kita Memvalidasi Website	161
6.3.2. Kesimpulan	161
BAB 7. Studi Kasus CSS	164
7.1. Kasus 1: Layout Halaman Web	164
7.1.1. Membuat Halaman Web Dua Kolom	164
7.1.2. Membuat Halaman Web Tiga Kolom	171

7.1.3. Menentukan Beda Padding dan Margin.....	173
7.2. Kasus 2: Menu Web	176
7.2.1. Membuat Menu Dropdown Sederhana.....	176
7.2.2. Membuat Menu Dock Apple.....	177
7.3. Kasus 3: Tampilan Gambar	181
7.3.1. Membuat Galeri Foto.....	181
7.3.2. Membuat Efek Pop-up Gambar	184
7.4. Kasus 4: Mouse Event	186
7.4.1. Membuat Efek Hover (Efek Opacity).....	186
7.4.2. Membuat Efek Hover (Efek Border)	187
7.5. Kasus 5: Form Web	188
7.5.1. Merapikan Form Tanpa Tabel.....	188
7.6. Kasus 6: Button Web	190
7.6.1. Membuat Metallic Button.....	190
7.6.2. Membuat Oval Button dengan Efek Hover	193
7.7. Kasus 7: Link Web	195
7.7.1. Membuat Animasi Underline pada Link.....	195
7.8. Kasus 8: Miscellaneous	197
7.8.1. Membuat Tooltips pada Tulisan	197
7.8.2. Animasi Flash pada Background Web	199
7.8.3. Membuat Sudut Melipat pada Elemen.....	202
7.8.4. Membuat Drop-caps pada Paragraf.....	204

Bab 2.

CSS dan Style Sheet



SYNTAX DARI STYLE SHEET

ATURAN CSS

STATEMENT

LINKING DAN EMBEDDING

CARA KERJA CSS

BAGIAN – BAGIAN DARI STYLE SHEET

BAGAIMANA STYLE SHEET BEKERJA?

***Panduan Lengkap
Menguasai Pemrograman CSS***

BAB 2

CSS dan Style Sheet

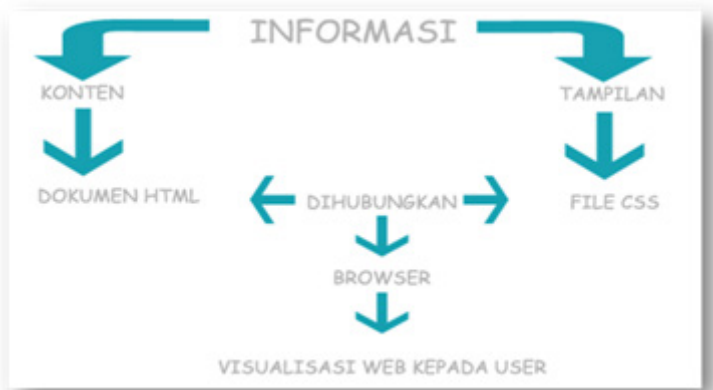
2.1 Bagaimana Style Sheet Bekerja?

Ok, sampai tahap ini kita telah mengetahui bahwa style sheet hanya sebuah file text biasa, atau text yang di-embed di bagian <head> dari sebuah dokumen HTML, yang sebagaimana mungkin membantu memisahkan konten dari tampilannya.

Konten dari sebuah halaman adalah sebuah file HTML. Dan tampilannya adalah file style sheet (CSS). Tetapi bagaimana cara menggabungkan kedua elemen web tersebut, sehingga menjadi sebuah halaman web yang tervisualisasikan dengan baik di mata user (pengunjung)?

Style sheet sebagai kumpulan dari beberapa perintah format tampilan dokumen, menyarankan browser bagaimana menggambarkan sebuah halaman web.

Disini CSS hanya “menyarankan” sebuah web browser bagaimana menggambarkan sebuah halaman bukan “memaksa” browser untuk menggambarkan sebuah web, karena sebenarnya CSS lebih hanya menyarankan bagaimana seharusnya sebuah halaman web ditampilkan oleh browser. Ini adalah sebuah perbedaan yang penting antara “memaksa” dan “menyarankan”. Lihat gambar 2.1.

