YAYASAN PRIMA AGUS TEKNIK

HTML CSS dan Javascript

Muhammad Sholikhan, S.Kom., M.Kom

HTML, CSS dan Javascript

Penulis :
Muhammad Sholikhan, S.Kom., M.Kom

ISBN :

Editor :
Dr. Joseph Teguh Santoso, S.Kom., M.Kom.

Penyunting :
Dr. Mars Caroline Wibowo. S.T., M.Mm.Tech

Desain Sampul dan Tata Letak :
Irdha Yunianto, S.Ds.,M.Kom.

Penebit :
Yayasan Prima Agus Teknik Bekerja sama dengan
Universitas Sains & Teknologi Komputer (Universitas STEKOM)

Redaksi :
Jl. Majapahit no 605 Semarang
Telp. (024) 6723456
Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

Distributor Tunggal :
Universitas STEKOM
Jl. Majapahit no 605 Semarang
Telp. (024) 6723456
Fax. 024-6710144
Email : info@stekom.ac.id

KATA PENGANTAR

Puji syukur pada Tuhan Yang Maha Esa bahwa buku yang berjudul "**HTML, CSS & JavaScript**" ini dapat diselesaikan dengan baik. Pernahkah kalian berpikir bagaimana sebuah website bisa dibuat dengan tampilan yang menarik dan user friendly serta dapat berjalan secara interaktif terhadap pengunjung/pengguna website itu sendiri? Sebuah website agar mempunyai UI dan UX yang baik tidak hanya menggunakan satu bahasa pemrograman saja, kita harus mengombinasikan beberapa bahasa pemrograman agar menghasilkan sebuah website yang baik. Sesuai dengan judul buku, buku ini memuat tentang HTML, CSS, dan Javascript. HTML (Hypertext Markup Language) merupakan bahasa pemrograman standar untuk dokumen yang dirancang sebagai kerangka atau susunan sebuah website yang kemudian akan diatur beberapa komponen supaya lebih terstruktur dan mempercantik kerangka HTML menggunakan CSS (Cascading Style Sheet). Terakhir akan ditambahkan sebuah bahasa pemrograman javascript yang mempunyai peran sebagai logika dan cara berpikirnya, seperti saat tombol ditekan akan muncul sesuatu. Javascript dapat disisipkan ke sebuah halaman web dengan tag script.

Buku ini cocok bagi anda yang tertarik untuk memulai membuat sebuah website karena berisikan pemahaman dasar mengenai bahasa pemrograman HTML, CSS, dan Javascript mulai dari teori, aturan dan tata cara penulisan coding yang baik, dan cara menghubungkan antara HTML, CSS, dan Javascript. Semoga buku ini bermanfaat bagi pembaca, akhir kata penulis ucapkkan terima kasih.

Penulis, September 2022

Muhammad Sholikhan, S.Kom., M.Kom.

DAFTAR ISI

BAB 1
HTML

1.1 MEMBUAT HALAMAN DASAR DENGAN HTML

HyperText Markup Language (HTML) adalah bahasa web. Saat kita membuka halaman web di browser web seperti Internet Explorer, Firefox, atau Safari, browser akan mengunduh dan menampilkan HTML. Pada intinya, HTML hanyalah sebuah dokumen, sama seperti dokumen yang dibuat di word. Program seperti Microsoft Word digunakan untuk melihat dokumen pengolah kata karena microsoft word tahu cara membaca dan menampilkan teks. Demikian juga dengan web, browser web adalah program yang mengetahui cara membaca dan menampilkan dokumen yang dibuat dengan HTML.

Dokumen Word Process dapat dibuat dan dibaca dengan satu program. Di sisi lain, dokumen HTML membutuhkan program yang berbeda untuk pembuatan dan pembacaan; Kita tidak dapat membuat dokumen HTML menggunakan browser. Untuk membuat dokumen HTML kita membutuhkan program yang disebut editor. Editor ini bisa sesederhana program Notepad yang disertakan dengan Microsoft Windows atau serumit Eclipse dan Microsoft Visual Studio. Umumnya kita dapat menggunakan program yang sama untuk membuat dokumen HTML yang digunakan untuk membuat program PHP.

1.2 MEMAHAMI BLOK BANGUNAN HTML

Dokumen HTML dapat disimpan di folder *Documents* dalam komputer. Namun jika dibuka, hanya kita saja yang bisa melihat isinya. Untuk menyiasati hal ini, kita memerlukan lebih banyak sumber daya, sebut saja sebagai server web. Menyimpan dokumen di server web memungkinkan orang lain untuk dapat melihat dokumen kita. Server web adalah komputer yang menjalankan perangkat lunak khusus yang mengetahui cara mengirim (atau menyajikan) halaman web ke banyak orang pada saat yang bersamaan. Dokumen HTML diatur dalam urutan tertentu, dengan bagian-bagian tertentu. Mereka terstruktur sedemikian rupa agar browser web dapat membaca dan menampilkannya. Karena hal tersebut, ketika kita membuat dokumen HTML, kita harus mengikuti strukturnya dan menyiapkan dokumen agar browser dapat membacanya.

1.3 JENIS DOKUMEN

Browser web dapat menampilkan beberapa jenis dokumen, tidak hanya HTML, jadi saat membuat dokumen web, hal pertama yang harus lakukan adalah memberi tahu browser tentang jenis dokumen apa yang akan digunakan. Kita karus menetapkan jenis dokumen dengan baris khusus HTML di bagian atas dokumen. Dalam istilah teknis, tipe dokumen disebut *Document Text Declaration*, atau disingkat DTD. Dalam versi HTML sebelumnya, developer terus-menerus menyalin dan menempelkan jenis dokumen ke dalam dokumen karena panjang dan rumit. Dengan dirilisnya versi terbaru HTML, yang disebut HTML5, jenis dokumen menjadi sangat disederhanakan. Jenis dokumen untuk HTML5 adalah:

```
<!doctype html>
```

Ini akan menjadi baris pertama dari setiap dokumen HTML yang kita buat, sebelum membuat dokumen lainnya. Setiap kali, kita perlu menampilkan HTML dan menyertakan jenis dokumennya yang kadang-kadang disebut *doctype*.

Bisa jadi kita akan tergoda untuk menggunakan <!doctype html5>, tetapi tidak ada nomor versi yang terkait dengan jenis dokumen HTML5. Karena ketika versi HTML berikutnya

keluar, kita tidak perlu lalgu memperbarui atau mengubah semua jenis dokumen yang telah kita buat pada HTML sebelumnya ke versi HTML6. Ada berbagai jenis dokumen lain yang lebih lama, diantaranya adalah:

HTML 4.01 Strict <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

HTML 4.01 Transitional<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

XHTML 1.0 Strict <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

XHTML 1.1 DTD <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

Jenis dokumen lain selain diatas juga ada, dan kebanyakan sama-sama rumit dan sulit untuk diingat. Jika kita melihat jenis dokumen ini di halaman web, kita akan tahu bahwa halaman tersebut mungkin menggunakan sintaks yang sedikit berbeda untuk membuat dokumen HTML-nya. Dokumen HTML terdiri dari huruf dan kata yang diapit tanda kurung siku, terkadang disebut tanda kurang dari atau lebih besar dari:

< >

Ini adalah elemen utama dalam dokumen HTML dan disebut sebagai elemen root:

<html>

Biasanya, elemen HTML memiliki tag pembuka < dan penutup >. Elemen ditutup dengan garis miring di depan nama elemen. Melihat <html> dalam dokumen berarti nantinya dokumen tersebut akan memiliki </html> untuk menutup elemen tersebut. Fungsi pembukaan <html> dan penutup </html> ini untuk membentuk dokumen yang dibungkus di dalam elemen-elemen tersebut.

Bagian dari Dokumen HTML

Selanjutnya, kita akan melihat tiga bagian utama dari dokumen HTML, kita juga akan melihat perbedaan antara dokumen HTML dan halaman HTML?. Kedua istilah tersebut dapat dipertukarkan. Daftar dibawah ini menunjukkan seluruh dokumen HTML.

Contoh 1 Daftar Halaman Web Dasar

```
<!doctype html>
<html>
<head>
<title>My First Document</title>
</head>
<body>
<div>My Web Page</div>
</body>
</html>
```

Teks HTML ini akan muncul di browser web seperti berikut ini:

My Web Page

Halaman memiiki judul di title bar atau tab bar browser dan teks yang menyatakan isi dari dokumen HTML tersebut. Bagian selanjutnya dalam bab ini menjelaskan cara menyempurnakan halaman ini dengan lebih banyak elemen HTML dan teks.

Elemen root

Meskipun bukan bagian dari dokumen HTML, elemen root adalah yang membungkus seluruh dokumen, muncul sebagai hal pertama setelah *doctype* dan hal terakhir dalam dokumen.

Elemen root dibuka dengan:

<html>

Elemen root ditutup dengan:

</html>

Bagian head dan elemen judul

Bagian head dokumen berisi informasi tentang dokumen itu sendiri. Bagian head dibuka dengan:

<head>

Bagian head ditutup dengan:

</head>

Bagian head tidak boleh disamakan dengan menu pada halaman itu sendiri. Bagian head adalah elemen di balik layar halaman, ini dapat berisi banyak informasi tentang halaman. Informasi tersebut mencakup hal-hal seperti judul halaman, bahasa halaman (Inggris, Indonesia, dan sebagainya), jenis informasi yang disajkan, dan hal-hal lain yang umum pada halaman tersebut. Elemen deskriptif di bagian head ini terkadang disebut elemen meta. Disetiap bagian head harus selalu memiki elemen judul. Elemen judul adalah elemen yang akan muncul di title bar browser web atau sebagai judul tab di browser web, ini ditunjukkan oleh gambar dibawah ini.

Gambar 1.1 Elemen judul muncul di tab atau title bar browser web.

Bagian body

Bagian body merupakan jantung dari halaman web. Ini adalah tempat untuk menempatkan semua teks dan gambar sebuah halaman, sebut saja ini adalah isi dari web.

Bagian body dibuka dengan:

<body>

Bagian body ditutup dengan:

</body>

Sama seperti bagian head yang berisi elemen lain seperti judul dan informasi meta, bagian body juga dapat berisi beberapa elemen HTML. Misalnya, di dalam bagian body ini dapat ditemukan semua tautan dan elemen gambar bersama dengan paragraf, tabel, dan apa pun yang diperlukan untuk menampilkan halaman.

1.4 MEMBUAT HTML YANG BAIK

Halaman web yang baik disusun dalam urutan yang logis. Ini berarti elemen-elemen harus ditempatkan diberbagai urutan tertentu agar dapat dibaca dengan benar oleh browser web. Ketika aturan, ketika membuka setiap elemen maka harus menutup elemen tersbeut juga menggunakan tag tertentu yang digunakan dnegan tanda kurung siku dan garis miring..

Menggunakan elemen yang sesuai

Halaman web menggunakan beberapa elemen halaman yang juga disebut sebagai tag. Beberapa elemen tersebut disajikan pada tabel dibawah ini:

Tabel 1.1 berbagai jenis elemen dan penggunaanya

Elemen	***Deskripsi***	***Jenis penggunaan***
<a>	Anchor	Membuat link ke halaman lain atau bagian dari dokumen yang sama.
 	Jeda baris	Memasuki jeda baris atau karakter kembali.
	Bagian dari halaman	Membuat area keseluruhan atau pembagian logis pada halaman, seperti bagian heading/menu, area konten, atau footer.
<form>	Form web	Membuat formulir web untuk menerima input pengguna.
<h1> hingga <h6>	Heading	Membuat wadah untuk judul, seperti teks judul.
<hr>	Hard rule	Membuat garis horizontal.
<img>	Image	Wadah untuk gambar.
<input>	Input	Elemen untuk menerima input pengguna.
<link>	Link sumber	Tautan ke sumber daya untuk halaman tersebut; jangan bingung dengan elemen Anchor
<p>	Paragraf dalam halaman	Membuat paragraf tekstual atau area dan wadah lain untuk teks.
<script>	Tag skrip	Menunjukkan skrip web atau program. Juga sering ditemukan di bagian kepala.
	Span	Membuat wadah untuk elemen. Sering digunakan bersama dengan informasi gaya.

Markup semantik merupakan istilah dalam penggunaan berbagai jenis elemen pada waktu yang tepat, ini berkaitan dengan konsep struktur dan layout elemen. Pertimbangkan manfaat markup semantik ini:

- **Meningkatkan hasil pencarian.** Manfaat utama markup semantik adalah pengunjung dan search engine sama-sama dapat menemukan informasi yang mereka butuhkan.
- **Menyederhanakan pemeliharaan**. Manfaat sekunder dari markup semantik adalah memudahkan pemeliharaan di kemudian hari.

Ketika sebuah halaman secara semantik benar dan HTML yang valid, maka akan terbentuk dengan baik.

1.5 PERBEDAAN ANTARA ELEMEN DIV DAN SPAN

Elemen dan keduanya adalah jenis wadah, tetapi dengan beberapa kualitas yang berbeda. Secara default, elemen memiliki lebar tak terbatas (setidaknya ke tepi browser), yang dapat kita lihat dengan menerapkan batas ke salah satunya, seperti ini:

```
<div style="border:1px solid green;">Hello</div>
```

Namun, elemen hanya selebar teks yang dikandungnya. Oleh karena itu, baris HTML berikut membuat batas hanya di sekitar kata Hello, yang tidak meluas ke tepi kanan browser

```
<span style="border:1px solid green;">Hello</span>
```

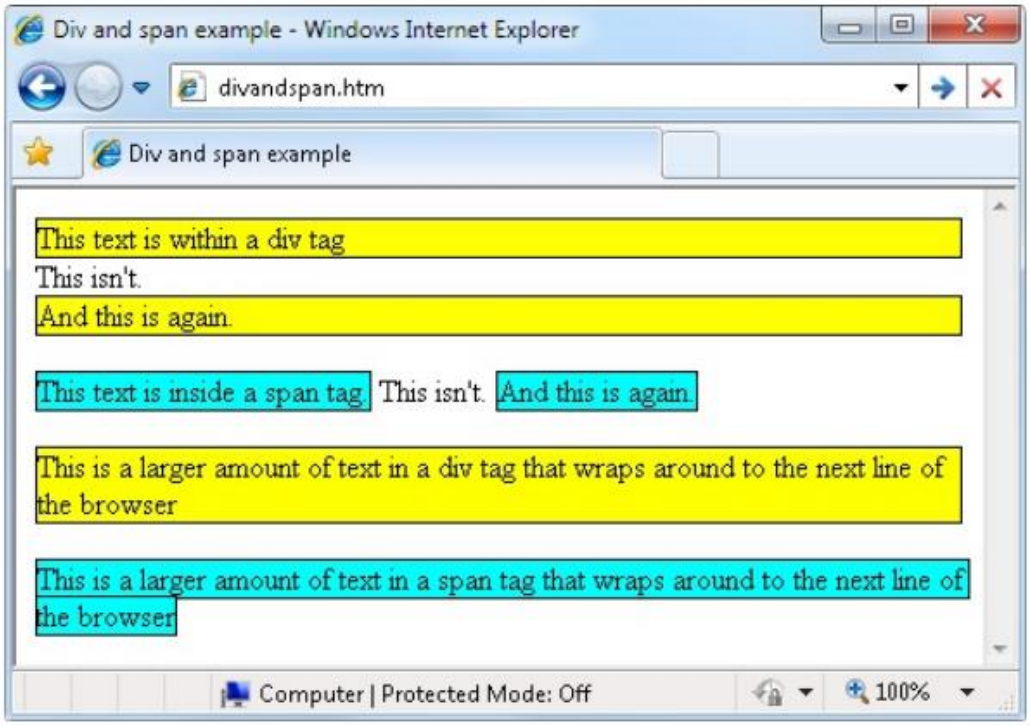
Juga, elemen mengikuti teks atau objek lain saat mereka membungkusnya, dan karena itu dapat memiliki batas yang rumit. Misalnya, pada contoh koding html dibawah, saya menggunakan CSS untuk membuat latar belakang semua elemen menjadi yellow, untuk membuat semua elemen menjadi cyan, dan untuk menambahkan batas pada keduanya, sebelum kemudian membuat beberapa contoh dan bagian.

Contoh 2 div dan span

```
<!DOCTYPE html>
<html>
<head>
<title>Div and span example</title>
<style>
div, span { border :1px solid black; }
div { background-color:yellow; }
span { background-color:cyan; }
</style>
</head>
<body>
<div>This text is within a div tag</div>
This isn't. <div>And this is again.</div><br>
<span>This text is inside a span tag.</span>
This isn't. <span>And this is again.</span><br><br>
<div>This is a larger amount of text in a div that wraps around
to the next line of the browser</div><br>
<span>This is a larger amount of text in a span that wraps around
to the next line of the browser</span>
</body>
</html>
```

Gambar 1.2 menunjukkan seperti apa contoh ini di browser web. Meskipun hanya dicetak dalam nuansa abu-abu dalam buku ini, gambar tersebut dengan jelas menunjukkan

bagaimana elemen meluas ke tepi kanan browser, dan memaksa konten berikut untuk muncul di awal posisi pertama yang tersedia di bawahnya.



Gambar 1.2 Berbagai elemen dengan lebar berbeda

Gambar tersebut juga menunjukkan bagaimana elemen menjaga dirinya sendiri dan hanya menggunakan ruang yang diperlukan untuk menyimpan kontennya, tanpa memaksa konten berikutnya muncul di bawahnya. Misalnya, di dua contoh gambar di bawah, kita juga dapat melihat bahwa ketika elemen membungkus tepi layar, elemen tersebut mempertahankan bentuk persegi panjang, sedangkan elemen hanya mengikuti alur teks (atau konten lainnya) mereka mengandung.

Catatan: Karena tag hanya bisa berbentuk persegi panjang, tag ini lebih cocok untuk memuat objek seperti gambar, kotak, kutipan, dan sebagainya, sedangkan tag paling baik digunakan untuk menyimpan teks atau atribut lain yang ditempatkan satu demi satu inline lain, dan yang harus mengalir dari kiri ke kanan (atau kanan ke kiri dalam beberapa bahasa)

Menempatkan teks pada halaman

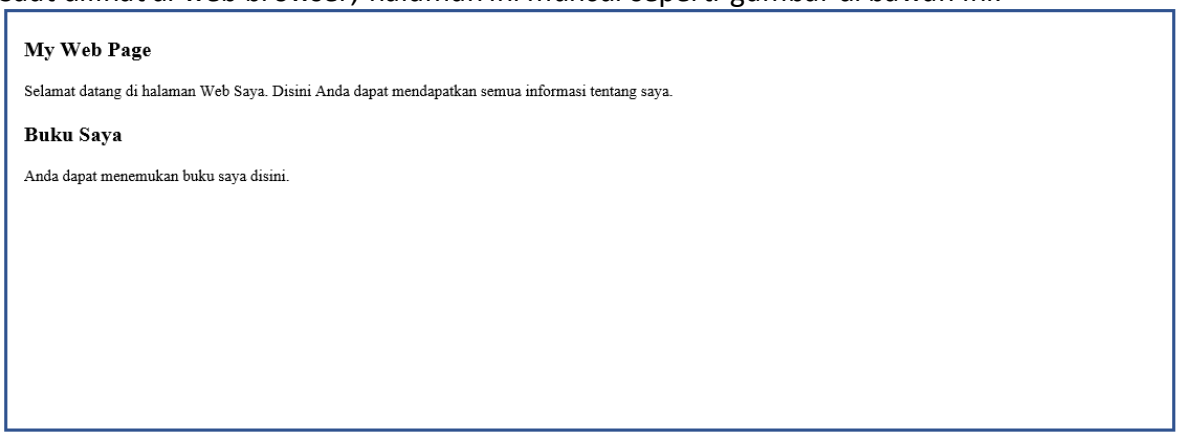
Ada banyak cara untuk menyisipkan teks ke dalam halaman web dan banyak elemen yang sesuai untuk menyimpan teks. Elemen heading seperti <h1>, <h2>, hingga <h6>, adalah tempat yang tepat untuk meletakkan heading, sedangkan <p>, , dan adalah wadah yang sesuai untuk teks bentuk yang lebih panjang, seperti paragraf. Daftar kode HTML dibawah ini menunjukkan halaman web sederhana dengan dua judul dan beberapa paragraf.

Contoh 3 Daftar Halaman Web dengan Judul dan Paragraf

```
<!doctype html>
<html>
<head>
<title>My First Document</title>
</head>
<body>
```

```
<h1>My Web Page</h1>
<p>Selamat datang dihalaman Web Saya. Disini kita dapat mendapatkan semua informasi tentang saya.</p>
<h2>Buku Saya</h2>
<p>Anda dapat menemukan buku saya disini.</p>
</body>
</html>
```

Saat dilihat di web browser, halaman ini muncul seperti gambar di bawah ini.



Gambar 1.3 Halaman web sederhana dengan judul dan paragraf.

Seperti yang kita lihat dari gambar diatas, informasi pada halaman menyertakan elemen <h1> yang diikuti dengan paragraf, elemennya adalah . Paragraf ditutup dengan elemen dan heading menggunakan elemenen <h2>. Judul keuda ditutup dengan elemen </h2>.

Buat Halaman Pertama Anda

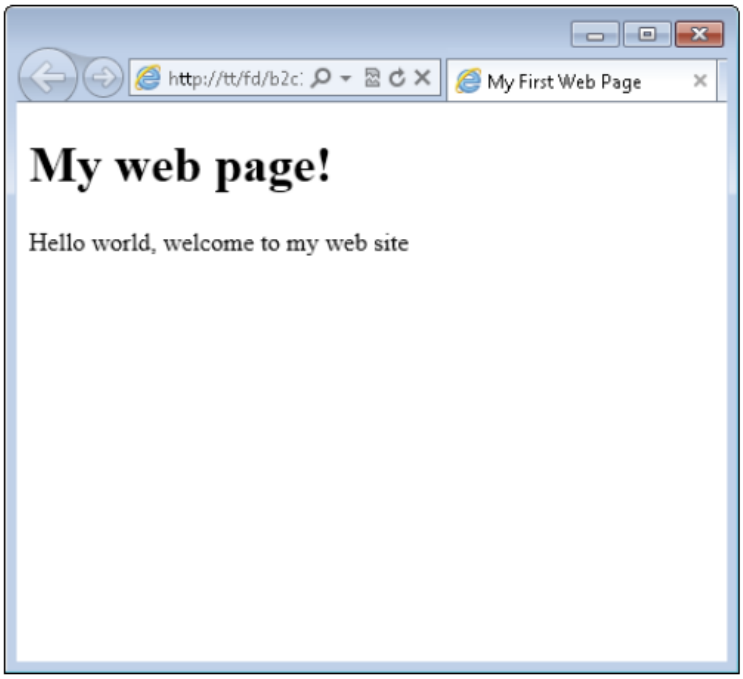
Untuk membuat HTML kita harus menggunakan text editor bukan pengolah kata seperti Microsoft Word. Microsoft Word atau program serupa seperti Pages di Mac menambahkan segala macam informasi pemformatan tambahan yang menghalangi pembuatan HTML, gunakan program seperti Notepad. Semakin sederhana bentuk HTML maka akan semakin baik. Mac menyertakan program bernama TextEdit yang dapat digunakan untuk membuat dokumen teks biasa — tetapi hati-hati: Program TextEdit akan mencoba menyimpan file dalam *Rich Text Forma*t (RTF) secara default. Saat membuat atau menyimpan file dengan **TextEdit**, pilih Usual Text dari menu **Format File drop-down**.

Untuk membuat halaman pertama Anda, ikuti langkah-langkah berikut ini:

1. Buka Text editor
2. Di text editor, masukkan HTML berikut.

```
<!doctype html>
<html>
<head>
<title>My First Web Page</title>
</head>
<body>
<h1>My web page!</h1>
<p>Hello world, welcome to my web site</p>
</body>
</html>
```

3. Simpan file dengan nama firstpage.html
   Simpan file persis seperti namanya, menggunakan huruf kecil di seluruh nama. Apache, server web yang digunakan untuk mengirim file ke browser Anda, peka huruf besar/kecil untuk nama file, jadi tetap menggunakan huruf kecil akan menghemat banyak masalah. Pastikan ekstensinya adalah .html dan bukan .txt atau ekstensi lainnya. Simpan file ke root dokumen Anda. Lokasi root dokumen tergantung pada bagaimana kita menginstal Apache dan pada jenis sistem yang kita gunakan. Jika kita menggunakan penyedia hosting, maka ini adalah titik di mana kita mengunggah file ke sistem mereka.
4. Buka browser web untuk load page
   Di browser web, arahkan ke http://localhost/firstpage.html. Setelah itu, kita akan melihat halaman seperti gambar dibawah ini.



Gambar 1.4 halaman pertama kita yang ditampilkan oleh browser

Memilih elemen level blok atau sebaris

Ketika mempertimbangkan jenis elemen mana yang akan ditambahkan ke halaman, pikirkan terlebih dahulu, apakah kita menginginkan elemen tersebut diperluas ke seluruh lebar halaman atau tidak.

- **Elemen level blok**: Elemen dan <p> keduanya dikenal sebagai elemen level blok. Elemen tingkat blok ditampilkan di seluruh lebar halaman; tidak ada yang bisa muncul di samping atau di samping elemen level blok. Pada dasarnya, anggap elemen level blok memiliki carriage return setelahnya.
- **Elemen sebaris**: Elemen tertentu, terutama elemen , dianggap sebagai elemen sebaris, yang berarti elemen lain dapat muncul di sebelahnya. Dengan kata lain, elemen sebaris tidak memiliki carriage return setelahnya

Memasukkan jeda baris dan spasi

Dalam pembuatan halaman, ada kalanya untuk menyisipkan jeda baris. Untuk melakukan tindakan ini dalam pengolah kata, cukup menekan tombol **Enter** atau **Return** pada keyboard. Hal-hal yang tidak begitu sederhana dalam HTML. Tidak peduli berapa kali tombol Enter ditekan dalam dokumen HTML, teks akan tetap ditampilkan pada baris yang sama di browser web. Perhatikan kode pada daftar dibawah ini.

Contoh 4 DaftarMencoba Memasukkan Carriage Returns ke dalam HTML

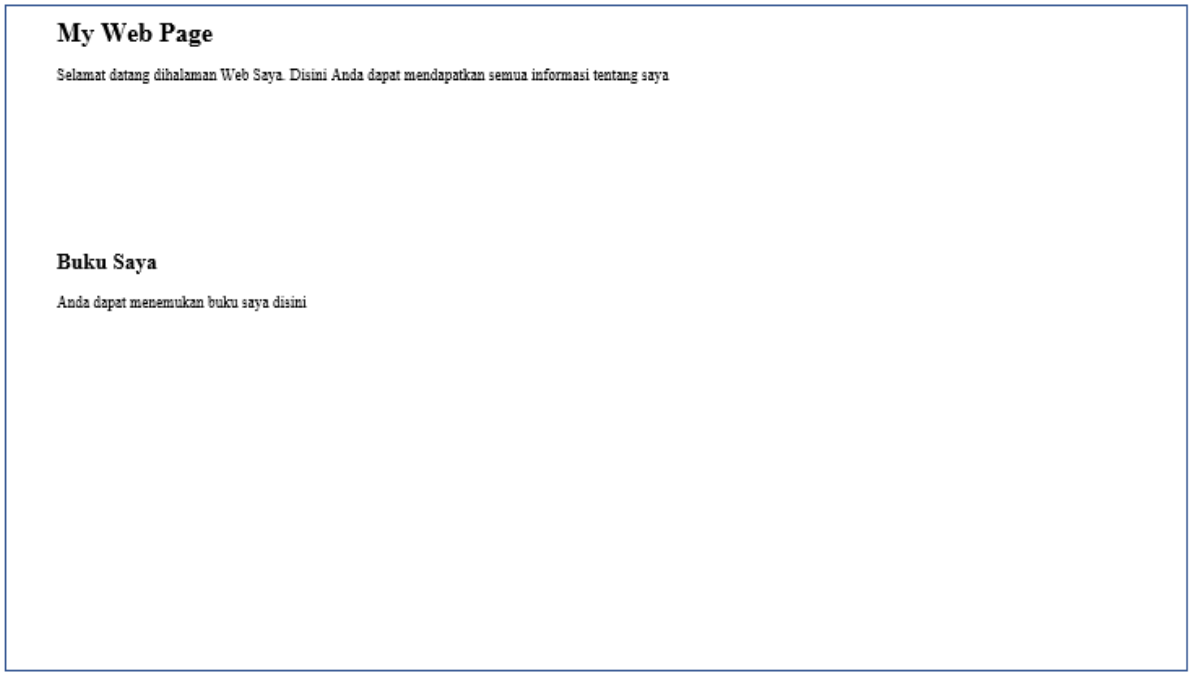
```
<!doctype html>
<html>
<head>
<title>My First Document</title>
</head>
<body>
<h1>My Web Page</h1>
<p>Selamat datang dihalaman Web Saya. Disini kita dapat mendapatkan semua informasi tentang saya.</p>


<h2>Buku Saya</h2>
<p>Anda dapat menemukan buku saya disini.</p>
</body>
</html>
```

Jika dilihat melalui web browser, outputnya sama seperti Gambar 1.5 pada bab sebelumnya. kita tidak melihat baris kosong antara paragraf pertama dan judul kedua. Hal yang sama terjadi pada spasi tambahan di HTML. Tidak peduli berapa kali spasi pada keyboard di dokumen web ditekan, maka outputnya pada html web adalah 1 spasi. Tag
 digunakan untuk menyisipkan jeda baris ke halaman web. Lihat kode daftar dibawah ini. Daripada menggunakan tombol **Enter** (atau **Return** di Mac) akan jauh lebih baik untuk menggunakan tag
 untuk menambahkan carriage return:

*Contoh 5 Menggunakan
 untuk Line Breaks*

```
<!doctype html>
<html>
<head>
<title>My First Document</title>
</head>
<body>
<h1>My Web Page</h1>
<p>Selamat datang dihalaman Web Saya. Disini kita dapat mendapatkan semua informasi tentang saya.</p>
<br>
<br>
<br>
<br>
<br>
<h2>Buku Saya</h2>
<p>Anda dapat menemukan buku saya disini.</p>
</body>
</html>
```

Saat dilihat di browser, efek yang diinginkan akan ditampilkan, seperti yang diilustrasikan pada gambar di bawah ini.



Gambar 1.5 menggunakan tag
 untuk memasukkan carriage returns.

Terkadang kita akan melihat garis miring tambahan di beberapa tag seperti
 sehingga akan ditulis sebagai
. Ini adalah peninggalan dari XHTML tetapi tidak diperlukan untuk HTML5. Kita harus melihat bahwa
 tidak memiliki mitra penutup, seperti </br>. Kita dapat menggunakan
 apa adanya, tanpa khawatir harus menutupnya. Menambahkan spasi ke HTML dilakukan dengan   entitas terkadang ditulis sebagai . Namun, ada cara yang lebih baik untuk mencapai spasi dalam HTML, terutama melalui penggunaan *Cascading Style Sheets* (CSS). Oleh karena itu, penggunaan   entitas tidak akan dibahas demi metode yang lebih umum dan didukung lebih luas melalui CSS.

Membuat dokumen lebih mudah dirawat

Developer sering menggunakan comment untuk mencatat informasi di balik layar tentang halaman atau tentang kode mereka, dan comment tidak ditampilkan di halaman web. Misalnya, komentar di halaman web mungkin seperti "Saya menambahkan ini pada 19/10/2012" atau "Ditambahkan untuk mendukung inisiatif penjualan kami." Jika kita mengunjungi halaman web, kita dapat melihat komentar tersebut hanya dengan melihat file HTML halaman tersebut.

Comment HTML dibuka dengan sintaks ini:

<!--

Comment HTML ditutup dengan sintaks ini:

-->

Segala sesuatu yang muncul dari awal <!-- hingga bagian pertama --> dianggap sebagai bagian dari komentar. Daftar dibawah ini berisi contoh dokumen HTML dengan komentar.

Contoh 6 Menambahkan Komentar HTML

```
<!doctype html>
<html>
```

```
<head>
<title>My First Document</title>
</head>
<body>
<h1>My Web Page</h1>
<p>Selamat datang dihalaman Web Saya. Disini kita dapat mendapatkan semua informasi tentang saya.</p>
<!—Menambahkan informasi tentang bukuk saya 10/1/2022 -->
<h2>Buku Saya</h2>
<p>Anda dapat menemukan buku saya disini.</p>
</body>
</html>
```

Contoh 6a Multi-line comment

```
<!doctype html>
<html>
<head>
<title>My First Document</title>
</head>
<body>
<h1>My Web Page</h1>
<p>Selamat datang dihalaman Web Saya. Disini kita dapat mendapatkan semua informasi tentang saya.</p>
<!—
Menambahkan informasi tentang bukuk saya
10/1/2012
-->
<h2>Buku Saya</h2>
<p>Anda dapat menemukan buku saya disini.</p>
</body>
</html>
```

Dalam komentar ini, kita dapat melihat bahwa teks sebenarnya dari komentar tersebut diindentasi, yang memunculkan poin penting lainnya: Menggunakan lekukan saat membuat dokumen akan sangat membantu. Dokumen lebih mudah dibaca dan dipelihara nanti ketika elemen diindentasi, sehingga kita dapat melihat dengan jelas secara visual elemen mana yang "di dalam" elemen lainnya.

Menambahkan daftar dan tabel

Daftar dan tabel membantu untuk mewakili jenis informasi tertentu. Misalnya, daftar pohon di halaman kampus paling baik diwakili dengan daftar seperti ini:

Pinus
Beringin
Palm

Tetapi jika ingin memasukkan lebih banyak informasi tentang pohon, tabel adalah format yang lebih baik:

Jenis Pohon	Deskripsi
Pinus	Pohon umum di lapangan
Beringin	Ada beberapa pohon beringin halaman kampus
Palm	Ada 5 pohon palm di halaman kampus dekat gazebo

HTML memiliki tag untuk membuat daftar dan tabel. Tabel 1.2 menjelaskan berbagai elemen tersebut.

Tabel 1.2 Daftar Umum dan Elemen Tabel dalam HTML

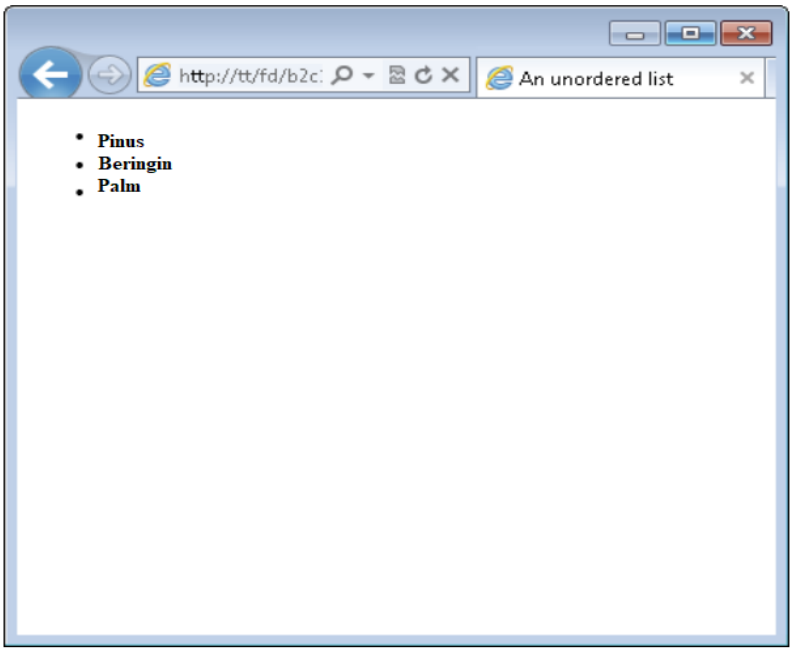
Eleemn	***Jenis***	***Deskripsi***
	List Item	Digunakan bersama dengan atau untuk membuat daftar informasi.
	Order List	Daftar informasi yang diurutkan, digunakan bersama dengan .
<table>	Tabel	Digunakan dengan <tr>, <td>, dan elemen lain untuk membuat tabel untuk menyajikan informasi.
<td>	Sel Tabel	Membuat sel di baris tabel
<th>	Header Tabel	Sel tabel yang merupakan heading.
<tr>	Row Tabel	Membuat deretan tabel.
	Unorder List	Terkait dengan dan untuk membuat daftar informasi

Saat membuat daftar, kita memiliki dua pilihan jenis daftar yang akan dibuat: daftar yang diurutkan atau daftar yang tidak diurutkan. Daftar berurut digunakan untuk hal-hal seperti membuat garis besar, sementara daftar yang tidak berurutan membuat hampir semua jenis daftar lainnya. Daftar berikut menunjukkan HTML yang digunakan untuk membuat daftar *unordered* standar.

Contoh 7 Membuat Daftar Unordered

```
<!doctype html>
<html>
<head>
<title>An unordered list</title>
</head>
<body>
<ul>
 <li>Pine</li>
 <li>Oak</li>
 <li>Elm</li>
</ul>
</body>
</html>
```

Jika dilihat di browser, HTML ini menghasilkan halaman seperti pada gambar dibawah ini.

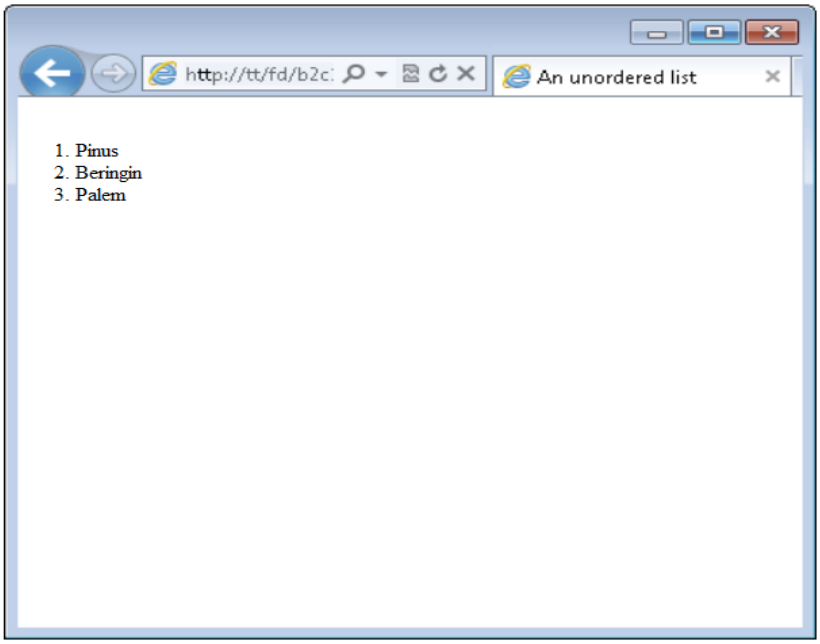


Gambar 1.6 Daftar tidak berurutan

Daftar tidak berurutan menggunakan gaya default untuk daftar, yang menambahkan poin di samping setiap item. Gaya peluru ini dapat diubah, atau bisa juga tidak di ikut sertakan dalam CSS. Membuat daftar terurut berarti cukup mengubah elemen <ul> menjadi <ol>. Prosesnya akan terlihat seperti ini:

```
<ol>
 <li>Pine</li>
 <li>Oak</li>
 <li>Elm</li>
</ol>
```

Saat dilihat di browser, poin dari contoh sebelumnya diganti dengan angka, seperti pada gambar dibawah ini.



Gambar 1.7 Daftar yang dipesan

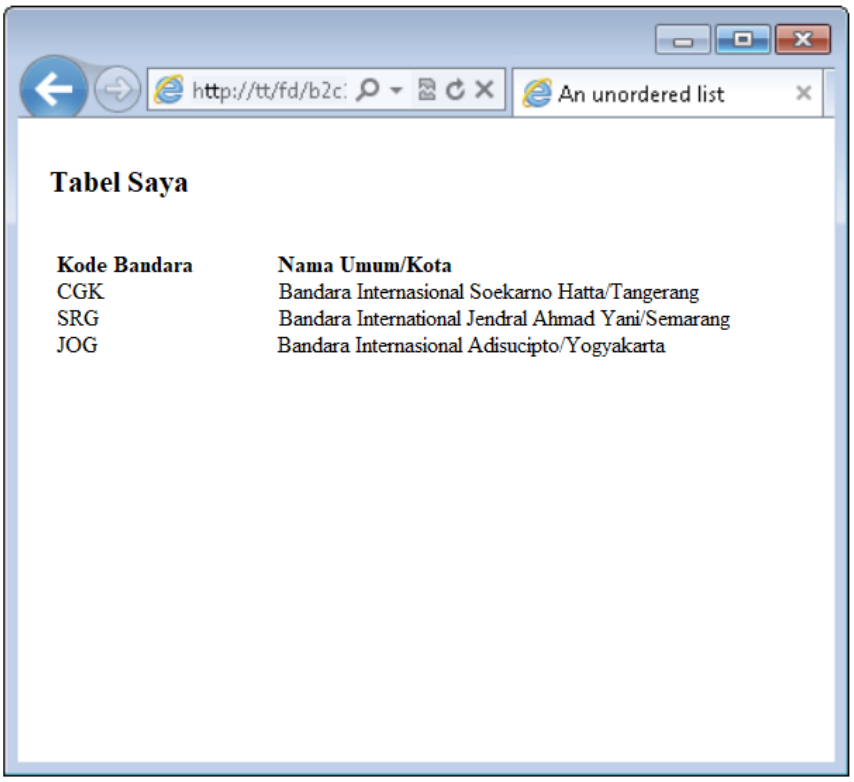
1.6 BERLATIH MEMBUAT TABEL

Untuk membuat tabel ikuti langkah berikut ini:

1. Buka text editor
2. Buat new text document pada text editor.
   Sebagian besar text editor akan terbuka dengan dokumen kosong. Jika terdapat sesuatu dalam dokumen, maka hapus terlebih dahulu sebelum melanjutkan.
3. Masukkan HTML berikut:

```
<!doctype html>
<html>
<head>
<title>My First Web Page</title>
</head>
<body>
<h1>My Table</h1>
<table>
        <tr>
        <th>Airport Code</th>
        <th>Common Name/City</th>
        </tr>
<tr>
        <td>CWA</td>
        <td>Central Semarang Airport</td>
        </tr>
        <tr>
        <td>ORD</td>
        <td>Chicago O'Hare</td>
        </tr>
        <tr>
        <td>LHR</td>
        <td>London Heathrow</td>
        </tr>
</table>
</body>
</html>
```

4. Simpan file sebagai table.html.
   Simpan file dengan ekstensi .html. File harus disimpan di root dokumen.
5. Lihat file di browser Anda.
   Buka browser web kita dan ketik http://localhost/table.html ke dalam bilah alamat. Prosesnya akan terlihat seperti gambar dibawah ini.



Gambar 1.8 Tabel latihan.

Perhatikan bahwa tabel tidak memiliki batas di sekitarnya. Jika ingin menambahkan batas, terus lakukan latihan ini. Jika tidak, lanjutkan ke bagian berikutnya.

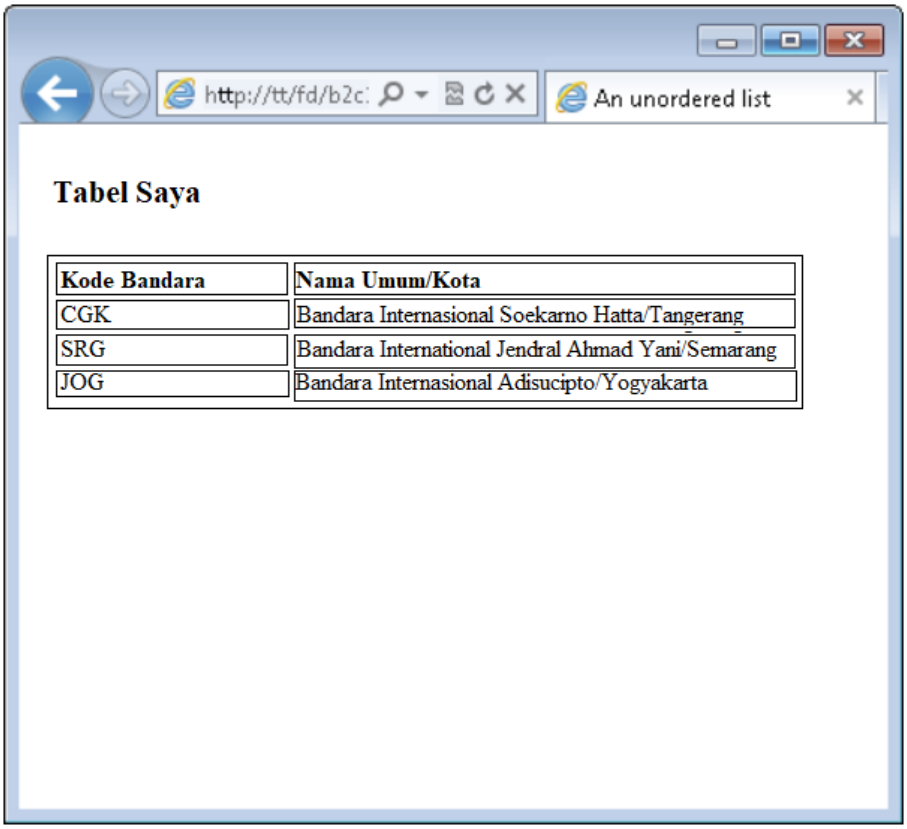
6. Buka table.html di text editor.
   Jika kita menutup text editor, buka lagi dan muat table.html.
7. Ubah kode di table.html menjadi berikut:

```
<!doctype html>
<html>
<head>
<title>My First Web Page</title>
</head>
<body>
<h1>My Table</h1>
<table border=”1”>
        <tr>
                <th>Airport Code</th>
                <th>Common Name/City</th>
        </tr>
        <tr>
                <td>CGK</td>
                <td>Bandara International Soekarno-Hatta/Tangerang</td>
        </tr>
        <tr>
                <td>SRG</td>
                <td>Bandara Internasional Jendral Ahmad Yani/Semarang</td>
        </tr>
        <tr>
```

```
        <td>JOG</td>
        <td>Bandara Internasional Adi Sucipto/Yogyakarta</td>
        </tr>
</table>
</body>
</html>
```

Perhatikan bahwa satu-satunya perubahan adalah menambahkan spasi dan kemudian border="1" di dalam elemen <table>.

8. Refresh table.html di browser Anda.
   Jika kita menutup browser, buka kembali dan buka http://localhost/table.html. Jika browser kita masih terbuka, tekan Ctrl+R untuk merefresh halaman (Command+R di Mac). Selanjutnya, hasil dari tabel adalah sebagai berikut.



Gambar 1.9 Tabel dengan batas di sekitar setiap sel

Saat menambahkan border="1" ke elemen <table>, ini disebut atribut. Atribut membantu untuk mendeskripsikan atau mendefinisikan elemen lebih lanjut atau memberikan detail tambahan tentang bagaimana elemen tersebut harus berperilaku.