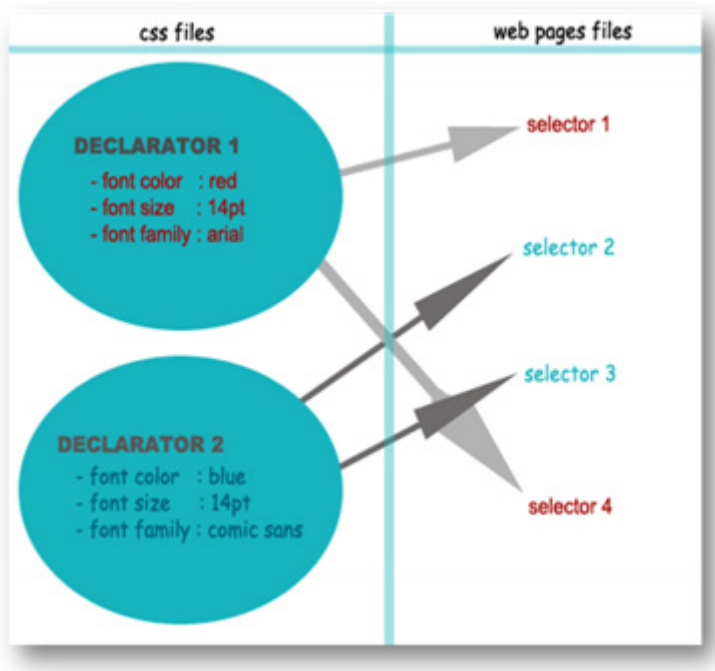


Gambar 2.1 Sketsa kerja dari CSS dan HTML

2.2 Cara Kerja CSS

CSS bekerja dengan menggunakan dua buah elemen penting untuk pemformatan tampilannya, diantaranya elemen selektor dan deklarator. Dua buah elemen ini berfungsi sebagai penentu format tampilan dan lainnya menempatkan format tampilan tersebut.

Deklarator berisi beberapa perintah-perintah CSS untuk menentukan format tampilan dari sebuah elemen pada halaman web. Sedangkan selektor adalah sebuah perintah lanjutan dari deklarator dan berfungsi menempatkan format tampilan dari deklarator pada halaman web. Lihat gambar 2.2.



Gambar 2.2 Kerja dari selector dan deklarator

Contoh dari halaman web yang memuat deklarator dan selektor adalah sebagai berikut:

```
<html>  
<head><title>Contoh deklarator dan selektor</title>  
<style type = "text/css">  
  .tampil {
```

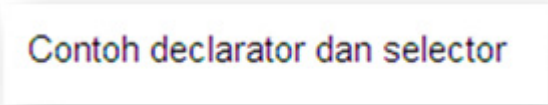
```

        font-family : arial ;
        font size : 14pt
    }
</style>
</head>

<body>
<table>
<tr>
<td class="tampil">Contoh deklarator dan selektor</td>
</tr>
</table>
</body>
</html>

```

Pada kode halaman web diatas, terlihat bahwa **.tampil** merupakan deklarator dan **class="tampil"** merupakan selektor dimana format tampilan dari deklarator akan ditampilkan. Kode halaman web tersebut akan menghasilkan tampilan seperti pada gambar 2.3.



Gambar 2.3 Hasil pembuatan CSS sederhana pada halaman HTML

2.3 Bagian-Bagian Style Sheet

Setiap file Cascading Style Sheets (CSS), baik itu dalam bentuk .css file ataupun ter-embed pada tag <head> dari sebuah halaman web, mempunyai sebuah kelanjutan kode yang disebut dengan statements. Statements mempunyai dua tugas penting, yaitu:

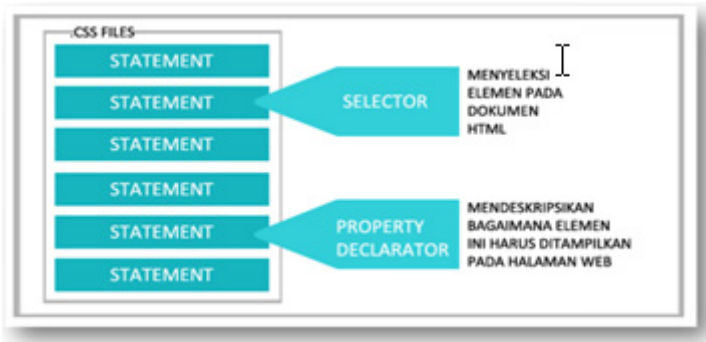
- Mengidentifikasi efek dari statement tersebut pada dokumen HTML.
- Statement menginformasikan browser, bagaimana menggambarkan sebuah halaman web.

Dengan elemen, yang dimaksud elemen disini adalah tag paragraf, link – link, daftar list dan seterusnya. Dalam aturan teknis HTML, elemen adalah tag yang dibuat pada halaman web. Bagian dari statement yang mengidentifikasi elemen



dari sebuah halaman dinamakan selektor. Selektor memilih elemen dari sebuah web.

Bagian dari statement yang menentukan bagaimana elemen yang telah dipilih selektor tadi seharusnya ditampilkan pada browser dinamakan deklarator atau deklarasi. Sebuah deklarator dapat menampung beberapa properti tampilan untuk elemen yang telah dipilih oleh selektor. Lihat gambar 2.4.



Gambar 2.4 Sketsa kerja dari deklarator; selektor dan statement

```
body {  
  font-family: Helvetica, sans-serif;  
  font-size: 1em;  
  text-align: justify;  
}
```

Ini adalah sebuah statement dalam sebuah style sheet. Selektor dari statement ini adalah body. Ini berarti statement ini akan berakibat pada element <body> pada setiap halaman web yang terhubung dengan style sheet ini.

Statement ini mempunyai deklarator dengan tiga properti (font-family, font-size dan text-align). Ini berarti setiap elemen yang dipilih oleh selektor ini akan berpengaruh pada tiga properti tersebut.

Deklarator ini akan mempunyai efek samping pada font yang dipakai pada elemen <body>, sehubungan dengan ukuran font tersebut dan posisi text. Lihat gambar 2.5.



Gambar 2.5 Struktur dari kode CSS

2.4 Syntax Style Sheet

Di dunia web HTML, tidak ada istilah seperti “web standar”. Sebagai hasilnya browser-browser memberi kemudahan pada syntax HTML yang tidak ditulis menurut aturan bahasa HTML yang seharusnya. Lain halnya dengan CSS, CSS memiliki sebuah spesifikasi standar.

Artinya jika anda tidak mengerti dasar-dasar aturan dari sebuah CSS, browser akan sulit menerjemahkan format tampilan yang anda inginkan bahkan bukan tidak mungkin, browser tidak akan membaca format tampilan CSS anda. Berikut ini adalah beberapa aturan dasar yang dapat anda terapkan pada file CSS anda;

- Setiap statement harus mempunyai selektor dan deklarator. Deklarator harus dibuat setelah selektor dan diapit 2 buah kurung kurawal (kurung kurawal buka dan tutup).
- Properti dari deklarator dipisahkan menggunakan titik koma (;)
- Setiap properti mempunyai nama dan diikuti oleh titik dua dan selanjutnya nilai (value) dari properti tersebut. Terdapat banyak tipe dari nilai (value) ini, tetapi setiap properti hanya bisa mengambil beberapa value seperti yang telah diatur pada spesifikasi CSS. Spesifikasi ini akan memberitahu nilai yang mungkin diletakkan pada setiap properti.
- Beberapa properti bisa menampung beberapa nilai, seperti pada contoh diatas (font-family), nilai baru harus dipisah dengan koma dan spasi. Kembali lagi, untuk aturan ini anda dapat melihat pada spesifikasi CSS



yang paling baru. Agar sesuai dengan rekomendasi CSS yang dibuat oleh W3C.