1.7 MENYERTAKAN TAUTAN DAN GAMBAR DI HALAMAN WEB ANDA

Apa jadinya web tanpa tautan — dan juga gambar? Tidak banyak web sama sekali. Tautan adalah item yang kita klik di dalam halaman web untuk terhubung atau memuat halaman lain, dan ketika kita berbicara tentang gambar, yang kita maksud adalah ilustrasi dan foto. Bagian ini membahas cara menambahkan tautan dan juga gambar ke halaman web Anda

Menambahkan tautan

Tautan ditambahkan dengan <a>, atau elemen anchor. Atribut href memberi tahu elemen anchor tujuan tautan. Tautan itu sendiri dapat ditambahkan ke apa saja di halaman.

Misalnya, kita dapat menautkan setiap pohon yang disebutkan di bagian sebelumnya ke artikel tentang masing-masing jenis pohon tersebut. Ketika sesuatu ditautkan, browser akan memberikan umpan balik visual bahwa ada tautan dengan menyorot dan menggarisbawahi area yang ditautkan. Seperti elemen HTML lainnya, elemen <a> memiliki tag penutup yang sesuai yang digunakan untuk memberi tahu browser kapan harus berhenti menyorot dan menggarisbawahi tautan.

Menautkan ke halaman lain

Menautkan ke halaman lain, baik di situs yang sama atau di situs yang berbeda, dilakukan dengan cara yang sama. Sebagai contoh, lihat HTML berikut:

Here's a link to web universitas stekom

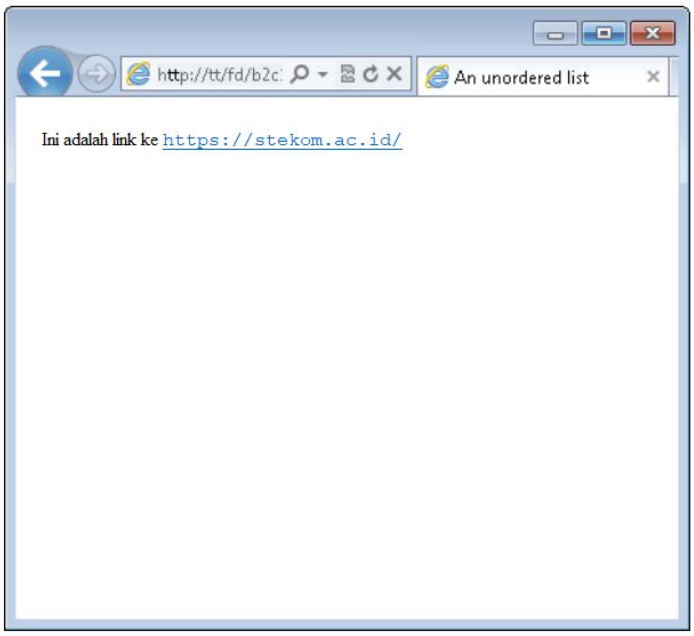
Baris ini menggunakan elemen paragraf untuk membuat kalimat, "Ini tautan ke web universitas stekom." Ini adalah web, kita memutuskan untuk benar-benar memberikan tautan sehingga pengunjung dapat mengklik kata-kata tertentu dan dibawa ke halaman itu. kita melakukannya dengan elemen <a> bersama dengan atribut href. Dalam hal ini, elemen <a> terlihat seperti ini:

Atribut href menunjuk ke URL http://www.braingia.org dan diapit oleh tanda kutip. Teks yang akan disorot kemudian muncul, diikuti dengan tag penutup . Berikut adalah latihan untuk menerapkan tautan ini.

1. Buka text editor.
2. Tempatkan HTML berikut pada text editor Anda:

```
<!doctype html>
<html>
<head>
<title>Link</title>
</head>
<body>
<p>Here's a link to <a href="http://www.braingia.
org">Steve Suehring's site</a></p>
</body>
</html>
```

3. Simpan file sebagai link.html.
   File harus disimpan ke root dokumen kita dengan nama link.html.
4. Buka browser kita dan lihat halamannya.
   Buka browser web kita dan arahkan ke http://localhost/link.html dengan memasukkan URL tersebut ke bilah alamat. kita akan melihat halaman seperti itu pada gambar dibawah ini.



Gambar 1.10 Halaman dengan tautan

Selalu tutup elemen <a> dengan tag penutup yang sesuai. Kesalahan yang sering terjadi adalah membiarkan elemen <a> terbuka, sehingga semua teks berikutnya disorot sebagai tautan. Contoh dan latihan menunjukkan cara menautkan ke halaman di situs web yang berbeda. Membuat tautan ke halaman di situs yang sama dilakukan dengan cara yang sama, tetapi alih-alih menyertakan skema *Uniform Resource Identifier* (URI) dan nama host (bagian http://www.stekom.ac.id dari contoh ini), kita hanya dapat menautkan ke halaman itu sendiri. Jika kita telah mengikuti latihan sebelumnya, maka kita harus memiliki halaman bernama table.html. Berikut HTML untuk membuat tautan ke table.html. HTML latihan sebelumnya disertakan sehingga kita dapat melihat keseluruhan konteks untuk tautan:

```
<!doctype html>
<html>
<head>
<title>Link</title>
</head>
<body>
<p>link ke <a href=”table.html”>contoh tabel</a></p>
</body>
</html>
```

Seperti sebelumnya, tautan terdapat di dalam elemen <p> tetapi perhatikan bahwa atribut href sekarang hanya menunjuk ke table.html.

Memahami tautan absolut versus relatif

Tautan yang ditunjukkan pada contoh sebelumnya disebut tautan relatif karena tidak dimulai dengan skema *Uniform Resource Identifier* (URI) (http://) atau garis miring awal (/). Tautan relatif mengasumsikan bahwa target (table.html dalam contoh) berada di direktori atau folder yang sama dengan dokumen atau halaman yang ditautkan. Dalam kasus contoh, tautan relatif berfungsi karena halaman saat ini, link.html, dan halaman yang ditautkan, table.html, keduanya ada di root dokumen.

Jika kedua halaman tidak berada di direktori yang sama (dengan kata lain, jika table.html berada di folder bernama tabel di root dokumen dan file link.html ada di folder

bernama link di root dokumen), maka kita perlu untuk membuat tautan absolut. Tautan absolut memberi tahu server di mana tepatnya mencari untuk menemukan target. Misalnya, tautan absolut mungkin terlihat seperti /tables/table.html. Tautan ini memberi tahu server bahwa ia perlu mulai mencari di root dokumennya untuk direktori yang disebut tabel dan kemudian harus menemukan file bernama tables.html di direktori tabel.

Gunakan tautan absolut ketika kita perlu memberikan referensi yang tepat atau absolut ke target yang ditautkan. Gunakan tautan relatif ketika sumber daya yang ditautkan akan selalu ditemukan di tempat yang sama dengan halaman yang menautkannya. Jika lokasi halaman atau target berubah, maka tautan relatif akan berhenti berfungsi.

Menautkan dalam satu halaman

Terkadang kita ingin menautkan dalam halaman yang sama. kita dapat melakukan ini pada halaman yang sangat panjang, di mana kita memiliki daftar isi di bagian atas dan kemudian artikel lengkap di bagian bawah halaman. Membuat tautan dalam halaman menggunakan elemen <a> yang sama dengan yang kita lihat, kali ini dengan atribut nama. Daftar dibawah menunjukkan HTML untuk membuat anchor dalam halaman.

Contoh 8 Anchor Dalam Halaman

```
<!doctype html>
<html>
<head>
<title>Link</title>
</head>
<body>
<ul>
 <li><a href=”#pine”>Pine</a></li>
 <li><a href=”#oak”>Oak</a></li>
 <li><a href=”#elm”>Elm</a></li>
</ul>
<p><a name=”pine”> Ada pohon pinus di halaman.</a><p>
<p><a name=”oak”> Ada beberapa pohon beringin halaman kampus.</a><p>
<p><a name=”elm”> Ada 5 pohon palm di halaman kampus dekat gazebo.</a><p>
</body>
</html>
```

Pada daftar kode diatas, tag href yang ditambahkan ke setiap item daftar menggunakan tanda pound atau hash (#). Ini adalah kunci yang digunakan untuk memberi tahu browser bahwa sumber daya akan ditemukan di halaman yang sama. Kemudian di HTML kita melihat elemen <a>lainnya, kali ini menggunakan atribut name. Atribut name itu sesuai dengan masing-masing atribut href dari halaman sebelumnya. Jadi, tidak ada lagi yang perlu ditambahkan untuk menambahkan tautan dalam halaman. Hanya perlu menggunakan tanda pound untuk menunjukkan bahwa sumber daya ditemukan kemudian di halaman dan kemudian menggunakan atribut name untuk membuat elemen lain cocok dengan itu.

Membuka tautan di jendela baru

Terkadang kita ingin membuat tautan terbuka di tab baru atau jendela baru. Saat pengunjung mengeklik tautan yang ditentukan sedemikian rupa, browser akan membuka tab baru dan memuat sumber daya tertaut di tab baru itu. Situs yang ada akan tetap terbuka di browser pengunjung juga. Jangan membuat setiap tautan terbuka di jendela baru. kita harus melakukannya hanya jika masuk akal, seperti yang mungkin terjadi ketika pengunjung berada di tengah-tengah proses yang panjang di situs web kita dan perlu menautkan ke referensi

sumber daya atau situs lain, seperti direktori kode ZIP atau persyaratan perjanjian layanan. Juga, apakah tautan terbuka di tab baru atau jendela baru tergantung pada browser; kita tidak dapat mengontrolnya.

Ini dapat dilakukan dengan menambahkan atribut target ke elemen <a> kita dengan nilai khusus, _blank. Misalnya, contoh sebelumnya menunjukkan cara membuat tautan ke situs web universitas stekom, www.stekom.ac.id. Ingatlah bahwa tautannya terlihat seperti ini:

web universitas stekom

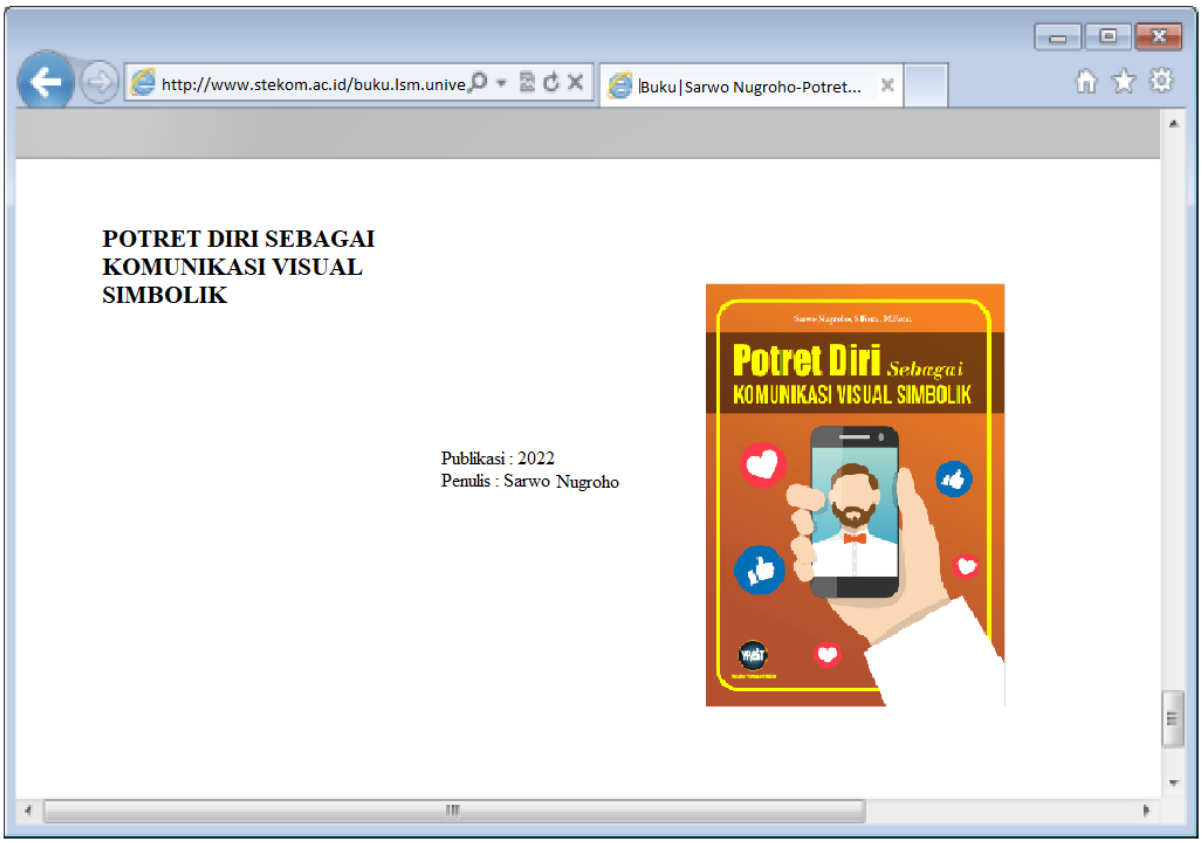
Untuk membuat tautan ini terbuka di jendela baru, kita menambahkan target="_blank" atribut/nilai pasangan ke elemen, sehingga terlihat seperti ini:

web Universitas stekom

Anda dapat mencobanya dengan membuka file link.html dari latihan sebelumnya dan menambahkan target="_blank" seperti yang ditunjukkan. Perhatikan penggunaan garis bawah sebelum kata kosong. Saat kita menyimpan file dan memuat ulang halaman itu (Ctrl+R atau Command+R), tautannya tidak akan terlihat berbeda. Namun, mengklik tautan akan membuka tab baru (atau jendela baru, tergantung pada browser dan konfigurasi Anda).

Menambahkan gambar

Gambar, seperti foto atau grafik, meningkatkan daya tarik visual halaman web. Gambar biasanya disematkan di halaman, seperti yang ditunjukkan pada contoh gambar dibawah ini, di mana foto sampul buku Sarwo Nugroho, ditampilkan.



Gambar 1.11 pada halaman web.

Dalam penyematan gambar, ada hal yang perlu di perhatikan ketika menyertakan gambar. Kita dapat menyematkan gambar appaun dalam website milik kita, dengan sayarat itu adalah hak milik sendiri, atau jka gambar tersebut memiliki hak cipta maka kita harus mendapatkan izin dari pencipta gambar tersebut sebelum menguploadnya ke website kita. Dalam pembuatan website, gambar biasanya dijadikan sebegai pelengkap, atau bahkan objek utama, ada juga yang digunakan sebagai latar belakang, ini tergantung pada kebutuhan kita dalam membuat website.

Referensi lokasi gambar

Gambar ditambahkan dengan elemen <img>. Sama seperti elemen <a>, elemen <img> menggunakan atribut untuk memberi tahu browser lebih banyak informasi tentang dirinya sendiri. Atribut src digunakan untuk memberi tahu browser di mana menemukan gambar. Sebelumnya, pada gambar dibawah atas. HTML membawa gambar tersebut ke halaman website dengan bentuk coding seperti ini:

```
<img src=”images/books_sarwonugroho.gif”>
```

Seperti yang kita lihat, elemen <img> menambahkan atribut src, yang kemudian merujuk ke lokasi gambar di server web. Elemen <img> tidak memiliki tag penutup </img> karena elemen ini tidak memiliki kontennya sendiri, berbeda dengan elemen <p>dan <a> — yang keduanya membutuhkan konten untuk masuk ke dalamnya dan oleh karena itu harus ditutup. Terkadang kita mungkin melihat elemen seperti <img> ditutup dengan /> alih-alih hanya >, seperti pada contoh. Keduanya adalah cara yang dapat diterima dan valid untuk menutup elemen jenis ini.

Elemen <img> harus selalu memiliki atribut alt. Atribut alt memberi tahu seach engine dan teknologi bantu tentang gambar yang digunakan. Saat digunakan dengan elemen <img>, atribut alt terlihat seperti ini:

```
<img src=” books_sarwonugroho” alt=”potret diri”>
```

Anda harus menggunakan deskripsi singkat sebagai isi dari atribut alt. Menggunakan sesuatu seperti "MySQL Bible adalah buku yang bagus dan semua orang harus membelinya" tidak menggambarkan gambarnya, tetapi "MySQL Bible" menggambarkannya.

Memilih gambar web yang bagus

Ketika memilih gambar untuk website, Kita perlu memastikan bahwa foto tidak blur saat foto tersebut diambil. Kita juga harus mempertimbangkan ukuran dari foto dan format foto. Web browser dapat melihat gambar berbagai format termasuk JPG, GIF, PNG dan beberapa lainnya. Ukuran file merupakan salah satu aspek terpenting untuk dipertimbangkan ketika memilih gambar. Saat menyertakan gambar dengan ukuran besar, seperti yang diambil pada pengaturan raw yang berukuran besat dengan kamera digital Anda, pengunjung harus mengunduh file tersebut terlebih dahulu sebelum dapat melihat gambar tersebut, dan tentunya ini lumayan memakan waktu, tergantung pada kecepatan koneksi pengunjung.

Untuk menyiasati hal ini, ketika menggunggah gambar kita harus mengubah ukuran gambar gambar terlebih dahulu agar gambar hanya berukuran kurang dari 100 KB. Aspek penting lainnya yang perlu dipertimbangkan adalah jumlah semua gambar pada halaman. Misalnya, dalam website kita membutuhkan sekitar 10 atau 15 gambar, jika ukurannya masing-masing adalah 100kb maka total semuanya adalah 1.5mb dan ini masih lumayan berat, saran saya akan lebih baik jika gambar yang akan digunakan dalam website memiki ukuran lebih kecil dari 100kb agar pemuatan gambar pada web yang diakses pengunjung lebih cepat.

Ketika memilih gambar, Kita juga perlu menyamakan semua format gambar agar lebih mudah diakses oleh pengunjung. Setidaknya, pastikan agar para pengunjung web kita tak perlu berpindah menggunakan aplikasi lainnya hanya untuk melihat gambar tersebut. Kenyamanan penjgunjung website kita sangat perlu untuk difikirkan. Ingatlah jumlah semua gambar saat menentukan ukuran gambar untuk halaman halaman web kita agar halaman yang bergambar tersebut lebih cepat diunduh oleh pengunjung.

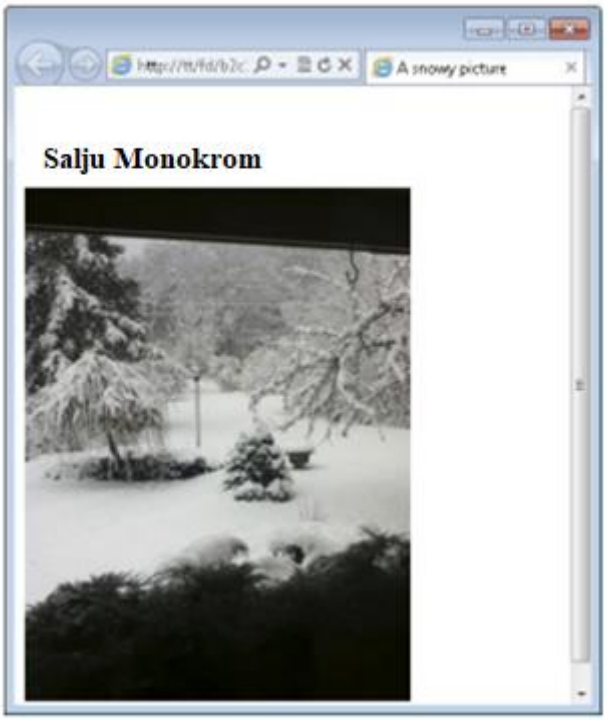
Membuat halaman dengan gambar

Saatnya membuat halaman dengan gambar sehingga kita dapat melihat bagaimana dan di mana gambar cocok dengan keseluruhan halaman HTML yang lebih besar. Ikuti langkah ini.

1. Buka Text editor
   Lihat pembahasan sebelumnya tentang text editor.
2. Masukkan HTML berikut:

   ```
   <!doctype html>
   <html>
   <head>
   <title>Gambar matahari terbit diakhir bulan agustus</title>
   </head>
   <body>
   <h1>SALJU MONOKROM</h1>
   <p><img src=”IMG01.jpg” alt=”Akhir DESEMBER 2022”></p>
   </body>
   </html>
   ```

   Saat kita membuat HTML ini, kita perlu menggunakan foto atau gambar lain milik kita sendiri atau kita dapat menggunakan file IMG01.jpg. Terlepas dari gambar yang kita pilih, kita perlu menempatkan file di root dokumen server web. Pastikan bahwa huruf besar dan kecil untuk nama file cocok dengan apa yang kita masukkan ke dalam atribut src. Dengan kata lain, jika gambar kita bernama Akhir DESEMBER, harus beratribut src harus “Akhir DESEMBER”.
3. Simpan file sebagai image.html.
   Simpan file persis seperti namanya, menggunakan huruf kecil di seluruh nama. File harus disimpan ke root dokumen Anda. Lokasi root dokumen tergantung pada bagaimana kita menginstal Apache dan pada jenis sistem yang kita gunakan. Jika kita menggunakan hostinger, maka kita hanya perlu mengunggah file ke sistem hosting tersebut.
4. Buka browser web kita untuk memuat halaman.
   Di browser web, arahkan ke http://localhost/image.html.

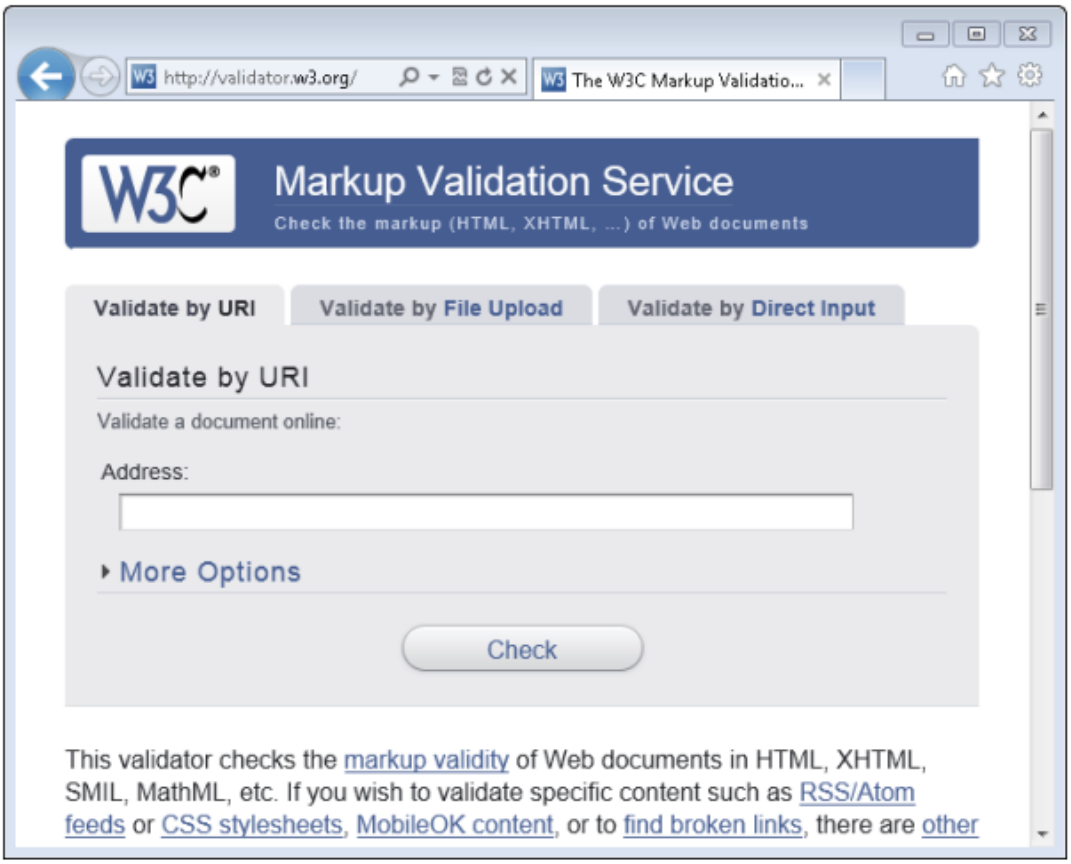


Gambar 1.12 Menambahkan gambar ke halaman.