- Beberapa nilai (value) mempunyai ukuran tertentu seperti pt (point) dan px (pixel), seperti yang terdapat pada font-size. Anda tidak dapat memberi spasi antara nilai dan ukurannya seperti 45pt atau 45px.
- Seperti pada HTML, spasi dapat membuat dokumen CSS anda mudah dibaca dan dimengerti. Anda dapat mempergunakan spasi untuk memudahkan anda untuk mengubah atau memperbaharui file CSS Anda.

2.5 Linking dan Embedding

Kita sudah mempelajari jauh tentang style sheet, apa style sheet tersebut, bagaimana style sheet bekerja dalam sebuah browser dan mengapa kita lebih memilih menggunakan style sheet. Jika sebuah file style sheet mengandung informasi yang memberitahu browser untuk menampilkan sebuah halaman tertentu, bagaimana sebuah browser tahu bahwa sebuah halaman tersebut seharusnya menggunakan style sheet dan dimana browser dapat menemukannya?

Ada dua cara, bagaimana cara sebuah style sheet dapat berhubungan dengan sebuah dokumen HTML

Pertama, style sheet dapat di-embed di elemen head pada dokumen html . Kita akan mempelajari bagaimana cara untuk meng-embed format CSS ini pada dokumen HTML, kita juga akan melihat mengapa cara ini bukan menjadi cara yang bagus untuk mengaplikasikan CSS pada sebuah dokumen.

Kita sudah banyak mengetahui bagaimana cara me-manage sebuah website dengan menggunakan style sheet. Kita juga mengetahui bagaimana sebuah style sheet dapat berpengaruh pada keseluruhan web, dan bagaimana perubahan itu tampil pada setiap halaman web yang dihubungkan dengan file style sheet tersebut.

Jika kita hanya meng-embed style sheet ini pada bagian <head> di dokumen web, kita akan kehilangan keuntungan-keuntungan menggunakan CSS ini. Dan ini berarti kita belum memisahkan konten dari tampilan seutuhnya.

Cara kedua, yaitu dengan meletakkan sebuah link pada <head> dari HTML document. Link ini berfungsi menghubungkan file dokumen HTML dengan style sheet yaitu CSS. Dengan link ini, ketika browser membuka halaman web, dan membaca kode link style sheet ini, maka browser akan mendownload file CSS ini, kemudian menggunakannya untuk menampilkan halaman web.



Anda dapat meletakkan beberapa link dari style sheet pada sebuah dokumen HTML. Ini tidak berarti browser menggunakan setiap style sheet yang terhubung dengan dokumen. Biasanya browser memilih salah satu dari style sheet yang terhubung tersebut (Secara teori, browser akan memberitahukan atau menanyakan style sheet yang akan digunakan).

2.5.1 Meng-embed Style Sheet di Dokumen HTML

Style sheet bisa di-embed pada elemen `<head>` di dokumen HTML. Format style sheet itu sendiri akan diletakkan di dalam elemen `<style>` seperti berikut;

```
<style type="text/css">Kode CSS akan diletakkan disini</style>
```

Seperti yang dijelaskan diatas, meng-embed style sheet bukanlah solusi yang baik untuk membuat web design, karena kita akan kehilangan salah satu kelebihan pada CSS adalah kemampuan untuk memodifikasi seluruh halaman website dengan hanya membuat perubahan pada satu file (CSS).

2.5.2 Menghubungkan Style Sheet di Dokumen HTML

Ingat, bahwa halaman web terhubung dengan style sheet. Style sheet tidak tahu halaman mana yang akan dihubungkan dengan format yang dideklarasikan pada file CSS tersebut. Untuk menghubungkan sebuah halaman (dokumen HTML) dengan sebuah file style sheet, anda dapat memberikan link pada elemen `<head>` dari dokumen HTML anda, dengan mengikuti syntax berikut;

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

Penjelasan dari setiap atribut elemen penghubung diatas sebagai berikut:

```
rel="stylesheet"
```

Menyatakan bahwa ini adalah sebuah link, dan akan memberitahu browser bahwa link ini berisi style sheet yang akan dipakai selanjutnya.

```
type="text/css"
```

Secara teori, style sheet bisa ditulis dalam berbagai bahasa pemrograman. Style sheet yang kita bicarakan disini adalah Cascading Style Sheet (CSS). Atribut ini berfungsi untuk memberitahu browser, format apa yang berada dalam file style sheet tersebut.



```
href="URLtempatfilecss.css"
```

Ini menyatakan dimana browser harus mencari file style sheet. Ini bisa dalam bentuk relative atau absolute URL. Ingat, relative URL adalah relative dengan dokumen HTML, tidak seperti URLs dalam style sheet yang relative dengan style sheet.

2.6 Statement

Sekarang kita sudah memahami bagaimana sebuah style sheet bekerja dan kita juga sudah mengetahui bagaimana seharusnya sebuah style sheet terlihat. Sekarang saatnya kita ke tahap selanjutnya untuk memahami CSS. CSS mempunyai beberapa jenis statement, yaitu: **@rules statement**.

Ada beberapa @rules dari CSS, diantaranya:

a. @import

Pada CSS1, @rule statement yang valid hanyalah @import. Aturan @import menginstruksikan browser untuk menginput style sheet dari sebuah sumber dan menggunakannya bersama-sama dengan style sheet sebelumnya. Style sheet dengan sebuah aturan @import di dalamnya adalah turunan dari style sheet yang diimport. Kita akan mendalami turunan dan pewarisan style sheet (cascade inheritance) lebih lanjut nanti.

Aturan @import mempunyai bentuk seperti berikut,

```
@import URL (http://www.westciv.com/styles/style1);
```

Bentuk lainnya anda dapat mengganti kurung dengan tanda petik.

```
@import "http://www.westciv.com/styles/style1";
```

URL yang dipakai bisa berasal dari file di server lokal atau di URL lain.

CSS 2 memperkenalkan konsep media dan memperbolehkan meng-import style sheet berdasarkan kondisi dari media dokumen akan ditampilkan.

b. @media

Aturan dari @media diperkenalkan pada CSS 2. Aturan @media memperbolehkan sebuah bagian dari style sheet hanya dipergunakan jika dokumen web ditampilkan pada media tertentu. Sebagai contohnya, anda bisa membuat sebuah aturan @media yang dapat diaplikasikan ketika sebuah halaman web di-print.



Aturan `@media` seperti meng-embed sebuah style sheet ke dalam style sheet. Jadi aturan `@media` dapat diaplikasikan pada halaman web sesuai dengan medium halaman web tersebut itu ditampilkan. Inilah salah satu kelebihan dari `@media`, aturan `@media` dapat mengubah tampilan sebuah halaman web sesuai dengan medium yang digunakan untuk menampilkannya.

Aturan `@media` mempunyai bentuk yang tetap yaitu menggunakan `@media` sebagai prefix dan kemudian diikuti oleh nama media tertentu, kemudian diikuti oleh format – format CSS yang akan diaplikasikan pada aturan `@media` tersebut dan ditutup dengan kurung kurawal. Sebagai contohnya aturan `@media` berikut akan mempunyai efek pada halaman web, sehingga ketika di-print halaman web tersebut akan berwarna putih.

```
@media print {  
    body {background-color: white}  
}
```

Kita akan mempelajari `@media` pada bab berikutnya.

c. `@page`

Aturan `@page` memperbolehkan kita untuk mengatur dengan baik bagaimana halaman web kita akan di-print. Bentuk dari `@page` adalah menuliskan `@page` pada awal style sheet, kemudian diikuti dengan format – format CSS dengan menggunakan kurung kurawal buka dan tutup. Contoh dari aturan `@page`,

```
@page {  
    margin: 5%  
}
```

Aturan `@page` akan dijelaskan lebih lanjut pada bab berikutnya.