HTML ini menggunakan elemen <img> untuk memuat foto bernama IMG01.jpg dari direktori saat ini. Dengan kata lain, IMG01.jpg berada di direktori yang sama dengan halaman image.html di server web. Hindari spasi dalam nama file gambar. Ingat juga bahwa URL, file, dan gambar peka huruf besar/kecil.

1.8 MENULIS HTML VALID

Saat kita membuat halaman web dengan HTML, ada aturan tertentu yang harus diikuti untuk memastikan bahwa browser web dapat membaca dan menampilkan halaman dengan benar. Versi saat ini dari spesifikasi HTML adalah HTML versi 5, yang hanya dikenal sebagai HTML5. Proses memvalidasi halaman berarti bahwa situs web khusus memeriksa kode HTML yang kita tulis dan bandingkan dengan spesifikasi untuk versi HTML tersebut. Situs web yang digunakan untuk memvalidasi HTML disebut *W3C Markup Validation Service* (sering disebut Validator W3C) dan dioperasikan oleh *World Wide Web Consortium* (W3C). Validator W3C dapat ditemukan di http://validator.w3.org dan gratis untuk digunakan.



Gambar 1.13 W3C

Layanan Validasi Markup, terkadang hanya disebut Validator.
Validasi HTML kita dengan salah satu dari tiga cara:

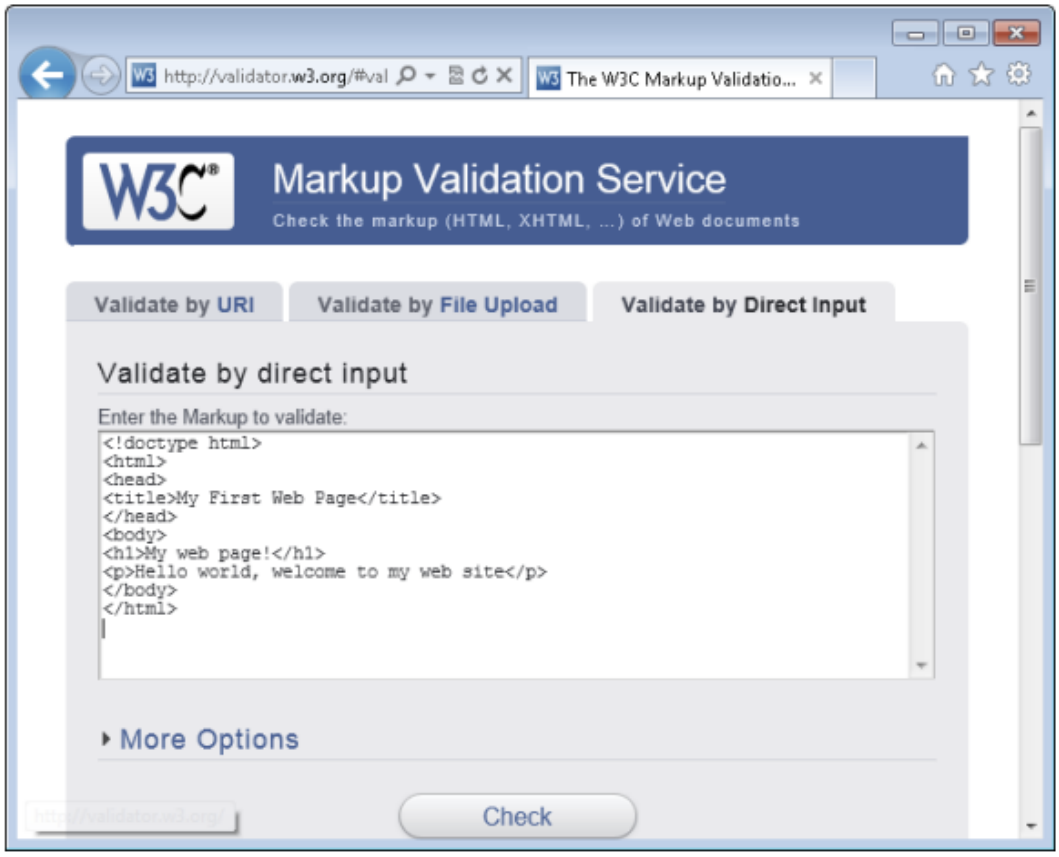
- **Menyediakan URL:** kita dapat memasukkan URL ke Validator dan secara otomatis akan mengambil HTML di URL tersebut dan mencoba memvalidasinya. Agar Validator dapat mengambil HTML kita menggunakan metode ini, halaman harus tersedia untuk umum. Ini biasanya tidak terjadi ketika kita menginstal server web di komputer Anda. Jika kita menggunakan hostinger eksternal, maka situs dan halaman kita mungkin tersedia di Internet. Dalam hal ini, kita dapat memasukkan URL di kotak alamat "Validate by URI".
- **Mengunggah file**: kita dapat mengunggah file menggunakan opsi "Validate by File Upload". Dengan menggunakan metode ini, kita memilih file di komputer Anda. File itu kemudian diunggah ke Validator.
- **Menempelkan HTML ke dalam Validator:** Ini berarti menyalin HTML dari text editor dan menempelkannya ke tab "Validate by Direct Input" di Validator. Opsi ini biasanya merupakan metode tercepat dan termudah dan itulah yang kami tunjukkan di bagian ini.

Validasi HTML

Jika kita telah mengikuti latihan di bab ini, maka kita telah membuat beberapa HTML. Latihan selanjutnya menggunakan W3C Validator untuk memastikan bahwa HTML yang kita tulis valid sesuai dengan spesifikasi HTML5. Ikuti langkah ini:

1. Buka firstpage.html menggunakan text editor
   Halaman ini adalah halaman pertama yang kita buat di bab ini. Namun, jika kita melewatkan latihan itu, buka salah satu file HTML yang kita buat di bab ini.
2. Sorot/pilih semua HTML di file yang terbuka.
   Gunakan mouse atau perangkat penunjuk untuk menyorot semua HTML atau tekan Ctrl+A di Windows atau Command+A di Mac.

3. Salin HTML ke clipboard Anda.
   Pilih Salin (ditemukan di menu Edit di sebagian besar text editor) atau tekan Ctrl+C di Windows atau Command+C di Mac untuk menyalin HTML yang disorot ke clipboard.
4. Buka browser web kita dan navigasikan ke W3C Validator.
   Dengan browser terbuka, ketik http://validator.w3.org di bilah alamat atau lokasi di browser dan tekan Enter untuk membuka Validator.
5. Pilih Validate Direct Input
   Tab Validasi dengan Input Langsung akan digunakan untuk menempelkan kode di papan klip Anda.
6. Tempelkan HTML ke dalam Validator.
   Tekan Ctrl+V di Windows atau Command+V di Mac untuk menempelkan HTML dari clipboard ke kotak Enter the Markup to Validate di halaman Validator. Jika kita menggunakan HTML dari firstpage.html, layar kita akan terlihat seperti pada Gambar dibawah ini.



Gambar 1.14 Menempelkan HTML ke dalam Validator

7. Klik Check
   Klik tombol Check pada halaman Validator untuk menjalankan validasi HTML Anda

Gambar 1.15 Dokumen HTML yang valid melewati Validator.

Perhatikan tiga peringatan dalam validasi ini. Menggulir ke bawah mengungkapkan bahwa salah satu peringatannya adalah bahwa validator HTML5 sebenarnya eksperimental saat ini, meskipun itu dapat berubah pada saat kita membaca ini. Dua peringatan lainnya terkait dengan pengaturan bahasa. Sebaiknya sertakan pengkodean karakter, yang membantu browser menentukan cara membaca dokumen, termasuk bahasa apa yang digunakan untuk HTML dan halaman. Lihat http://www.w3.org/International/tutorials/tutorial-char-enc/#Slide0250 untuk informasi lebih lanjut tentang pengkodean karakter.

BAB 2
CSS

2.1 MENAMBAHKAN STYLE DENGAN CSS

Dalam bab ini, kita mempelajari apa itu *Cascading Style Sheets* (CSS) dan bagaimana menggunakannya untuk berbagai tujuan layout dan gaya. Kami menyarankan kita untuk mengerjakan bab ini dari awal sampai akhir, karena beberapa latihan dibangun di atas latihan sebelumnya.

Menemukan Apa yang Dapat dan Tidak Dapat Dilakukan CSS untuk Halaman Web Anda

Bagian ini melihat CSS dari tingkat tinggi untuk memberi kita dasar di mana kita akan belajar cara menggunakan CSS di situs web Anda.

Apa itu CSS?

CSS melengkapi HTML dengan memberikan tampilan dan nuansa ke halaman web. Halaman HTML yang kita buat di bab sebelumnya tampak cukup sederhana, dengan font dan ukuran font default. Dengan menggunakan CSS, kita dapat membumbui tampilan itu, menambahkan warna dan gambar latar belakang, mengubah font dan ukuran font, menggambar batas di sekitar area, dan bahkan mengubah layout halaman itu sendiri CSS memiliki bahasanya sendiri, terpisah dari HTML, tetapi kita tidak akan melakukannya. t menggunakan CSS tanpa halaman HTML. Dengan kata lain, meskipun HTML dapat berdiri sendiri dan menampilkan halaman ke browser, CSS tidak bisa. kita tidak akan menulis halaman CSS. Sebaliknya, kita menulis HTML dan kemudian menggunakan CSS untuk membantu menata halaman tersebut agar terlihat seperti yang kita inginkan. Seperti HTML, CSS ditentukan oleh spesifikasi, dengan versi terbaru adalah CSS 3, yang dikenal sebagai CSS3.

Menggunakan CSS (*Cascading Style Sheets*), kita dapat menerapkan gaya ke halaman web kita agar terlihat persis seperti yang kita inginkan. Ini berfungsi karena CSS terhubung ke DOM (Document Object Model). Dengan CSS dan integrasinya dengan DOM, kita dapat dengan cepat dan mudah menata ulang elemen apa pun. Misalnya, jika kita tidak menyukai tampilan default <h1>, <h2>, dan tag heading lainnya, kita dapat menetapkan gaya baru untuk mengesampingkan pengaturan default untuk jenis dan ukuran font yang digunakan, atau apakah huruf tebal atau miring harus diatur, dan banyak lagi properti juga. Salah satu cara kita dapat menambahkan gaya ke halaman web adalah dengan menyisipkan pernyataan yang diperlukan ke kepala halaman web di antara tag <head> dan </head>. Jadi, untuk mengubah gaya tag <h1>, kita dapat menggunakan kode berikut (saya akan menjelaskan sintaksnya nanti):

```
<style>
h1 { color:red; font-size:3em; font-family:Arial; }
</style>
```

Dalam halaman HTML ini mungkin terlihat seperti Contoh 1 (lihat Gambar 2.1), yang, seperti semua contoh dalam bab ini, menggunakan deklarasi DOCTYPE HTML5 standar.

Contoh 1. Halaman HTML sederhana

```
<!DOCTYPE html>
<html>
<head>
```

```
<title>Hello World</title>
<style>
h1 { color:red; font-size:3em; font-family:Arial; }
</style>
</head>
<body>
<h1>Hello there</h1>
</body>
</html>
```

Gambar 2.1 Menata tag, dengan gaya asli yang ditampilkan di sisipan

Mengapa Menggunakan CSS?

Sebelum adanya CSS, seorang developer HTML mengubah font dan warna dengan cara mengubah atribut pada setiap elemen. Jika developer HTML ingin semua judul terlihat dengan cara tertentu, maka ia harus mengubah setiap judul tersebut, dan, bayangkan saja untuk melakukan ini pada lebih dari 10 halaman atau bhakan 50 halaman, sangat melelahkan, membosankan dan memakan waktu bukan. CSS meringankan beban pembaruan elemen satu per satu ini dan membuatnya sehingga kita dapat menerapkan satu gaya tunggal di satu atau beberapa elemen. Kita dapat menerapkan beberapa gaya ke elemen yang sama, dan kita dapat menargetkan gaya tertentu hingga ke elemen individual. Misalnya, jika ingin semua judul menjadi font tebal tetapi judul tertentu harus miring, kita dapat melakukannya dengan CSS. Gunakan CSS untuk membuat perubahan pada layout, tampilan, dan nuansa halaman web. CSS mempermudah pengelolaan perubahan ini.

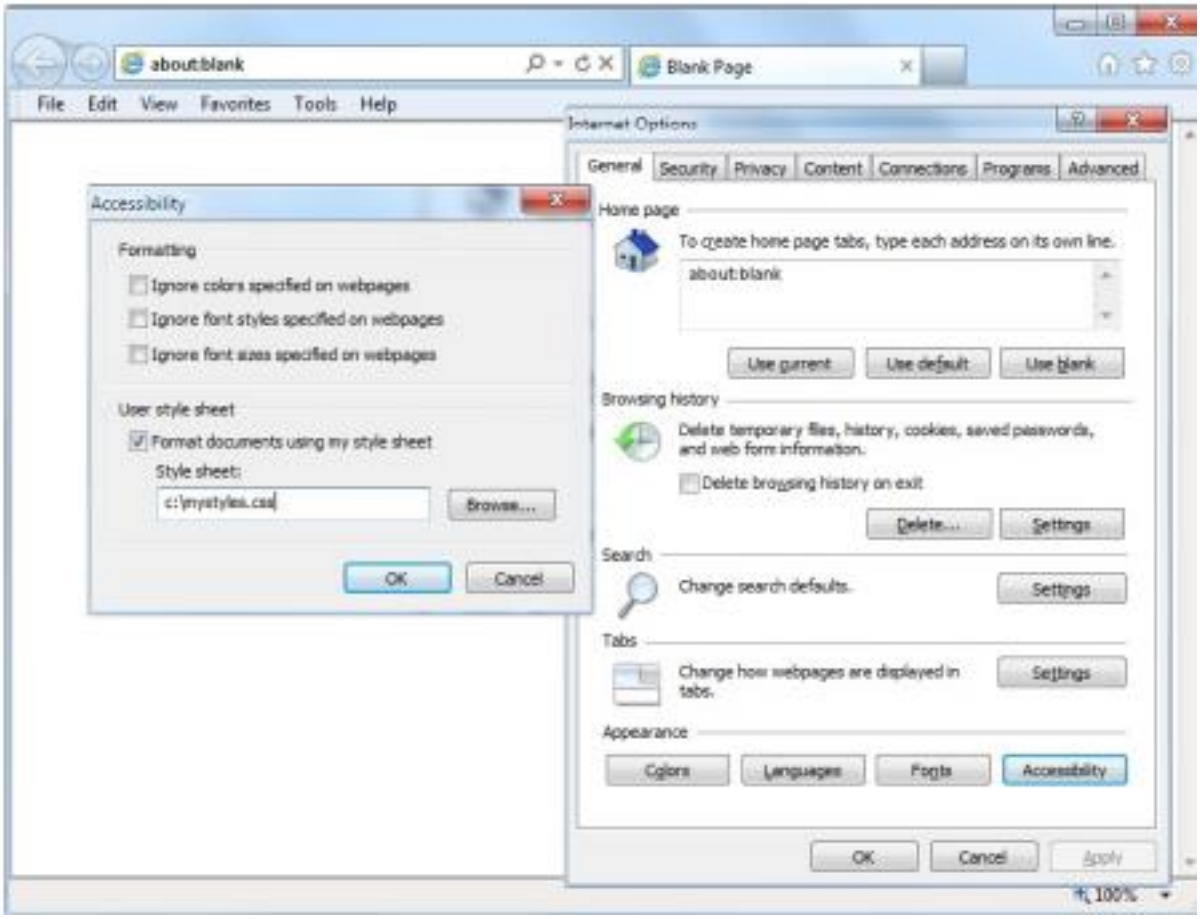
2.2 KETERBATASAN CSS

CSS memiliki batasan, dan keterbatasan utama CSS adalah tidak semua browser web mendukung CSS dengan cara yang persis sama. Satu browser mungkin menafsirkan layout dengan cara yang sedikit berbeda, menempatkan item lebih tinggi atau lebih rendah atau di tempat yang sama sekali berbeda. Selain itu, browser lama tidak mendukung versi CSS yang lebih baru, khususnya spesifikasi CSS3. Ini berarti bahwa browser tersebut tidak dapat menggunakan beberapa fitur dari spesifikasi CSS3. Untuk menyiasatinya, kita dapat menggunakan versi lama dari spesifikasi yang lebih banyak didukung oleh browser lama tersebut.

Kuncinya adalah menguji di beberapa browser. Browser web seperti Firefox, Chrome, dan Safari semuanya dapat diunduh gratis, dan Microsoft menawarkan perangkat lunak yang disebut PC Virtual untuk Kompatibilitas Aplikasi, yang merupakan versi Windows gratis, terbatas waktu, yang menyertakan versi Internet Explorer yang lebih lama. kita dapat menjalankannya di dalam perangkat lunak emulasi Virtual PC gratis Microsoft. Dengan menguji di browser lain, kita dapat melihat bagaimana situs akan terlihat di browser tersebut dan memperbaiki masalah layout sebelum menyebarkan situs ke Internet.

2.3 TIPE GAYA

Ada sejumlah jenis gaya yang berbeda, mulai dari gaya default yang diatur oleh browser (dan user style apa pun yang mungkin telah diterapkan di browser untuk mengganti defaultnya), melalui gaya sebaris atau yang disematkan, hingga style sheet eksternal. Gaya yang didefinisikan di setiap jenis memiliki hierarki prioritas, dari rendah hingga tinggi.



Gambar 2.2 Menerapkan user style ke Internet Explorer

Gaya Default

Tingkat prioritas gaya terendah adalah gaya default yang diterapkan oleh browser web. Gaya ini dibuat sebagai fallback ketika halaman web tidak memiliki gaya apa pun, dan gaya ini dimaksudkan sebagai kumpulan gaya umum yang akan ditampilkan dengan cukup baik di sebagian besar kasus. Pra-CSS, ini adalah satu-satunya gaya yang diterapkan pada dokumen, dan hanya sedikit dari mereka yang dapat diubah oleh halaman web (seperti tampilan font, warna, dan ukuran, dan beberapa argumen ukuran elemen).

User style

Ini adalah prioritas gaya tertinggi berikutnya, dan didukung oleh sebagian besar browser modern tetapi diterapkan secara berbeda oleh masing-masing browser. Jika kita ingin mempelajari cara membuat gaya default kita sendiri untuk menjelajah, gunakan mesin pencari untuk memasukkan nama browser kita diikuti dengan "user style" (mis., "User style Firefox" atau "User style Opera") untuk mengetahui caranya . Gambar 2.3 menunjukkan style sheet pengguna yang diterapkan ke Microsoft Internet Explorer. Jika user style ditetapkan yang telah ditetapkan sebagai default browser, maka pengaturan default browser akan ditimpa. Gaya apa pun yang tidak ditentukan dalam style sheet pengguna akan mempertahankan nilai defaultnya seperti yang diatur di browser.

2.4 MENYAMBUNGKAN CSS KE HALAMAN

Kita dapat menambahkan CSS ke halaman dengan beberapa cara berbeda:

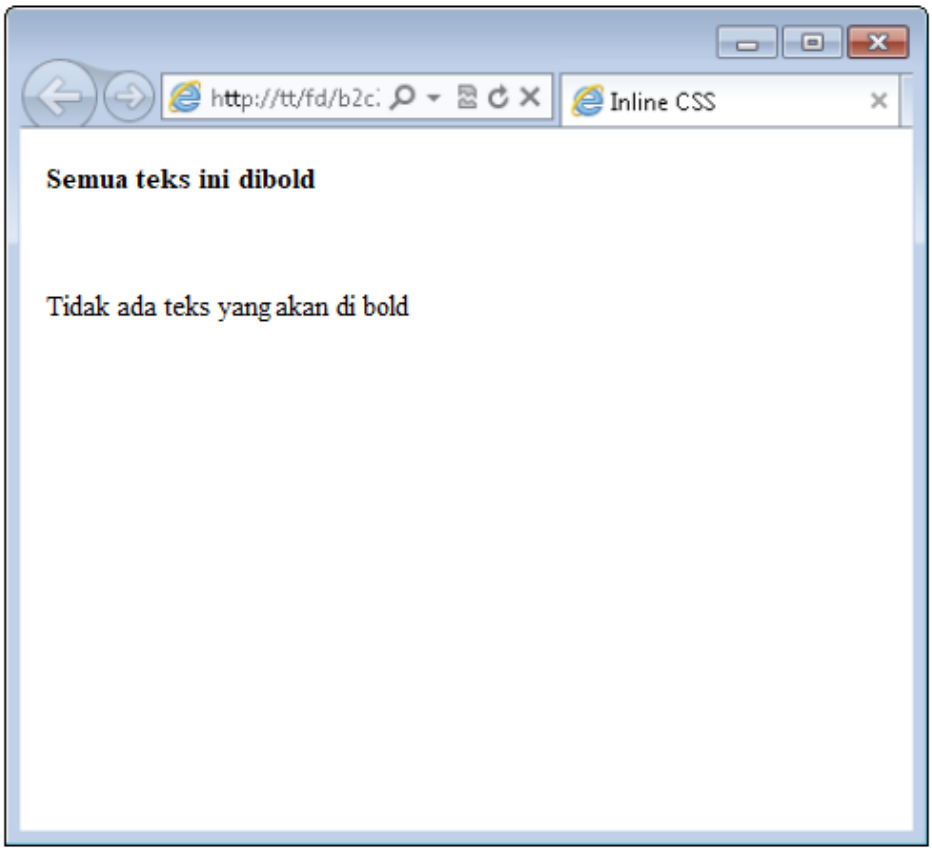
- Langsung ke elemen HTML
- Dengan style sheet internal
- Dengan style sheet eksternal

Cara yang paling dapat digunakan kembali untuk menambahkan CSS ke halaman adalah dengan menggunakan style sheet eksternal, tetapi yang paling sederhana adalah menambahkan informasi gaya secara langsung pada sebuah elemen.

Menambahkan style ke elemen HTML

Menambahkan gaya ke hampir semua elemen HTML dengan atribut gaya, seperti dalam contoh ini yang membuat semua teks di paragraf pertama menjadi font tebal:

```
<p style=”font-weight: bold;”>All of this text will be bold.</p>
```



Gambar 2.3 Teks tebal diatur dengan CSS

Menghitung Spesifisitas

Kami menghitung kekhususan aturan dengan membuat nomor tiga bagian berdasarkan jenis selektor dalam daftar bernomor di bagian sebelumnya. Bilangan majemuk ini mulai terlihat seperti [0,0,0]. Saat memproses aturan, setiap selektor yang mereferensikan ID menambah angka pertama sebanyak 1, sehingga angka majemuk menjadi [1,0,0]. Mari kita lihat aturan berikut, yang memiliki tujuh referensi, dengan tiga di antaranya ke ID #heading, #main, dan #menu. Sehingga bilangan majemuk menjadi [3,0,0].

```
#heading #main #menu .text .quote p span {
// Rules go here;
}
```

Kemudian jumlah kelas dalam selektor ditempatkan di bagian kedua dari bilangan majemuk. Dalam contoh ini ada dua (.text dan .quote), sehingga bilangan majemuk menjadi [3,2,0]. Akhirnya, semua selektor yang tag elemen referensi dihitung, dan nomor ini ditempatkan di bagian terakhir dari nomor gabungan. Dalam contoh, ada dua (p dan span), sehingga bilangan majemuk akhir menjadi [3,2,2], itu saja yang diperlukan untuk membandingkan kekhususan aturan ini dengan yang lain, seperti berikut ini:

```
#heading #main .text .quote .news p span {
// Rules go here;
}
```

Di sini, meskipun tujuh elemen juga direferensikan, sekarang hanya ada dua referensi ID, tetapi ada tiga referensi kelas, yang menghasilkan bilangan majemuk [2,3,2]. Karena 322 lebih besar dari 232, contoh pertama lebih diutamakan daripada yang terakhir.