2.7 Menuliskan Komentar pada Style Sheet

Karena CSS dapat dikembangkan dan disebarkan, kita dapat melengkapi style sheet dengan komentar-komentar penjelas. Ini untuk membantu untuk pengembangan kode – kode CSS tersebut nantinya. Komentar pada style sheet mempunyai bentuk seperti berikut:

```
/* ini adalah komentar penjelas pada style sheet */
```

2.8 Menuliskan Komentar pada Dokumen HTML

Karena beberapa browser tidak mendukung penggunaan cascading style sheets



(CSS), maka untuk memastikan agar style sheet ini tidak ditampilkan dalam bentuk text ke dalam halaman web, maka kita dapat menambah tag komentar HTML. Bentuk awal dari komentar HTML adalah sebagai berikut:

```
<!-- isi dari komentar HTML -->
```

Komentar HTML sebaiknya tidak diletakkan pada external style sheet, tapi diletakkan pada tag <head> dari halaman web yang menggunakan internal style sheet.

2.9 Aturan CSS

Aturan pada CSS adalah fokus utama ketika kita akan memulai sebuah design web menggunakan CSS. CSS mempunyai dua buah komponen utama yaitu selektor dan deklarator. Sekarang kita akan menilik lebih jauh penggunaan selektor dan deklarator pada halaman web.

Komponen selektor akan menyeleksi bagian elemen tertentu pada halaman web yang akan dipengaruhi oleh style sheet. Selektor akan menyeleksi elemen tertentu menggunakan format yang telah ditentukan sebelumnya pada deklarator.

Format nama dari selektor mengikuti format yang telah dijabarkan sebelumnya pada deklarator. Selektor dapat berupa tag yang sudah ada sebelumnya pada halaman web maupun kita dapat membuat selektor unik atau baru yang dijabarkan pada deklarator.

Komponen deklarator akan membuat sebuah format untuk selektor. Pada deklarator terdapat beberapa macam deklarasi CSS baik itu berupa tampilan maupun komposisi yang nantinya akan diterapkan pada selektor tertentu.

Deklarator akan memerintahkan browser untuk bagaimana cara menampilkan sebuah halaman web. Deklarator dapat berupa tag – tag elemen yang sudah ada sebelumnya pada halaman web ataupun dapat berupa deklarator unik yang kita buat sendiri.

Salah satu contoh elemen web yang paling sering dipakai yaitu paragraf. Untuk menggambarkan suatu paragraf pada halaman web, caranya cukup mudah yaitu dengan menggunakan deklarator p dan selektor <p>. Misalkan kita menginginkan agar setiap paragraf menggunakan huruf Sans Comic

Berikut adalah deklarator dari paragraf

```
p
```



Nah, untuk elemen-elemen lain seperti `<body>`. Kita dapat membuat deklaratornya dengan menghilangkan tanda `<` dan `>`. Jadi kita dapat membuat deklaratornya pada file style sheet dengan cara seperti ini:

```
body
```

Banyak elemen-elemen deklarator lainnya yang dapat anda gunakan untuk mengubah tampilan halaman web anda. Untuk pemaparan selanjutnya mengenai deklarator-deklarator ini anda dapat lihat pada bab selanjutnya.

Nah, sebelumnya kita mempunyai rencana untuk membuat tampilan setiap paragraf menggunakan huruf Comic sans. Sebelumnya kita hanya membuat deklarator dari paragraf tersebut yaitu menggunakan deklarator `p`. Untuk dapat membuat setiap paragraf menggunakan huruf Comic sans, kita harus menambahkan baris kode untuk memerintahkan browser menggambarkan paragraf tersebut dengan mengubah setiap huruf dalam paragraf tersebut dengan menggunakan huruf jenis Comic sans. Baris kode yang dimaksud yaitu:

```
{font-family: Comic Sans MS}
```

Dan selanjutnya jika kita menginginkan agar hurufnya berwarna merah, kita dapat menambahkan baris kode berikut,

```
{font-family: Comic Sans MS ; color: red}
```

Nah, sekarang kita akan menggabungkan selektor dan deklarator tersebut, hasilnya akan seperti berikut:

```
p {font-family: Comic Sans MS ; color: red}
```

Kita akan mempelajari tentang berbagai aturan-aturan dalam CSS, karena bagian ini merupakan hal yang sangat penting dalam men-design web dengan menggunakan style sheet. Aturan dalam CSS akan digunakan untuk menciptakan sebuah format CSS yang lebih kompleks dan benar.