Dalam kasus di mana ada sembilan atau kurang dari setiap jenis dalam bilangan majemuk, Kita dapat mengonversinya langsung ke bilangan desimal, yang dalam hal ini adalah 352. Aturan dengan angka lebih rendah dari ini akan memiliki prioritas lebih rendah, dan aturan dengan angka lebih tinggi nomor akan lebih diutamakan. Di mana dua aturan berbagi nilai yang sama, aturan yang paling baru diterapkan menang

Style sheet Eksternal

Jenis gaya berikutnya adalah yang ditetapkan dalam style sheet eksternal. Pengaturan ini akan menimpa semua yang ditetapkan baik oleh pengguna atau oleh browser. Style sheet eksternal adalah cara yang disarankan untuk membuat gaya karena Kita dapat menghasilkan style sheet yang berbeda untuk tujuan yang berbeda seperti penataan gaya untuk penggunaan web umum, untuk dilihat pada browser seluler dengan layar yang lebih kecil, untuk tujuan pencetakan, dan sebagainya. Cukup terapkan yang dibutuhkan untuk setiap jenis media saat membuat halaman web.

Style Internal

Lalu ada gaya internal, yang kita buat dalam tag <style> ... </style>, dan yang lebih diutamakan daripada semua jenis gaya sebelumnya. Namun, pada titik ini, kita harus mulai memecah pemisahan antara gaya dan konten, karena setiap style sheet eksternal yang dimuat pada saat yang sama akan memiliki prioritas yang lebih rendah.

Stykle Inline

Terakhir, gaya sebaris adalah tempat Kita menetapkan properti langsung ke elemen. Mereka memiliki prioritas tertinggi dari semua jenis gaya, dan digunakan seperti ini:

```
<a href="http://google.com" style="color:green;">Visit Google</a>
```

Dalam contoh ini, tautan yang ditentukan akan ditampilkan dalam warna hijau, terlepas dari pengaturan warna default atau lainnya yang diterapkan oleh jenis style sheet lainnya, baik secara langsung ke tautan ini atau secara umum untuk semua tautan.

2.5 CASCADE CSS

Salah satu hal yang paling mendasar tentang properti CSS adalah bahwa mereka mengalir, itulah sebabnya mereka disebut Cascading Style Sheets. Tapi apa artinya ini? Cascading adalah metode yang digunakan untuk menyelesaikan potensi konflik antara berbagai jenis style sheet yang didukung browser, dan menerapkannya dalam urutan prioritas menurut siapa yang membuatnya, metode yang digunakan untuk membuat gaya, dan jenis properti yang dipilih.

Creator Style Sheet

Ada tiga jenis utama style sheet yang didukung oleh semua browser modern. Dalam urutan prioritas dari tinggi ke rendah, mereka adalah:

1. dibuat oleh penulis dokumen
2. dibuat oleh pengguna
3. dibuat oleh browser

Ketiga set style sheet ini diproses dalam urutan terbalik. Pertama, default di browser web diterapkan ke dokumen. Tanpa default ini, halaman web yang tidak menggunakan style sheet akan terlihat buruk. Mereka termasuk wajah font, ukuran, dan warna; jarak elemen; batas dan spasi tabel; dan semua standar wajar lainnya yang diharapkan pengguna. Selanjutnya, jika pengguna telah membuat gaya apa pun untuk digunakan alih-alih gaya standar, gaya ini diterapkan, menggantikan gaya default browser apa pun yang mungkin bertentangan. Terakhir, gaya apa pun yang dibuat oleh penulis dokumen saat ini kemudian diterapkan, menggantikan gaya apa pun yang telah dibuat baik sebagai default browser atau oleh pengguna.

Metode Style sheet

Style sheet dapat dibuat melalui tiga metode berbeda. Dalam urutan prioritas dari tinggi ke rendah, mereka adalah:

1. Sebagai gaya sebaris
2. Dalam style sheet yang disematkan
3. Sebagai style sheet eksternal

Sekali lagi, metode pembuatan style sheet ini diterapkan dalam urutan prioritas yang terbalik. Oleh karena itu, semua style sheet eksternal diproses terlebih dahulu, dan gayanya diterapkan ke dokumen. Selanjutnya, setiap gaya yang disematkan (dalam tag <style> ... </style>) diproses, dan semua yang bertentangan dengan aturan eksternal akan diprioritaskan dan akan menimpanya. Terakhir, gaya apa pun yang diterapkan langsung ke elemen sebagai gaya sebaris (seperti ...) diberi prioritas tertinggi, dan menimpa semua properti yang ditetapkan sebelumnya.

Mengimpor Style Sheet

Saat kita ingin menata seluruh situs, daripada satu halaman, cara yang lebih baik untuk mengelola style sheet adalah dengan memindahkannya sepenuhnya dari halaman web ke file yang terpisah, lalu mengimpor yang kita butuhkan. Ini memungkinkan menerapkan style sheet yang berbeda untuk layout yang berbeda (seperti web dan cetak), tanpa mengubah HTML. Ada beberapa cara berbeda untuk mencapai ini, bagian pertama adalah dengan menggunakan arahan CSS @import seperti ini:

```
<style>
@import url('styles.css');
</style>
```

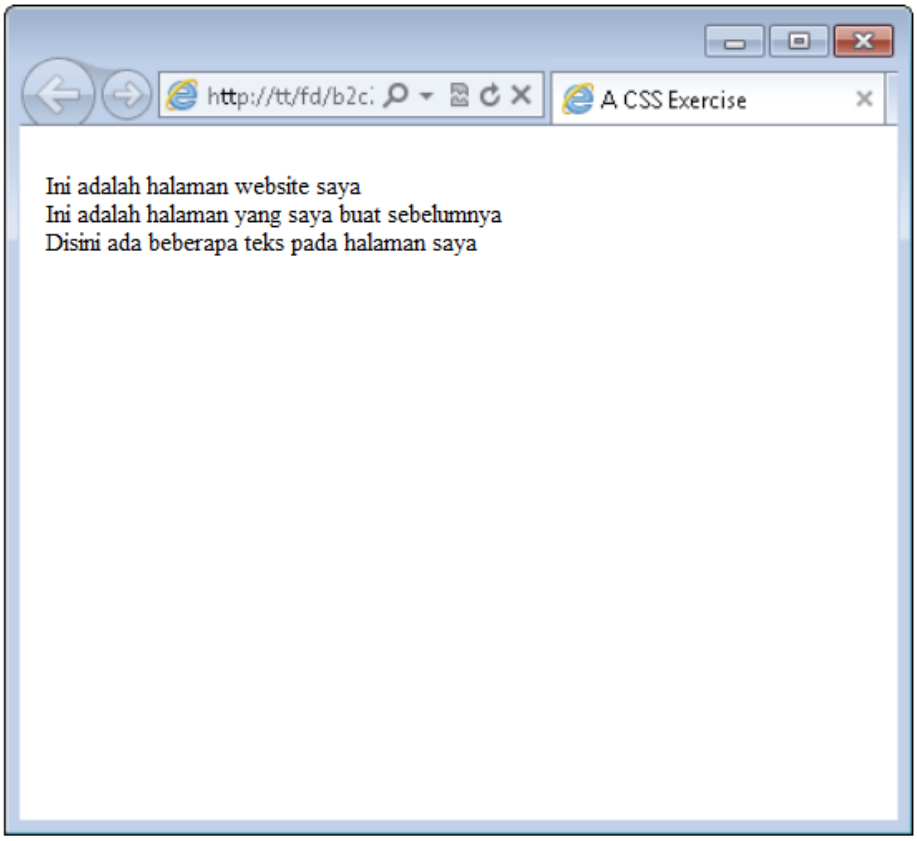
Pernyataan ini memberitahu browser untuk mengambil style sheet dengan nama styles.css. Perintah @import cukup fleksibel karena kita dapat membuat style sheet yang menarik daripada style sheet lain, dan seterusnya. kita hanya perlu memastikan bahwa tidak ada tag <style> atau </style> di salah satu style sheet eksternal Anda, atau mereka tidak akan berfungsi.

Berikut contoh yang bisa kita coba. kita membuat beberapa HTML terlebih dahulu dan kemudian mulai menambahkan gaya ke dalamnya.

1. Buka text editor
2. Di dalam dokumen teks kosong, tempatkan HTML berikut:

```
<!doctype html>
<html>
<head>
<title>A CSS Exercise</title>
</head>
<body>
<div>This is my web page.</div>
<div>
 This is the <span>nicest</span> page I've made yet.
</div>
<div>Here is some text on my page.</div>
</body>
</html>
```

3. Simpan file sebagai css.html.
   Di dalam text editor, simpan file menggunakan nama css.html, pastikan tidak ada spasi atau karakter lain dalam nama file. File harus disimpan di dalam root dokumen Anda.
4. Buka browser web kita dan lihat halamannya.
   Di dalam bilah alamat browser web, ketik http://localhost/css.htmldan kita akan melihat halaman yang mirip dengan yang ditunjukkan pada gambar berikut.

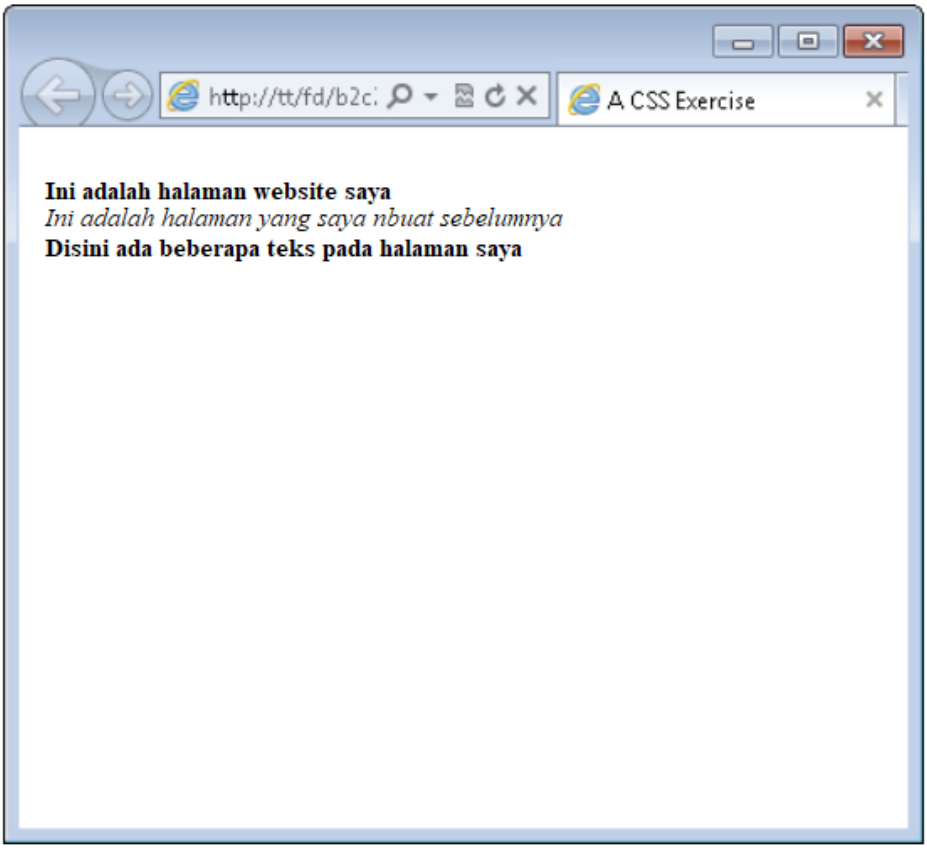


Gambar 2.4 Membuat halaman web sederhana.

5. Tutup browser.
   Sekarang setelah kita memverifikasi bahwa halaman tersebut berfungsi, tutup browser.
6. Beralih ke text editor untuk mengedit HTML.
   Di dalam text editor, edit HTML dari Langkah 2 untuk menambahkan CSS. Jika kita menutup file, buka kembali di text editor.
7. Ubah HTML untuk menambahkan dua atribut gaya yang berbeda, seperti yang ditunjukkan di sini:

```
<!doctype html>
<html>
<head>
<title>A CSS Exercise</title>
</head>
<body>
<div style="font-weight: bold;"> Ini adalah halaman website saya.</
div>
<div>
Ini adalah <span style="font-style:
italic;">nicest</span> halaman yang saya buat sebelumnya.
</div>
<div style="font-weight: bold;"> Disini ada beberapa teks pada
halaman saya.</div>
</body>
</html>
```

8. Simpan file.
   Anda dapat menyimpannya sebagai css.html atau menyimpannya sebagai css2.html jika kita tidak ingin menimpa file css.html asli Anda. File harus disimpan di root dokumen Anda.
9. Buka browser web kita dan lihat halamannya.
   Mengetik di http://localhost/css.html (atau css2.html jika kita menyimpannya sebagai css2.html) mengungkapkan file, sekarang dengan gaya sebaris diterapkan ke dua area. Ini diilustrasikan seperti gambar berikut ini.



Gambar 2.5 Menambahkan gaya sebaris ke HTML.

Latihan ini membuat file HTML yang menggunakan elemen dan . HTML kemudian ditata menggunakan gaya sebaris. Gaya sebaris menyesuaikan bobot font dan gaya font untuk membuat teks tebal untuk dua elemen dan teks miring untuk satu elemen. Ketika digunakan dengan CSS, font-weight dan font-style dikenal sebagai properti. Properti ini kemudian diberi nilai, seperti tebal dan miring. Saat kita melihat terminologi bahwa properti CSS telah diubah, kita tahu bahwa properti itu adalah nama dan nilainya adalah untuk mengubah properti itu.

Selektor Style sheet

Ada tiga cara berbeda untuk memilih elemen yang akan ditata. Mulai dari urutan prioritas tertinggi ke terendah, mereka adalah:

- Referensi oleh ID individu atau selektor atribut
- Referensi dalam kelompok berdasarkan kelas
- Referensi oleh tag elemen (seperti <p> atau <b>)

Selektor diproses sesuai dengan jumlah dan jenis elemen yang dipengaruhi oleh aturan, yang sedikit berbeda dari dua metode sebelumnya untuk menyelesaikan konflik. Ini karena aturan tidak harus berlaku hanya untuk satu jenis selektor pada satu waktu, dan mungkin merujuk ke

banyak selektor yang berbeda. Oleh karena itu, diperlukan suatu metode untuk menentukan prioritas aturan yang dapat memuat kombinasi selektor apa saja. Ini dilakukan dengan menghitung kekhususan setiap aturan dengan mengurutkannya dari lingkup tindakan terluas hingga tersempit.

Menggunakan style sheet internal

Menerapkan gaya ke elemen individual dengan cepat akan menjadi rumit ketika kita memiliki halaman web yang besar. Seperti yang kita lihat di latihan sebelumnya, untuk membuat teks dari dua elemen menjadi tebal, kita perlu menambahkan atribut gaya ke setiap elemen . Untungnya, ada cara yang lebih baik. kita dapat membuat area khusus halaman web untuk menyimpan informasi gaya. Informasi gaya ini kemudian diterapkan ke elemen yang sesuai dalam HTML. Ini mengurangi kebutuhan untuk menambahkan atribut gaya ke setiap elemen. kita menambahkan gaya internal di dalam bagian <head> halaman web menggunakan elemen <style>. Daftar dibawah ini menunjukkan HTML dengan elemen <style>.

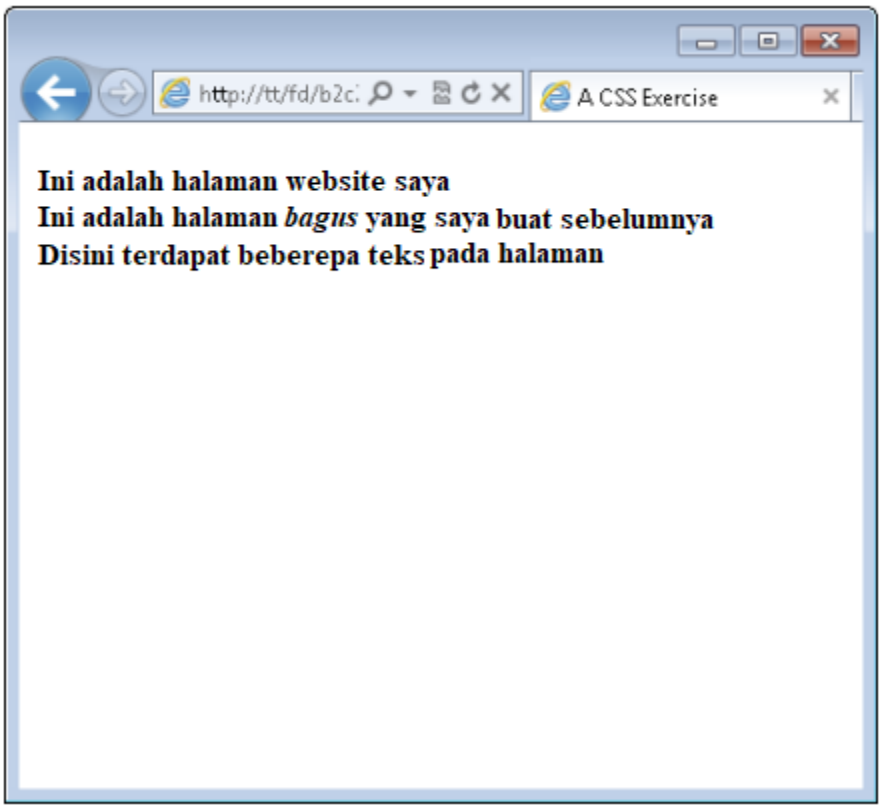
Contoh 2 Menggunakan Style sheet Internal

```
<!doctype html>
<html>
<head>
 <title>A CSS Exercise</title>
 <style type=”text/css”>
 div {
 font-weight: bold;
 }
 span {
 font-style: italic;
 }
 </style>
</head>
<body>
<div>Ini adalah halaman website saya.</div>
<div>
Ini adalah <span>nicest</span> halaman yang saya buat sebelumnya.
</div>
<div>Disini ada beberapa teks pada halaman saya.</div>
</body>
</html>
```

Halaman menambahkan style sheet internal untuk menambahkan font tebal ke elemen dan font gaya miring ke semua elemen di halaman.

```
<style type=”text/css”>
 div {
 font-weight: bold;
 }
 span {
 font-style: italic;
 }
 </style>
```

Elemen <style> menggunakan atribut type untuk memberi tahu browser jenis informasi gaya apa yang diharapkan. Dalam hal ini, kami menggunakan gaya tipe teks/css. Perhatikan juga tag penutup, yang diperlukan.



Gambar 2.6 Menggunakan style sheet internal

Perhatikan baik-baik diatas; terdapat sedikit perbedaan dengan tampilan dari gambar sebelumnya, bagian baris kedua dari gambar sebelumnya tidak dicetak tebal, tetapi baris muncul dalam huruf tebal pada gambar selanjutnya. Perbedaan ini muncul karena gaya internal sheet menargetkan semua elemen di halaman, bukan hanya yang spesifik yang diubah dengan metode gaya sebaris yang ditunjukkan sebelumnya.

Menggunakan style sheet eksternal

Sampai sini kita telah melihat bagaimana gaya sebaris menambahkan informasi gaya ke setiap elemen satu per satu, ini bisa menjadi membosankan. Dilanjutkan dengan cara menggunakan style sheet internal untuk membuat gaya informasi untuk halaman secara keseluruhan. Browser membaca style sheet eksternal ini seperti halnya membaca gaya yang diterapkan di dalam halaman itu sendiri, dan menerapkan gaya tersebut sesuai dengan itu. kita menambahkan atau menyertakan style sheet eksternal dengan elemen <link>, yang berada di area <head> sebuah halaman HTML.

Elemen <link> khas untuk menambahkan CSS terlihat seperti ini:

```
<link rel=”stylesheet” type=”text/css” href=”style.css”>
```

Baris itu menyertakan file bernama style.css di direktori saat ini dan memasukkannya ke dalam halaman. Semua informasi <style> dan gaya sebaris dapat dihapus menggantikan satu baris di bagian <head> halaman. Di dalam style sheet eksternal terdapat aturan yang harus diterapkan — dan hanya aturan yang harus diterapkan. kita tidak perlu menyertakan atribut gaya atau bahkan elemen pembuka atau penutup <style> dalam style sheet eksternal.

```
div {
 font-weight: bold;
 }
 span {
 font-style: italic;
 }
```

Sekarang style sheet eksternal dapat dibagikan di antara beberapa file HTML. Jika kita perlu membuat perubahan gaya, kita hanya perlu mengedit satu file CSS, dan itu secara otomatis menerapkan gaya ke semua halaman yang menggunakan file CSS itu. Seperti yang kita lihat, file CSS eksternal membuat pemeliharaan situs web menjadi lebih mudah. Style sheet eksternal adalah metode yang disarankan untuk menggunakan CSS, dan dengan hanya beberapa pengecualian, sisa buku ini menggunakan CSS yang disertakan dari style sheet eksternal.

Penargetan Style

Ingat masalah yang diidentifikasi sebelumnya, di mana font tebal diterapkan ke semua elemen pada halaman, ketika kita mungkin tidak ingin menerapkannya ke semua elemen tersebut. kita dapat memperbaiki masalah tersebut dengan menargetkan atau mempersempit cakupan aturan CSS menggunakan selector yang lebih spesifik. CSS menggunakan selector untuk menentukan elemen atau menerapkan aturan. Dalam contoh style sheet internal sebelumnya selector adalah elemen , atau semua elemen pada halaman.

2.6 MENGIMPOR CSS DARI DALAM HTML

Anda juga dapat menyertakan style sheet dengan tag <link> HTML seperti ini:

```
<link rel='stylesheet' type='text/css' href='styles.css'>
```

Ini memiliki efek yang sama persis dengan direktif @import, kecuali bahwa <link> adalah tag khusus HTML dan bukan direktif gaya yang valid, sehingga tidak dapat digunakan dari dalam satu style sheet untuk menarik yang lain, dan juga tidak dapat ditempatkan dalam sepasang tag <style> ... </style>. Sama seperti kita dapat menggunakan beberapa @import directives dalam CSS kita untuk menyertakan beberapa style sheet eksternal, kita juga dapat menggunakan elemen <link> sebanyak yang kita suka dalam HTML Anda.

Pengaturan Gaya Tertanam

Juga tidak ada yang menghentikan kita untuk mengatur atau mengganti gaya tertentu secara individual untuk halaman saat ini berdasarkan kasus per kasus dengan menyisipkan deklarasi gaya langsung di dalam HTML, seperti ini (yang menghasilkan teks biru miring di dalam tag):

```
<div style='font-style:italic; color:blue;'>Hello there</div>
```

Tapi ini harus dicadangkan hanya untuk keadaan yang paling luar biasa, karena melanggar pemisahan konten dan presentasi.

Memilih elemen HTML

Hampir semua elemen HTML dapat menjadi target selector, bahkan hal-hal seperti elemen <body>. Faktanya, elemen <body> sering digunakan sebagai selector untuk

menargetkan gaya di seluruh halaman, seperti kumpulan font apa yang akan digunakan untuk halaman. kita melihat contohnya di bagian berikutnya, "Changing font." kita telah melihat contoh menggunakan elemen HTML sebagai penyeleksi. kita cukup menggunakan nama elemen, tanpa tanda kurung di sekitarnya. Alih-alih seperti dalam HTML, kita menggunakan div saat menggunakannya sebagai selector CSS. Berikut tampilannya:

```
div {
 font-berat: tebal;
 }
```

Seperti yang kita lihat, nama elemen, div, diikuti oleh kurung kurawal. Ini menunjukkan bahwa aturan dimulai. Di dalam kurung kurawal pembuka dan penutup yang sesuai, properti, font-weight, dipilih, diikuti oleh titik dua (:). Nilai tersebut kemudian disetel menjadi tebal. Baris diakhiri dengan titik koma (;). Titik koma ini memberi tahu browser bahwa baris sudah selesai; dengan kata lain, pasangan properti/nilai ditutup. Beberapa properti dapat diatur dalam selector yang sama. Mengambil contoh sebelumnya, kita dapat mengubah gaya font menjadi tebal dan miring, seperti ini:

```
div {
 font-weight: bold;
 font-style: italic;
 }
```

Setiap baris diakhiri dengan titik koma, dan seluruh aturan diapit oleh kurung kurawal buka dan tutup.

Memilih elemen individu

Id (singkatan dari identifier) memungkinkan kita untuk memilih satu dan hanya satu elemen dalam sebuah halaman. Untuk melakukannya, kita perlu memodifikasi HTML untuk menambahkan atribut id dan memberikan nama untuk elemen tersebut. Misalnya, pertimbangkan HTML seperti ini:

```
<div>Agus Widodo</div>
```

Jika ingin menerapkan font tebal ke elemen itu, kita bisa memilih semua elemen tetapi itu juga akan menerapkan font tebal ke elemen lain di halaman. Sebagai gantinya, solusinya adalah menambahkan id ke tertentu, seperti:

```
<div id=”myName”> Agus Widodo</div>
```

Nilai id disetel ke myName. Perhatikan kasus yang digunakan dalam contoh ini, dengan huruf besar N. Huruf besar ini harus dicocokkan dengan CSS. Untuk memilih id ini di dalam CSS, kita menggunakan karakter hash (#), seperti:

```
#myName
```

Dengan mengingat hal itu, membuat #myName id tebal terlihat persis seperti contoh yang telah kita lihat, cukup ganti #myName untuk div:

```
#myName {
```

```
font-weight: bold;
}
```

Selalu cocokkan case yang kita gunakan di HTML dengan case yang kita gunakan di CSS. Jika kita menggunakan huruf besar semua dalam memberi nama ID di HTML, maka gunakan huruf besar semua di CSS. Jika kita menggunakan semua huruf kecil di HTML, gunakan huruf kecil di CSS. Jika kita menggunakan kombinasi, seperti contoh, maka cocokkan kombinasi tersebut di CSS. Saat menggunakan ID dalam HTML, penting untuk menyadari bahwa ID harus digunakan sekali di seluruh halaman. Tak masalah menggunakan ID yang sama di halaman yang berbeda, tetapi ID tersebut seharusnya hanya muncul sekali dalam satu halaman.

Solusi yang lebih baik untuk menyetel gaya elemen adalah dengan menetapkan ID ke dalamnya dalam HTML, seperti ini:

```
<div id='welcome'>Hello there</div>
```

Ini menyatakan bahwa konten dengan sambutan ID harus menerapkan gaya yang ditentukan dalam pengaturan gaya sambutan. Pernyataan CSS yang cocok untuk ini mungkin terlihat seperti berikut:

```
#welcome { font-style:italic; color:blue; }
```

Catatan: Perhatikan penggunaan simbol #, yang menyatakan bahwa hanya ID dengan nama selamat datang yang harus diberi gaya dengan pernyataan ini.

Memilih sekelompok elemen

Kita telah mempelajari cara menargetkan elemen HTML di seluruh halaman dan cara menargetkan hanya satu elemen individual. Kelas CSS digunakan untuk memilih beberapa elemen dan menerapkan gaya ke dalamnya. Berbeda dengan selectoran yang terjadi saat kita memilih semua elemen , kelas CSS hanya diterapkan pada elemen tertentu yang kita pilih. Elemen HTML bahkan tidak perlu dari jenis yang sama; kita dapat menerapkan kelas CSS yang sama ke , ke tag <img>, dan ke elemen <p> sama.

2.7 COMMENT CSS

Di dalam aturan CSS yang ditampilkan di dekatnya, ada komentar: /* CSS Goes Here */. Sama seperti di HTML di mana kita dapat menggunakan komentar untuk membantu menjelaskan bagian kode tertentu, demikian juga kita dapat menggunakan komentar di CSS untuk membantu menjelaskan CSS. Seperti komentar HTML, komentar dalam CSS tidak terlihat di output halaman tetapi, juga seperti komentar HTML, komentar CSS dapat dilihat dengan melihat sumber dari dokumen HTML atau CSS itu sendiri. Ini berarti pengunjung juga dapat melihatcomment.

Komentar dalam CSS dibuka dengan /* dan ditutup dengan */. Segala sesuatu yang muncul di antara /* dan */ diperlakukan sebagai komentar.

Seperti id, kelas diterapkan terlebih dahulu ke elemen HTML dengan atribut. Atributnya adalah kelas dengan judul yang tepat, seperti dalam contoh ini:

```
<div class=”boldText”>This text has a class.<div>
```

Seperti pada contoh id, kelas juga peka huruf besar/kecil. Kasing yang digunakan dalam HTML harus cocok dengan yang ada di CSS. Sedangkan selector ID menggunakan tanda pound (#) di

CSS, kelas menggunakan satu titik atau titik. Dalam contoh sebelumnya, di mana kelas diberi nama boldText di HTML, itu akan direferensikan seperti ini di CSS:

```
.boldText {
/* CSS Goes Here */
}
```

Dalam contoh ini, kelas boldText dipilih. Kelas dapat digunakan untuk memecahkan masalah yang ditemukan sebelumnya (di bagian "Menggunakan style sheet internal"), di mana font tebal diterapkan ke semua elemen karena CSS menggunakan selector div. kita dapat menggunakan kelas dalam HTML untuk menargetkan hanya elemen-elemen yang ingin kita targetkan.

Merupakan ide yang baik untuk mengomentari aturan CSS Anda, bahkan jika kita hanya menjelaskan kelompok utama pernyataan daripada semua atau sebagian besar dari mereka. kita dapat melakukan ini dengan dua cara berbeda. Pertama, kita dapat menempatkan komentar di dalam sepasang tag /* ... */, seperti ini:

```
/* This is a CSS comment */
```

Atau kita dapat memperpanjang komentar melalui banyak baris, seperti ini:

```
/*
A Multi
line
comment
*/
```

Saatnya menguji teori itu. Ikuti langkah ini.

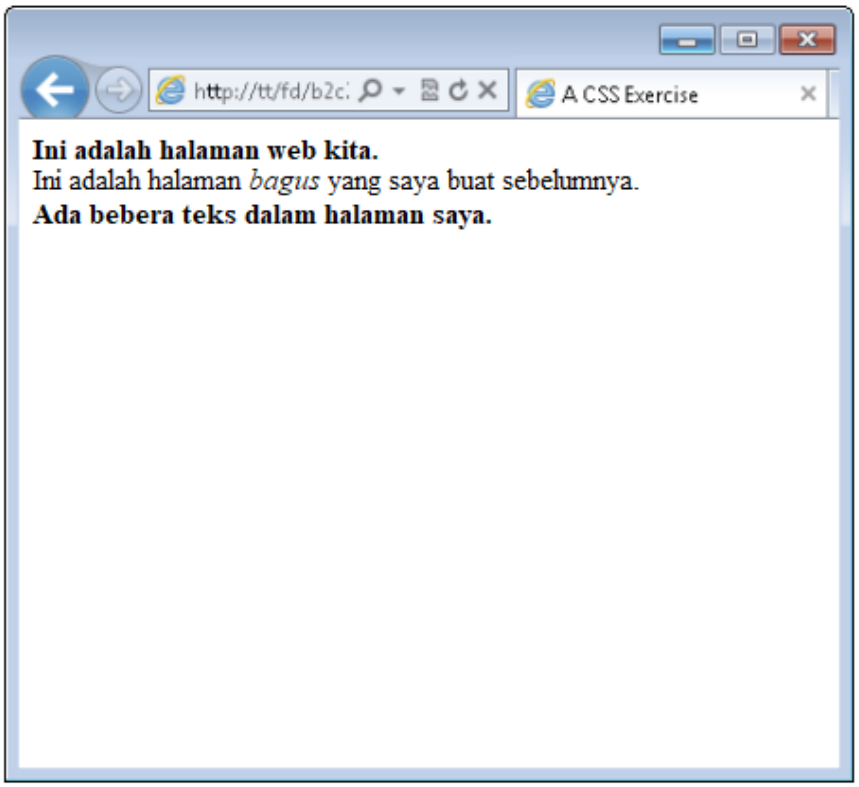
1. Buka text editor
2. Buka css.html
3. Ubah css.html untuk mengubah CSS

```
<!doctype html>
<html>
<head>
<title>A CSS Exercise</title>
<link rel=”stylesheet” type=”text/css” href=”style.
css”>
</head>
<body>
<div class=”boldText”>This is my web page.</div>
<div>
 Ini adalah <span>nicest</span> halaman yang saya buat sebelumnya.
</div>
<div class=”boldText”>Here is some text on my page.</
div>
</body>
</html>
```

4. Simpan file
   Anda dapat menyimpannya sebagai css.html atau mengganti namanya menjadi css3.html. Simpan file di root dokumen Anda.
5. Buat file teks kosong baru.
   Menggunakan text editor, buat file kosong baru.
6. Tempatkan CSS berikut dalam file.

```
.boldText {
 font-weight: bold;
 }
 span {
 font-style: italic;
 }
```

7. Simpan file.
   Simpan file sebagai style.css di dalam root dokumen Anda. Perhatikan bahwa kita harus memastikan bahwa file diberi nama dengan huruf kecil semua dan memiliki ekstensi file yang benar, .css.
8. Buka browser kita dan lihat file css.html.
   Ketik http://localhost/css.html di bilah alamat browser. Jika kita menyimpan file sebagai css3.html, gunakan itu sebagai ganti css.html. Outputnya akan terlihat seperti pada gambar dibawah ini.



Gambar 2.7 Halaman dengan eksternal style sheet.

2.8 MENGGUNAKAN KELAS

Jika kita ingin menerapkan gaya yang sama ke banyak elemen, kita tidak perlu memberikan masing-masing ID yang berbeda karena kita dapat menentukan kelas untuk mengelola semuanya, seperti ini:

Hello

Ini menyatakan bahwa konten elemen ini (dan elemen lain yang menggunakan kelas) harus menerapkan gaya yang didefinisikan di kelas selamat datang. Setelah kelas diterapkan, kita dapat menggunakan aturan berikut, baik di header halaman atau di dalam style sheet eksternal untuk menyetel gaya kelas:

.welcome { font-style:italic; color:blue; }

Alih-alih simbol #, yang dicadangkan untuk ID, pernyataan kelas diawali dengan . (Titik).

Menggunakan Titik Koma

Dalam CSS, titik koma digunakan untuk memisahkan beberapa pernyataan CSS pada baris yang sama. Tetapi jika hanya ada satu pernyataan dalam aturan (atau dalam pengaturan gaya sebaris dalam tag HTML), kita dapat menghilangkan titik koma, seperti yang kita bisa untuk pernyataan terakhir dalam grup. Namun, untuk menghindari kesalahan CSS yang sulit ditemukan, kita dapat memilih untuk selalu menggunakan titik koma setelah setiap pengaturan CSS. kita kemudian dapat menyalin dan menempelkannya, dan sebaliknya memodifikasi properti, tanpa khawatir menghapus titik koma di tempat yang tidak terlalu diperlukan atau harus menambahkannya jika diperlukan.

2.9 ATURAN CSS

Setiap pernyataan dalam aturan CSS dimulai dengan pemilih, yang merupakan item yang akan diterapkan aturan. Misalnya, dalam tugas ini, h1 adalah selektor yang diberi ukuran font 240% lebih besar dari default:

h1 { font-size:240%; }

font-size adalah properti. Memberikan nilai 240% ke properti font-size dari selektor memastikan bahwa konten dari semua <h1> ... </h1> pasangan tag akan ditampilkan pada ukuran font yang 240% dari ukuran default. Semua perubahan aturan harus berada dalam simbol { dan } yang mengikuti pemilih. Dalam ukuran font: 240%; bagian sebelum : (titik dua) adalah properti, sedangkan sisanya adalah nilai yang diterapkan padanya. Terakhir datang ; (titik koma) untuk mengakhiri pernyataan. Dalam contoh ini, karena ukuran font adalah properti terakhir dalam aturan, titik koma tidak diperlukan (tetapi akan diperlukan jika tugas lain mengikuti).

Anda dapat membuat beberapa deklarasi gaya dalam beberapa cara berbeda. Pertama, kita dapat menggabungkannya pada baris yang sama, seperti ini:

h1 { font-size:240%; color:blue; }

Ini menambahkan tugas kedua yang mengubah warna semua judul <h1> menjadi biru. kita juga dapat menempatkan tugas satu per baris, seperti berikut:

```
h1 { font-size:240%;
color:blue; }
```

Atau kita dapat memberi ruang lebih sedikit pada tugas, sehingga mereka berbaris di bawah satu sama lain dalam kolom di titik dua, seperti ini:

```
h1 {
font-size:240%;
color :blue;
}
```

Dengan cara ini, kita dapat dengan mudah melihat di mana setiap kumpulan aturan baru dimulai, karena selektor selalu berada di kolom pertama, dan penetapan berikutnya berbaris rapi dengan semua nilai properti yang dimulai pada offset horizontal yang sama. Dalam contoh sebelumnya, titik koma terakhir tidak diperlukan, tetapi jika kita ingin menggabungkan kelompok pernyataan seperti itu menjadi satu baris, ini sangat cepat dilakukan dengan semua titik koma yang sudah ada. kita dapat menentukan selektor yang sama sebanyak yang kita inginkan, dan CSS menggabungkan semua properti. Jadi contoh sebelumnya juga dapat ditentukan sebagai:

```
h1 { font-size: 240%; }
h1 { color : blue; }
```

Bagaimana jika kita menentukan properti yang sama ke selektor yang sama dua kali?

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

Nilai terakhir yang ditentukan—dalam hal ini, biru—akan berlaku. Dalam satu file, mengulangi properti yang sama untuk selektor yang sama tidak akan ada gunanya, tetapi pengulangan seperti itu sering terjadi di halaman web kehidupan nyata ketika beberapa style sheet diterapkan. Ini adalah salah satu fitur berharga dari CSS, dan dari mana istilah cascading berasal.

2.10 FONT DAN TIPOGRAFI

Ada empat properti font utama yang dapat kita gaya menggunakan CSS: keluarga, gaya, ukuran, dan berat. Di antara mereka, kita dapat menyempurnakan cara teks ditampilkan di halaman web kita dan/atau saat dicetak.

Mengubah Font

Sejauh ini kita telah melihat banyak perubahan berat font untuk membuat font tampak tebal dan sedikit tentang gaya font untuk membuat font muncul miring. Namun, kita dapat melakukan lebih banyak hal dengan font di web menggunakan CSS, termasuk memilih jenis font dan memilih ukuran dan warna font.

font-family

Properti font-family menetapkan font yang akan digunakan. Ini juga mendukung daftar berbagai font dalam urutan preferensi dari kiri ke kanan, sehingga gaya dapat mundur dengan anggun saat pengguna tidak menginstal font pilihan. Misalnya, untuk menyetel font default untuk paragraf, kita dapat menggunakan aturan CSS seperti ini:

```
p { font-family:Verdana, Arial, Helvetica, sans-serif; }
```

Jika nama font terdiri dari dua kata atau lebih, kita harus menyertakan nama dalam tanda kutip, seperti ini:

```
p { font-family:"Times New Roman", Georgia, serif; }
```

Istilah *font family* menggambarkan jenis huruf atau tampilan font yang digunakan untuk teks. Font family dapat diubah menggunakan CSS tetapi ada batasan besar: Font yang akan kita gunakan juga harus tersedia di komputer pengunjung. Secara praktis, ini berarti kita harus menggunakan font "ramah web" tertentu yang muncul di sebagian besar komputer pengunjung. Ini juga berarti bahwa kita tidak selalu dapat menjamin font apa yang akan dilihat pengunjung. Jika pengunjung tidak memiliki font yang kita tentukan, browser pengunjung tersebut akan otomatis mengganti jenis font tersebut ke font umum. Properti CSS untuk font disebut font-family.

Catatan: Karena harus tersedia di hampir semua browser web dan sistem operasi, jenis font paling aman untuk digunakan di halaman web adalah Arial, Helvetica, Times New Roman, Times, Courier New, dan Courier. Font Verdana, Georgia, Comic Sans MS, Trebuchet MS, Arial Black, dan Impact aman untuk penggunaan Mac dan PC, tetapi tidak dapat diinstal pada sistem operasi lain seperti Linux. Font umum lainnya tetapi kurang aman adalah Palatino, Garamond, Bookman, dan Avant Garde. Jika kita menggunakan salah satu font yang kurang aman, pastikan kita menawarkan fallback dari satu atau lebih font yang lebih aman di CSS kita sehingga halaman web kita akan terdegradasi dengan anggun di browser tanpa font pilihan Anda. Gambar 2.8 menunjukkan dua set aturan CSS yang diterapkan.

Gambar 2.8 Memilih font-family

Saat menyetel font, praktik terbaik adalah memberikan daftar font yang dapat dipilih browser, seperti dalam contoh ini:

```
font-family: arial, helvetica, sans-serif;
```

Anda dapat mengatur font yang disarankan untuk seluruh halaman HTML dengan menggunakan selector untuk elemen <body>, seperti dalam contoh ini:

```
body {
 font-family: arial, helvetica, sans-serif;
}
```

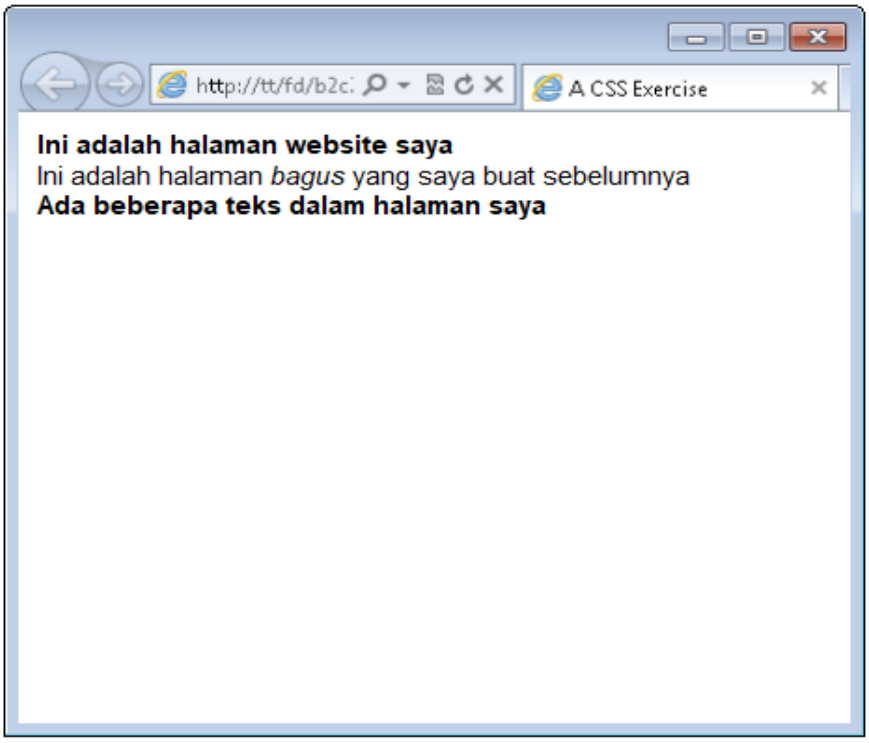
Setiap halaman yang menggunakan aturan CSS itu akan mencoba menampilkan teksnya terlebih dahulu dengan font Arial. Jika font tersebut tidak tersedia, font arial digunakan selanjutnya. Jika font tersebut tidak tersedia, maka font sans-serif digunakan. Jika tidak ada yang tersedia, maka browser memilih font untuk digunakan sendiri. Nilai umum untuk font-family adalah: arial, helvetica, sans-serif, “Arial Black”, Gadget, sans-serif, Georgia, serif, “Times New Roman”, Times, serif.

Daftar kode dibawah menunjukkan CSS yang kita lihat pada contoh sebelumnya. Cantuman ini menambahkan properti CSS font-family ke isi halaman, yang berarti bahwa pengaturan font-family ini akan diterapkan di seluruh halaman.

Daftar : Mengatur Nilai Font-Family dengan CSS

```
body {
 font-family: arial ,sans-serif;
}
.boldText {
 font-weight: bold;
}
span {
 font-style: italic;
}
```

Jika dilihat di browser menggunakan HTML yang sama dari latihan sebelumnya, hasilnya terlihat seperti gambar dibawah ini:



Gambar 2.9 Mengubah font family dengan CSS.

font-style

Dengan properti font-style kita dapat memilih untuk menampilkan font secara normal, miring, atau miring. Aturan berikut membuat tiga kelas (normal, miring, dan miring) yang dapat diterapkan ke elemen untuk membuat efek ini:

```
.normal { font-style:normal; }
.italic { font-style:italic; }
.oblique { font-style:oblique; }
```

Pengukuran

CSS mendukung berbagai unit pengukuran yang mengesankan, memungkinkan kita untuk menyesuaikan halaman web kita secara tepat dengan nilai tertentu, atau menurut dimensi relatif.

Font size

Mengatur ukuran huruf

Seberapa besar teks yang muncul di halaman web adalah ukuran fontnya. Kita dapat mengatur ukuran font menggunakan properti font-size CSS.

Piksel

Ukuran piksel bervariasi sesuai dengan dimensi dan kedalaman piksel monitor pengguna. Satu piksel sama dengan lebar/tinggi satu titik di layar, sehingga pengukuran ini paling cocok untuk monitor. Sebagai contoh:

```
.classname { margin:5px; }
```

Poin

Sebuah titik setara dengan ukuran 1/72 inci. Pengukuran berasal dari latar belakang desain cetak dan paling cocok untuk media itu, tetapi juga biasa digunakan pada monitor. Sebagai contoh:

```
.classname { font-size:14pt; }
```

Inci

Satu inci setara dengan 72 poin dan juga merupakan jenis pengukuran yang paling cocok untuk dicetak. Sebagai contoh:

```
.classname { width:3in; }
```

Sentimeter

Sentimeter adalah unit pengukuran lain yang paling cocok untuk dicetak. Satu sentimeter sedikit di atas 28 poin. Sebagai contoh:

```
.classname { height:2cm; }
```

Milimeter

Satu milimeter adalah 1/10 sentimeter (atau hampir 3 poin). Milimeter adalah ukuran lain yang paling cocok untuk dicetak. Sebagai contoh:

.classname { font-size:5mm; }

Foto

Pica adalah ukuran tipografi cetak lainnya, yang setara dengan 12 poin. Sebagai contoh:

.classname { font-size:1pc; }

Em

Em sama dengan ukuran font saat ini dan oleh karena itu merupakan salah satu pengukuran yang lebih berguna untuk CSS karena digunakan untuk menggambarkan dimensi relatif. Sebagai contoh:

.classname { font-size:2em; }

Ex

Mantan juga terkait dengan ukuran font saat ini; itu setara dengan tinggi huruf kecil x. Ini adalah unit pengukuran yang kurang populer yang paling sering digunakan sebagai perkiraan yang baik untuk membantu mengatur lebar kotak yang akan berisi beberapa teks. Sebagai contoh:

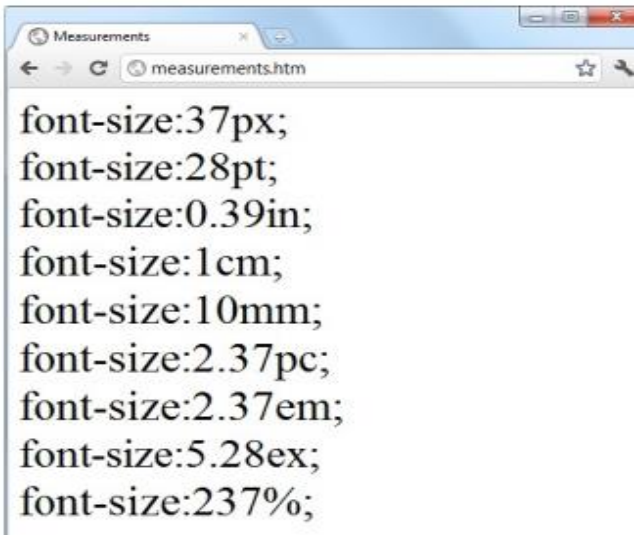
.classname { width:20ex; }

Persen

Unit ini terkait dengan em karena ukurannya tepat 100 kali lebih besar (bila digunakan pada font). Sedangkan 1 em sama dengan ukuran font saat ini, ukuran yang sama adalah 100 dalam persen. Saat tidak terkait dengan font, unit ini relatif terhadap ukuran wadah properti yang sedang diakses. Sebagai contoh:

.classname { height:120%; }

Gambar 2.10 menunjukkan masing-masing jenis pengukuran ini pada gilirannya digunakan untuk menampilkan teks dalam ukuran yang hampir sama.



Gambar 2.10 Pengukuran berbeda yang menampilkan hampir sama

Seperti yang dijelaskan di bagian sebelumnya tentang pengukuran, ada banyak cara kita dapat mengubah ukuran font. Tapi ini semua bermuara pada dua jenis utama: tetap dan relatif. Pengaturan tetap terlihat seperti aturan berikut, yang mengatur ukuran font paragraf default menjadi 14 poin:

```
p { font-size:14pt; }
```

Atau, kita mungkin ingin bekerja dengan ukuran font default saat ini, menggunakannya untuk menata berbagai jenis teks seperti heading. Dalam aturan berikut, ukuran relatif dari beberapa header ditentukan, dengan tag <h4> mulai 20% lebih besar dari default, dan dengan setiap ukuran lebih besar, 40% lebih besar dari yang sebelumnya:

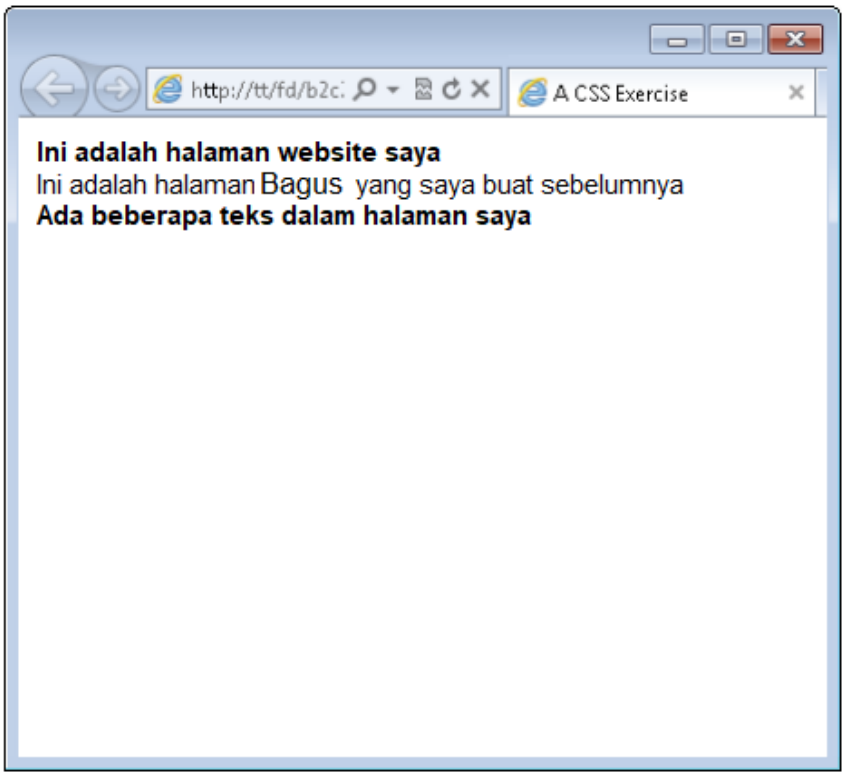
```
h1 { font-size:240%; }
h2 { font-size:200%; }
h3 { font-size:160%; }
h4 { font-size:120%; }
```

Gambar 2.11 menunjukkan pilihan ukuran font yang digunakan.

Gambar 2.11 Mengatur empat ukuran heading dan ukuran paragraf default

Metode untuk memilih ukuran font

Saat memilih metode ukuran font, kita dapat menggunakan persentase, em, poin, dan piksel. Poin dan piksel adalah ukuran tetap dan beberapa browser dapat mengalami kesulitan mengubah ukurannya, atau lebih tepatnya, browser tidak mengizinkan pengunjung untuk mengubah ukuran teks tanpa menggunakan alat zoom. Persentase dan em memungkinkan pengubahan ukuran.



Gambar 2.12 Mengubah ukuran font dengan CSS.

Ketika digabungkan dengan HTML dari latihan sebelumnya, kita mendapatkan halaman seperti itu pada Gambar dibawah ini. Perhatikan peningkatan ukuran font untuk kata yang paling bagus, berkat peningkatan ukuran yang disetel dengan em.

Saat menggunakan em untuk ukuran font, nilai em 1,0 sesuai dengan 100%. Oleh karena itu, 0.9em akan menjadi sekitar 90%, sedangkan 1.7em (seperti dalam contoh) pada dasarnya adalah 170%. Font yang diatur dengan piksel atau titik menggunakan singkatannya, seperti dalam contoh berikut:

```
font-size: 12px;
font-size: 12pt;
```

font-weight

Menggunakan properti font-weight kita dapat memilih seberapa berani untuk menampilkan font. Ini mendukung sejumlah nilai, tetapi nilai utama yang akan kita gunakan cenderung normal dan tebal, seperti ini:

```
.bold { font-weight:bold; }
```

Mengatur warna font

Sama seperti ukuran font yang dapat diatur, warna font juga dapat diatur. Kita harus berhati-hati dalam memilih warna font karena ini akan mempengaruhi kenyamanan mata dalam melihat font berwarna. Dalam hal ini, kita dapat menggunakan nama yang ramah untuk warna umum, seperti merah, biru, hijau, dan sebagainya, atau kita dapat menggunakan kode heksadesimal, atau singkatnya kode heksa. Kode hex adalah kode tiga sampai enam karakter yang sesuai dengan campuran warna *Red, Green dan Blue* (RGB) yang sesuai untuk

mendapatkan warna yang diinginkan. Tabel dibawah ini menunjukkan beberapa kode hex umum dan warna yang sesuai.

Tabel 2.1 kode warna untuk Hex

Kode	***Warna***
#FF0000	Merah
#00FF00	Hijau
#0000FF	Biru
#666666	Abu-abu gelap
#000000	Hitam
#FFFFFF	Putih
#EEEE00	Kuning
#FFA500	Oranye

Kode hex adalah cara yang lebih akurat dan disukai untuk mengatur warna dalam HTML tetapi sulit untuk diingat. Alat seperti Lab Warna Visibone di www.visibone.com/colorlab sangat penting untuk mendapatkan kode hex yang sesuai dengan warna yang ingin kita gunakan. Warna font diatur menggunakan properti CSS warna, seperti dalam contoh ini (yang merupakan kode untuk merah):

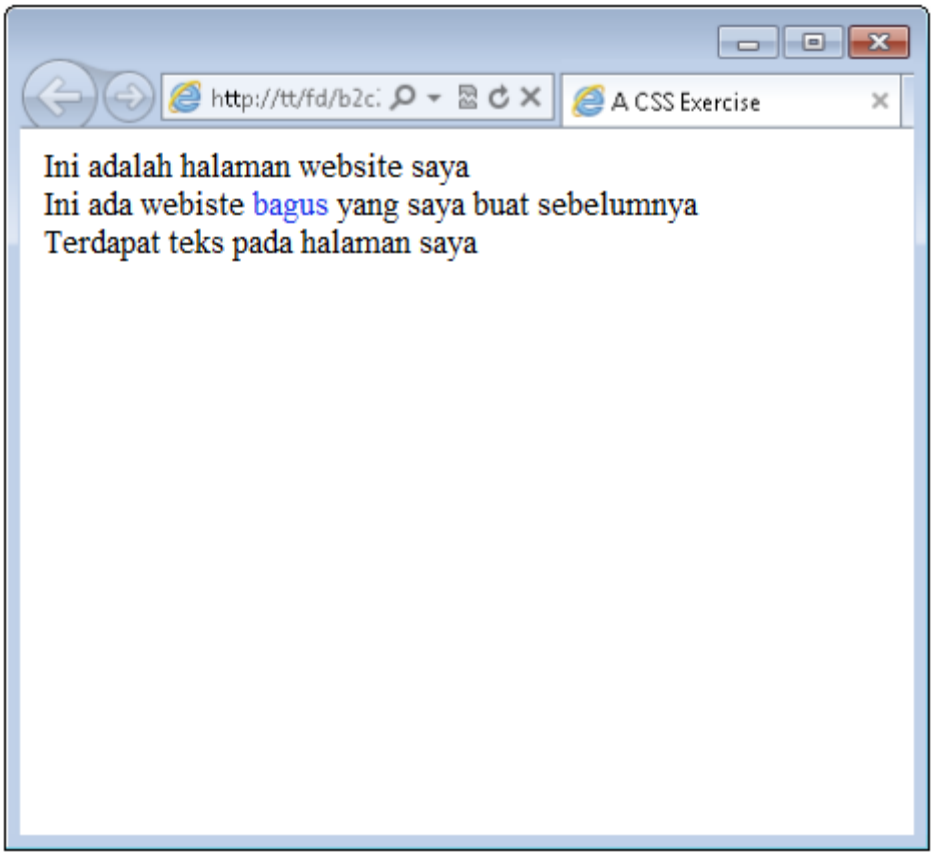
color: #FF0000;

Daftar dibawah ini menunjukkan CSS untuk mengubah warna elemen menjadi biru menggunakan kode hex:

Contoh Mewarnai Font Menggunakan CSS

```
span {
 color: #0000FF;
}
```

Jika dilihat di browser dengan HTML yang dibuat sebelumnya di bab ini, outputnya terlihat seperti Gambar dibawah ini. Perhatikan warna biru untuk kata “bagus” di halaman.



Gambar 2.13 Mengubah warna font menjadi biru

2.11 MENAMBAHKAN BORDER

Batas dapat membantu memberikan pemisahan visual antar elemen pada halaman. kita dapat menambahkan border di sekitar apa saja di HTML dan ada beberapa gaya border untuk dipilih. Border ditambahkan dengan properti CSS border. Saat membuat border dengan CSS, kita mengatur tiga hal:

- Ketebalan border
- Gaya Border
- Warna border

Ketiga item ini diatur dalam daftar, dipisahkan oleh spasi, seperti dalam contoh ini:

```
border: 1px solid black;
```

Dalam contoh ini, border akan dibuat dan tebalnya 1 piksel. Border akan padat dan akan berwarna hitam. Beberapa gaya batas umum ditunjukkan pada tabel berikut.

Tabel 2.2 border style di CSS

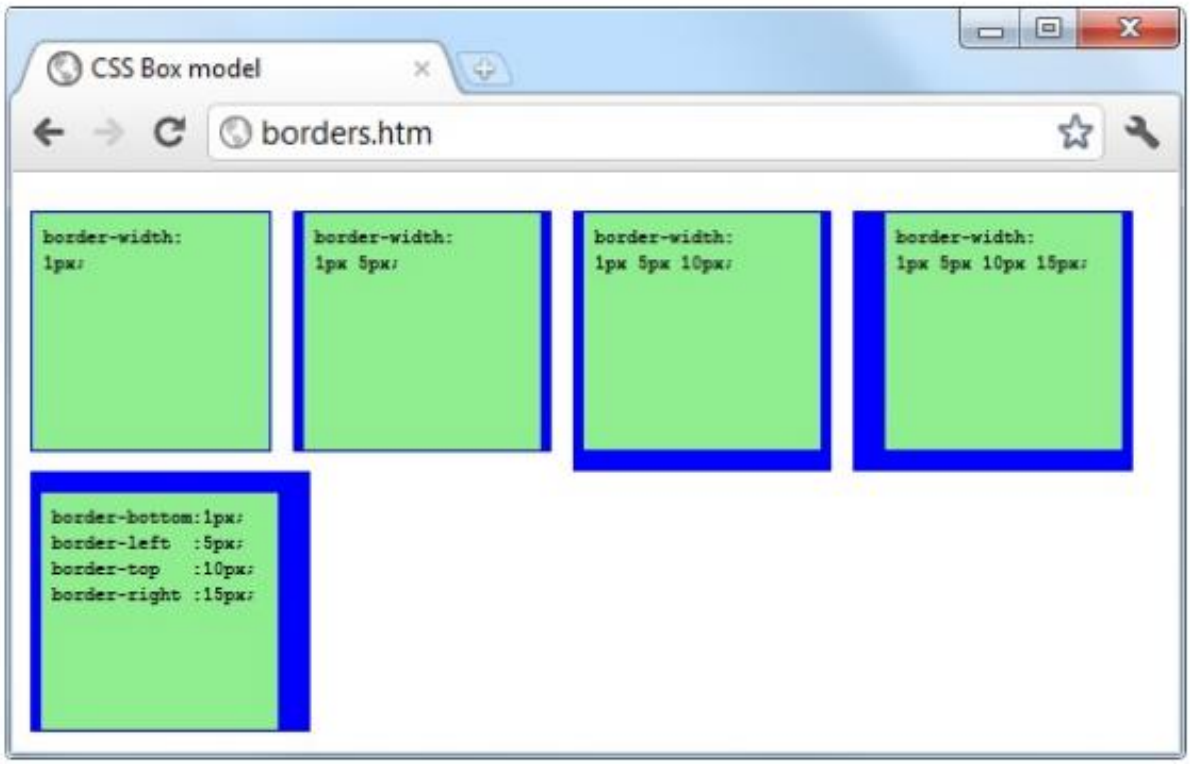
Style	***Deskripsi***
Solid	Garis padat
Titik-titik	Garis titik-titik
Tanda hubung	Garis dengan efek tanda hubung
Double	Dua garis padat

Menerapkan Border

Tingkat batas model kotak mirip dengan margin kecuali bahwa tidak ada keruntuhan. Ini adalah level berikutnya saat kita beralih ke model kotak. Properti utama yang digunakan untuk mengubah batas adalah batas, batas kiri, batas atas, batas kanan, dan batas bawah, dan masing-masing dapat memiliki subproperti lain yang ditambahkan sebagai sufiks, seperti -warna, -gaya, dan -lebar . Empat cara untuk mengakses pengaturan properti individual yang digunakan untuk properti margin juga berlaku dengan properti border-width, jadi semua yang berikut ini adalah aturan yang valid:

```
/* All borders
border-width:1px;
Top/bottom left/right
border-width:1px 5px;
Top left/right bottom
border-width:1px 5px 10px;
Top right bottom left */
border-width:1px 5px 10px 15px;
```

Gambar 2.14 menunjukkan masing-masing aturan ini diterapkan pada sekelompok elemen persegi. Bagian pertama, kita dapat dengan jelas melihat bahwa semua batas memiliki lebar 1 piksel. Elemen kedua, bagaimanapun, memiliki lebar batas atas dan bawah 1 piksel, sedangkan lebar sisinya masing-masing 5 piksel.



Gambar 2.14 Menerapkan nilai aturan batas panjang dan singkat

Elemen ketiga memiliki lebar atas 1 piksel, sisi-sisinya lebar 5 piksel, dan bagian bawahnya lebar 10 piksel. Elemen keempat memiliki lebar batas atas 1 piksel, lebar batas kanan 5 piksel, lebar batas bawah 10 piksel, dan lebar batas kiri 15 piksel. Elemen terakhir, di bawah yang sebelumnya, tidak menggunakan aturan singkatan; sebagai gantinya, masing-

masing lebar border diatur secara terpisah. Seperti yang kita lihat, dibutuhkan lebih banyak mengetik untuk mencapai hasil yang sama.

Saatnya latihan untuk membuat batas di sekitar beberapa elemen. Mengikuti langkah-langkah ini.

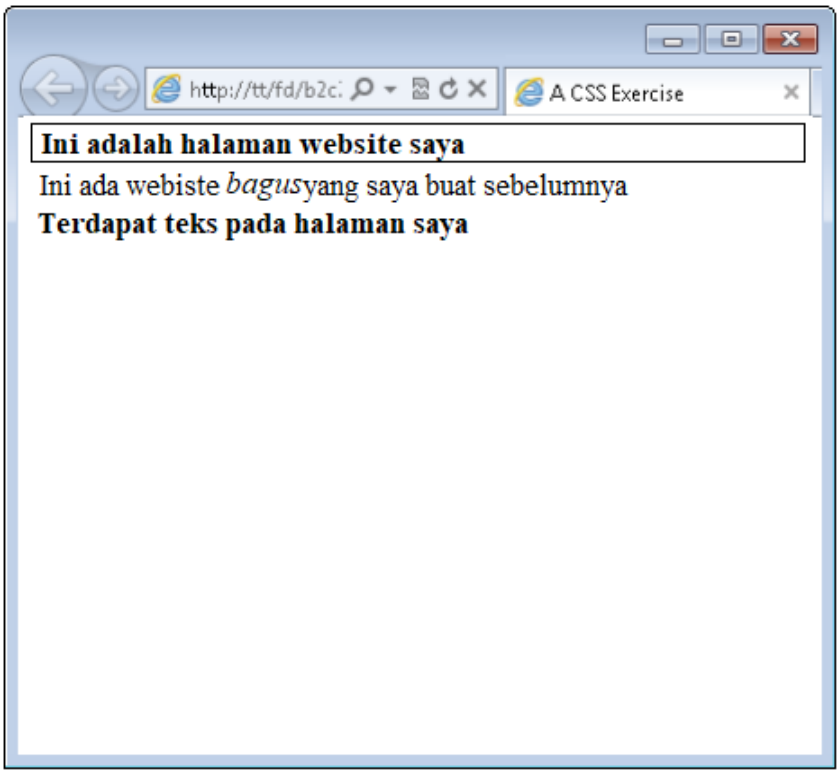
1. Buka text edirot
2. Verifikasi file HTML dari latihan sebelumnya.
   HTML dari latihan sebelumnya adalah titik awal untuk latihan ini. Jika milik kita tidak terlihat seperti ini, ubah agar terlihat seperti HTML ini. Bagi kita yang memiliki file ini persis seperti pada latihan sebelumnya, satu-satunya hal yang perlu kita lakukan adalah menambahkan kelas bernama addBorder di elemen pertama.

```
<!doctype html>
<html>
<head>
<title>A CSS Exercise</title>
<link rel=”stylesheet” type=”text/css” href=”style.
css”>
</head>
<body>
<div class=”boldText addBorder”>This is my web page.</
div>
<div>
 This is the <span>nicest</span> page I’ve made yet.
</div>
<div class=”boldText”>Here is some text on my page.</
div>
</body>
</html>
```

3. Save file HTML
   Simpan ini dengan format css-border.html dan tempatkan ini pada dokumen root Anda
4. Open file CSS
   File CSS harus berisi kelas yang disebut boldText dan aturan CSS yang mengubah semua elemen menjadi miring. Di dalam file CSS, tambahkan dan ubah CSS kita sehingga terlihat seperti berikut:

```
.boldText {
 font-weight: bold;
}
span {
 font-style: italic;
}
.addBorder {
 border: 3px double black;
}
```

5. Simpan file CSS
   Simpan file sebagai style.css di root dokumen Anda.
6. Lihat halaman di browser.
   Buka browser web kita dan arahkan ke http://localhost/css-border.html untuk melihat halaman. kita akan melihat halaman seperti gambar dibawah ini.



Gambar 2.15 Menambahkan batas ke elemen div

2.12 MENGELOLA GAYA TEKS

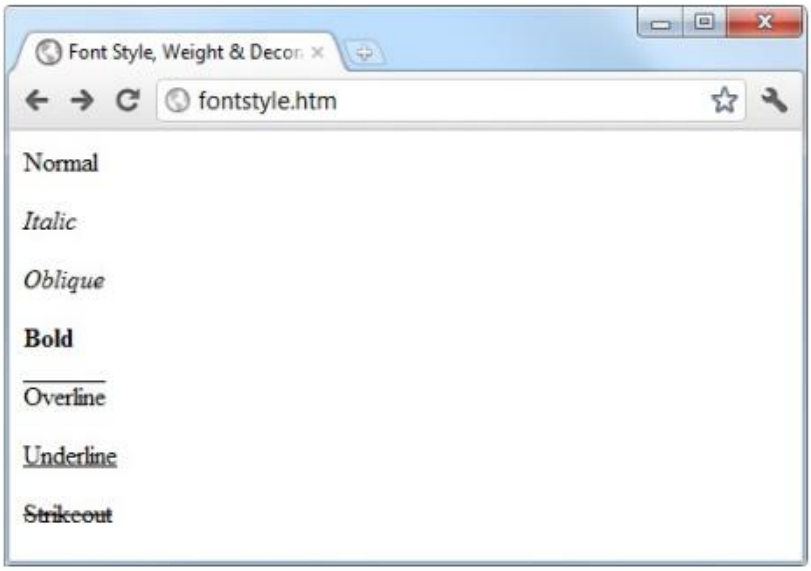
Terlepas dari font yang digunakan, kita dapat memodifikasi lebih lanjut cara teks ditampilkan dengan mengubah dekorasi, spasi, dan perataannya. Ada persilangan antara properti teks dan font, dalam efek seperti miring atau teks tebal dicapai melalui properti *font-style* dan font-weight, sementara yang lain seperti menggarisbawahi memerlukan properti teks-dekorasi.

Dekorasi

Dengan properti dekorasi teks, kita dapat menerapkan efek ke teks seperti garis bawah, garis tembus, garis atas, dan kedipan. Aturan berikut membuat kelas baru yang dipanggil over yang menerapkan overline ke teks (bobot garis over, under, dan through akan cocok dengan font):

```
.over { text-decoration:overline; }
```

Pada Gambar 2.16 Anda dapat melihat pilihan gaya font, bobot, dan dekorasi.



Gambar 2.16 Contoh gaya dan aturan dekorasi yang tersedia

Spasi

Sejumlah properti yang berbeda memungkinkan kita untuk mengubah spasi baris, kata, dan huruf. Misalnya, aturan berikut mengubah spasi baris untuk paragraf dengan memodifikasi properti tinggi baris menjadi 25% lebih besar, sedangkan properti spasi kata diatur ke 30 piksel, dan spasi huruf diatur ke 3 piksel:

```
p {
line-height :125%;
word-spacing :30px;
letter-spacing:3px;
}
```

Alignment

Empat jenis perataan teks tersedia di CSS: kiri, kanan, tengah, dan ratakan. Dalam aturan berikut, teks paragraf default diatur ke justifikasi penuh:

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

Transformasi

Ada empat properti yang tersedia untuk mengubah teks: tidak ada, huruf besar, huruf besar, dan huruf kecil. Aturan berikut membuat kelas yang disebut atas yang akan memastikan bahwa semua teks ditampilkan dalam huruf besar saat digunakan:

```
.upper { text-transform:uppercase; }
```

Indentasi

Menggunakan properti indentasi teks, kita dapat membuat indentasi baris pertama dari blok teks dengan jumlah tertentu. Aturan berikut mengindentasi baris pertama setiap paragraf sebanyak 20 piksel, meskipun unit pengukuran yang berbeda atau peningkatan persen juga dapat diterapkan:

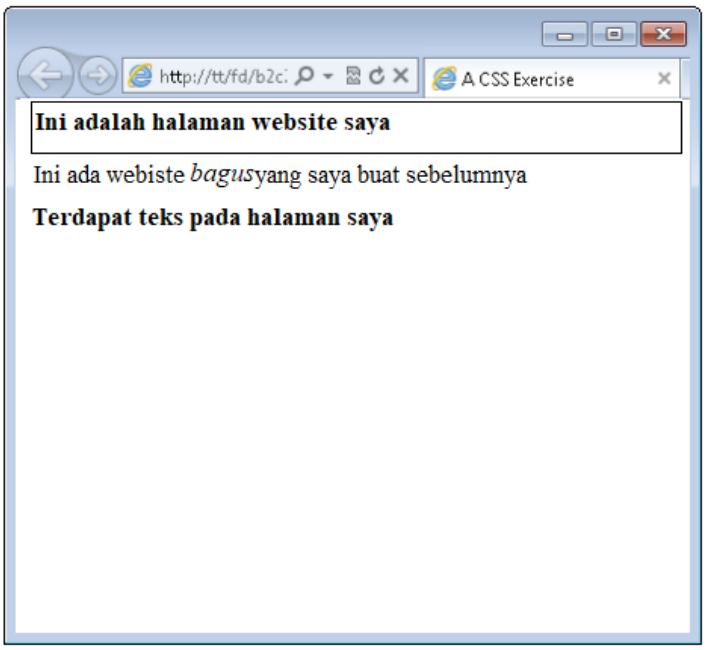
p { text-indent:20px; }
Pada Gambar 19-9, aturan berikut telah diterapkan pada bagian teks:
p { line-height :150%;
word-spacing :10px;
letter-spacing:1px;
}
.justify { text-align :justify; }
.uppercase { text-transform:uppercase; }
.indent { text-indent :20px; }

Gambar 2.17 Aturan indentasi, huruf besar, dan spasi diterapkan

Anda mungkin telah memperhatikan dalam latihan ini bahwa kita sekarang memiliki dua kelas di pertama di halaman. Itu adalah fitur hebat dari kelas karena kita dapat menggunakan lebih dari satu elemen untuk menggabungkannya. kita dapat bereksperimen dengan CSS dari latihan ini untuk menambahkan gaya batas yang berbeda ke elemen yang berbeda di halaman. kita mungkin tidak menyukai seberapa dekat teks dengan border pada Gambar dibawah. Kami yakin tidak. kita dapat mengubah ini dengan CSS. Properti padding CSS mengubah seberapa dekat teks akan sampai ke tepi bagian dalam border. Misalnya, kita dapat mengubah CSS untuk kelas addBorder agar terlihat seperti ini:

```
.addBorder {
 border: 3px double black;
 padding: 5px;
}
```

Ketika kita melakukannya, halaman yang dihasilkan akan terlihat seperti digambar dibawah ini.



Gambar 2.18 Menambahkan padding dalam kelas addBorder.

Padding dapat ditambahkan untuk memindahkan teks lebih jauh dari bordernya. Padding dapat diterapkan ke elemen apa pun, terlepas dari apakah elemen tersebut memiliki batas, untuk memindahkan konten elemen tersebut.

Saat kita menambahkan padding, konten elemen menjauh dari semua tepi. Namun, kita juga dapat menambahkan padding agar konten menjauh dari atas, bawah, kanan, atau kiri, atau kombinasi apa pun di dalamnya. Ini dilakukan dengan properti *padding-top, padding-bottom*, *padding-right,* dan *padding-left*, masing-masing. Ada metode pintasan untuk menyetel padding yang melihat semua padding didefinisikan pada satu baris. Pintasan itu tidak digunakan di sini, tetapi kita akan melihatnya di CSS orang lain. Di mana padding memindahkan elemen dari dalam, ada juga properti untuk memindahkan atau menggeser elemen dari luar. Elemen ini disebut margin, dan kita akan membahasnya nanti di bab ini ketika kita berbicara tentang membuat layout halaman.

2.13 MENYESUAIKAN PADDING

Level model kotak terdalam (selain konten elemen) adalah padding, yang diterapkan di dalam batas dan/atau margin apa pun. Properti utama yang digunakan untuk memodifikasi padding adalah padding, padding-left, padding-top, padding-right, dan padding-bottom. Empat cara mengakses pengaturan properti individual yang digunakan untuk properti margin dan border juga berlaku dengan properti padding, jadi semua berikut ini adalah aturan yang valid:

```
/* All padding
padding:1px;
Top/bottom and left/right
padding:1px 2px;
Top, left/right and bottom
padding:1px 2px 3px;
Top, right, bottom and left */
```

padding:1px 2px 3px 4px;

Gambar 2.19 menunjukkan aturan padding (ditampilkan dalam huruf tebal) pada Contoh 3 yang diterapkan ke beberapa teks dalam sel tabel (seperti yang didefinisikan oleh aturan display:table-cell;, yang membuat elemen enkapsulasi ditampilkan seperti a table cell), yang tidak diberi dimensi sehingga hanya akan membungkus teks sedekat mungkin. Akibatnya ada padding 10 piksel di atas elemen dalam, 20 piksel di sebelah kanan, 30 piksel di bawahnya, dan 40 piksel di sebelah kirinya.

Contoh 3. Menerapkan bantalan

```
<!DOCTYPE html>
<html>
<head>
<title>Padding</title>
<style>
#object1 {
border-style:solid;
border-width:1px;
background :orange;
color :darkred;
font-family :Arial;
font-size :12px;
text-align :justify;
display :table-cell;
width :148px;
padding :10px 20px 30px 40px; }
</style>
</head>
<body>
<div id='object1'>To be, or not to be that is the question:
Whether 'tis Nobler in the mind to suffer
The Slings and Arrows of outrageous Fortune,
Or to take Arms against a Sea of troubles,
And by opposing end them.</div>
</body>
</html>
```

Gambar 2.19 Menerapkan nilai padding yang berbeda ke objek

Isi Objek

Jauh di dalam level model kotak, di tengahnya, terletak sebuah elemen yang dapat ditata dengan semua cara yang dibahas dalam bab ini, dan yang dapat (dan biasanya akan) berisi sub-elemen lebih lanjut, yang pada gilirannya dapat berisi sub-sub- elemen, dan seterusnya, masing-masing dengan pengaturan gaya dan model kotaknya sendiri.

2.14 SELEKTOR CSS

Cara kita mengakses satu atau lebih elemen disebut seleksi, dan bagian dari aturan CSS yang melakukan ini dikenal sebagai pemilih. Seperti yang kita duga, ada banyak jenis pemilih.

Selektor Jenis

Selektor jenis bekerja pada jenis elemen HTML seperti atau <i>. Misalnya, aturan berikut akan memastikan bahwa semua teks dalam tag ... sepenuhnya dibenarkan:

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

Selektor Keturunan

Selektor turunan memungkinkan kita menerapkan gaya ke elemen yang ada di dalam elemen lain. Misalnya, aturan berikut menyetel semua teks dalam tag ... menjadi merah, tetapi hanya jika teks tersebut muncul di dalam tag ... (seperti ini: Hello di sana):

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

Selektor turunan dapat terus bersarang tanpa batas, jadi berikut ini adalah aturan yang benar-benar valid untuk membuat teks menjadi biru di dalam teks tebal, di dalam elemen daftar dari daftar yang tidak diurutkan:

```
ul li b { color:blue; }
```

Sebagai contoh praktis, misalkan kita ingin menggunakan sistem penomoran yang berbeda untuk daftar terurut yang bersarang di dalam daftar terurut lainnya. kita dapat mencapai ini dengan cara berikut, yang akan menggantikan penomoran numerik default (mulai dari 1) dengan huruf kecil (mulai dari a):

```
<!DOCTYPE html>
<html>
<head>
<style>
ol ol { list-style-type:lower-alpha; }
</style>
</head>
<body>
<ol>
li>One</li>
<li>Two</li>
<li>Three
<ol>
<li>One</li>
<li>Two</li>
<li>Three</li>
</ol>
</li>
</ol>
</body>
</html>
```

Hasil pemuatan HTML ini ke browser web adalah sebagai berikut, di mana kita dapat melihat bahwa elemen daftar kedua ditampilkan secara berbeda:

1. One
2. Two
3. Three
   a. One
   b. Two
   c. Three

Selector Child

Selektor turunan mirip dengan selektor turunan tetapi lebih membatasi kapan gaya akan diterapkan, dengan memilih hanya elemen yang merupakan anak langsung dari elemen lain. Misalnya, kode berikut menggunakan selektor turunan yang akan mengubah teks tebal apa pun dalam paragraf menjadi merah, meskipun teks tebal itu sendiri berada di dalam huruf miring (seperti ini <i>Hello di sana< /i>):

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

Dalam contoh ini, kata Hello ditampilkan dengan warna merah. Namun, ketika jenis perilaku yang lebih umum ini tidak diperlukan, selektor anak dapat digunakan untuk mempersempit

cakupan pemilih. Misalnya, selektor anak berikut akan menyetel teks tebal menjadi merah hanya jika elemen tersebut adalah anak langsung dari sebuah paragraf, dan tidak berisi elemen lain:

```
p > b { color:red; }
```

Sekarang kata Hello tidak akan berubah warna karena bukan turunan langsung dari paragraf. Sebagai contoh praktis, misalkan kita hanya ingin memberanikan elemen <li> yang merupakan turunan langsung dari elemen <ol>. kita dapat mencapai ini sebagai berikut, di mana elemen <li> yang merupakan anak langsung dari elemen <ul> tidak dikuatkan:

```
<!DOCTYPE html>
<html>
<head>
<style>
ol > li { font-weight:bold; }
</style>
</head>
<body>
<ol>
<li>One</li>
<li>Two</li>
<li>Three</li>
</ol>
<ul>
<li>One</li>
<li>Two</li>
<li>Three</li>
</ul>
</body>
</html>
```

Hasil loading HTML ini ke dalam browser adalah sebagai berikut:

1. One
2. Two
3. Three

- One
- Two
- Three

Selektor ID

Jika kita memberi elemen nama ID (seperti ini:) maka kita dapat langsung mengaksesnya dari CSS dengan cara berikut, yang mengubah semua teks dalam elemen menjadi miring:

```
#mydiv { font-style:italic; }
```

ID hanya dapat digunakan sekali dalam dokumen, jadi hanya kemunculan pertama yang ditemukan yang akan menerima nilai properti baru yang ditetapkan oleh aturan CSS. Tetapi di CSS kita dapat langsung mereferensikan ID apa pun yang memiliki nama yang sama, selama ID tersebut muncul dalam tipe elemen yang berbeda, seperti ini:

Hello Hello

Karena ID biasanya hanya berlaku untuk elemen unik, aturan berikut akan menerapkan garis bawah hanya untuk kemunculan pertama myid:

#myid { text-decoration:underline; }

Namun, kita dapat memastikan bahwa CSS menerapkan aturan untuk kedua kejadian seperti ini:

span#myid { text-decoration:underline; }
Atau lebih ringkasnya seperti ini (lihat Memilih berdasarkan Grup):
div#myid { text-decoration:underline; }
span#myid, div#myid { text-decoration:underline; }

Catatan: Saya tidak menyarankan menggunakan bentuk pemilihan ini karena JavaScript apa pun yang juga harus mengakses elemen-elemen ini tidak dapat dengan mudah melakukannya karena fungsi getElementByID() yang umum digunakan hanya akan mengembalikan kemunculan pertama. Untuk mereferensikan contoh lain, sebuah program harus mencari seluruh daftar elemen dalam dokumen—tugas yang lebih sulit untuk dilakukan. Jadi umumnya lebih baik untuk selalu menggunakan nama ID yang unik.

Selektor Kelas

Bila ada sejumlah elemen di halaman yang ingin kita bagikan gaya yang sama, kita dapat menetapkan semua nama kelas yang sama (seperti ini:); kemudian, buat aturan tunggal untuk memodifikasi semua elemen tersebut sekaligus, seperti pada aturan berikut, yang membuat offset margin kiri 10 piksel untuk semua elemen menggunakan kelas:

.myclass { margin-left:10px; }

Di browser modern, kita dapat membuat elemen HTML menggunakan lebih dari satu kelas dengan memisahkan nama kelas dengan spasi, seperti ini: . Namun, ingat bahwa beberapa browser yang sangat lama hanya mengizinkan satu nama kelas dalam argumen kelas. kita dapat mempersempit cakupan tindakan kelas dengan menentukan jenis elemen yang harus diterapkan. Misalnya, aturan berikut ini hanya berlaku untuk paragraf yang menggunakan kelas utama:

p.main { text-indent:30px; }

Dalam contoh ini, hanya paragraf yang menggunakan kelas utama (seperti ini: <p class="main">) yang akan menerima nilai properti baru. Jenis elemen lain yang mungkin

mencoba menggunakan kelas (seperti) tidak akan terpengaruh oleh aturan ini.

Atribut Selektor

Banyak tag HTML mendukung atribut, dan menggunakan jenis selektor ini dapat menyelamatkan kita dari keharusan menggunakan ID dan kelas untuk merujuknya. Misalnya, kita dapat langsung mereferensikan atribut dengan cara berikut, yang menyetel semua elemen dengan atribut type="submit" ke lebar 100 piksel:

```
[type="submit"] { width:100px; }
```

Jika kita ingin mempersempit cakupan pemilih, misalnya, hanya membentuk elemen input dengan tipe atribut tersebut, kita dapat menggunakan aturan berikut sebagai gantinya:

```
form input[type="submit"] { width:100px; }
```

Catatan: Selektor atribut juga berfungsi pada ID dan kelas sehingga, misalnya, [class~="classname"] berfungsi persis seperti selektor kelas .classname (kecuali bahwa yang terakhir memiliki prioritas lebih tinggi). Demikian juga, [id="idname"] sama dengan menggunakan selektor ID #idname. Selektor kelas dan ID diawali dengan # dan . karena itu dapat dilihat sebagai singkatan untuk penyeleksi atribut, tetapi dengan prioritas yang lebih tinggi. Operator ~= cocok dengan atribut meskipun itu adalah salah satu dari grup atribut yang dipisahkan spasi.

Selektor Universal

Wildcard * atau selektor universal cocok dengan elemen apa pun, jadi aturan berikut akan membuat dokumen berantakan total dengan memberikan batas hijau ke semua elemen -elemennya:

```
* { border:1px solid green; }
```

Oleh karena itu, kecil kemungkinan kita akan menggunakan * sendiri, tetapi sebagai bagian dari aturan majemuk, itu bisa sangat kuat. Misalnya, aturan berikut akan menerapkan gaya yang sama seperti sebelumnya, tetapi hanya untuk semua paragraf yang merupakan sub-elemen dari elemen dengan kotak ID, dan hanya selama mereka bukan turunan langsung:

```
#boxout * p {border:1px solid green; }
```

Mari kita lihat apa yang terjadi di sini. Selektor pertama setelah #boxout adalah simbol *, jadi ini merujuk ke elemen apa pun di dalam objek boxout. Selektor p berikut kemudian mempersempit fokus pemilihan dengan mengubah selektor untuk diterapkan hanya pada paragraf (sebagaimana didefinisikan oleh p) yang merupakan sub-elemen dari elemen yang dikembalikan oleh selektor *. Oleh karena itu, aturan CSS ini melakukan tindakan berikut (di mana saya menggunakan istilah objek dan elemen secara bergantian):

1. Temukan objek dengan ID boxout.
2. Temukan semua sub-elemen objek yang dikembalikan pada langkah 1.

3. Temukan semua p sub-elemen dari objek yang dikembalikan pada langkah 2 dan, karena ini adalah selektor terakhir dalam grup, temukan juga semua p sub-dan sub-elemen (dan seterusnya) dari objek yang dikembalikan pada langkah 2.
4. Terapkan gaya dalam karakter {and } ke objek yang dikembalikan pada langkah 3.

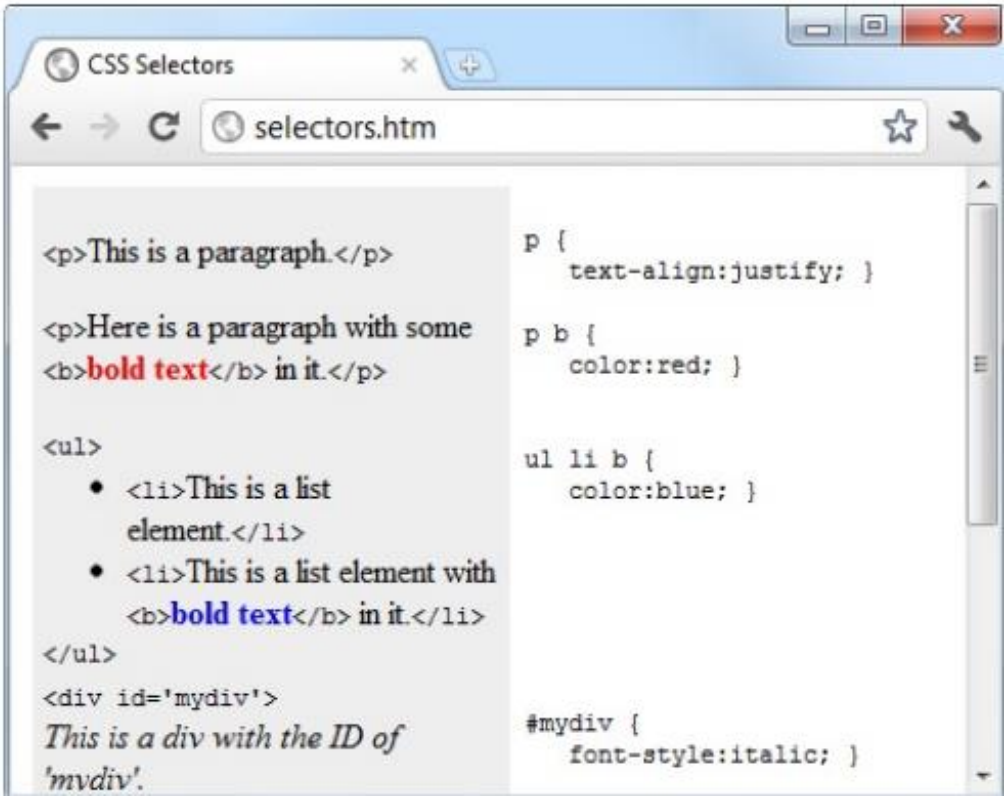
Hasil bersih dari ini adalah bahwa batas hijau hanya diterapkan pada paragraf yang adalah cucu (atau cicit, dll.) dari elemen utama.

2.15 MEMILIH BERDASARKAN GRUP

Menggunakan CSS kita dapat menerapkan aturan ke lebih dari satu elemen, kelas, atau jenis selektor lainnya secara bersamaan dengan memisahkan selektor dengan koma. Jadi, misalnya, aturan berikut akan menempatkan garis oranye putus-putus di bawah semua paragraf, elemen dengan ID idname, dan semua elemen yang menggunakan class classname:

```
p, #idname, .classname { border-bottom:1px  dotted orange; }
```

Gambar 2.20 menunjukkan berbagai selektor yang digunakan, dengan aturan yang diterapkan bersamanya.



Gambar 2.20 Beberapa aturan HTML dan CSS yang digunakan olehnya

Mengubah List style

Kita bisa. Gaya peluru untuk daftar ditentukan oleh properti CSS list-style-type. Ada banyak nilai untuk properti list-style-type. Tabel 2-3 menunjukkan beberapa yang umum.

Tabel 2.3 List Style

Style	***Deskripsi***
circle	Menyediakan peluru tipe lingkaran
decimal	Gaya default untuk daftar , angka sederhana.

disc	Gaya default untuk daftar , gaya lingkaran yang diisi.
none	Menghapus gaya sepenuhnya untuk daftar.
square	Sebuah peluru persegi.
upper-roman	Angka Romawi huruf besar, seperti dalam garis besar

Menggunakan basis angka yang berbeda

Di mana ada lebih dari sembilan tipe dalam suatu angka, kita harus bekerja di basis angka yang lebih tinggi. Misalnya, kita tidak dapat mengubah bilangan majemuk [11,7,19] menjadi desimal hanya dengan menggabungkan ketiga bagian tersebut. Sebagai gantinya, kita dapat mengonversi angka ke basis yang lebih tinggi seperti basis 20 (atau lebih tinggi jika ada lebih dari 19 jenis apa pun). Untuk melakukan ini, kalikan tiga bagian dan tambahkan hasilnya seperti ini, dimulai dengan angka paling kanan dan bekerja di kiri:

20 × 19 = 380
20×20 × 7 = 2800
20×20×20 × 11 = 88000
Total in decimal = 91180

Di sebelah kiri, ganti nilai 20 dengan basis yang kita gunakan. Setelah semua bilangan majemuk dari seperangkat aturan dikonversi dari basis ini ke desimal, mudah untuk menentukan spesifisitas, dan oleh karena itu didahulukan, dari masing-masing. Untungnya, prosesor CSS menangani semua ini untuk Anda, tetapi mengetahui cara kerjanya membantu kita membuat aturan dengan benar dan memahami prioritas apa yang akan mereka miliki.

Mengubah gaya poin

Cara terbaik untuk melihat gaya ini beraksi adalah dengan mencobanya.

1. Buka teks Editor
2. Ubah atau buat ul.html.
   Di dalam file, gunakan HTML berikut. Jika menggunakan ul.html, kita hanya perlu menambahkan elemen <link> untuk memasukkan file CSS.

```
<!doctype html>
<html>
<head>
<title>An unordered list</title>
<link rel=”stylesheet” type=”text/css” href=”ul.css”>
</head>
<body>
<ul>
 <li>Pine</li>
 <li>Oak</li>
 <li>Elm</li>
</ul>
</body>
</html>
```

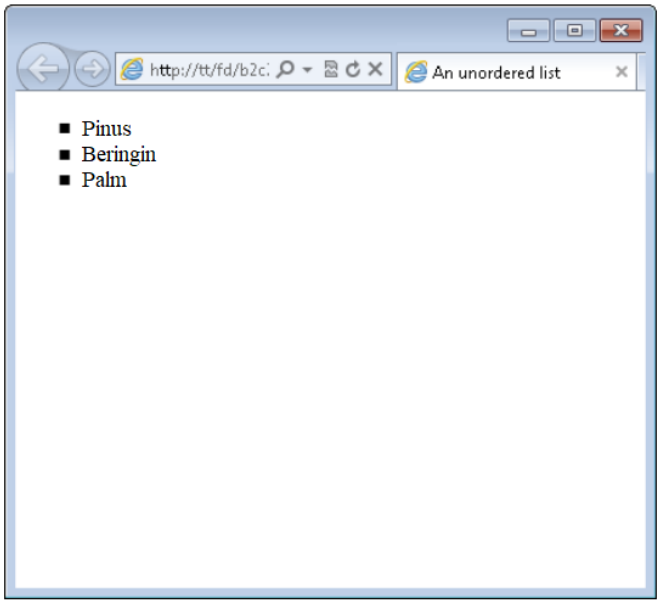
3. Simpan file.

Simpan file sebagai ul.html di root dokumen Anda.

4. Buat file baru.
   Buat dokumen teks kosong baru menggunakan text editor.
5. Tempatkan CSS berikut di dokumen baru:

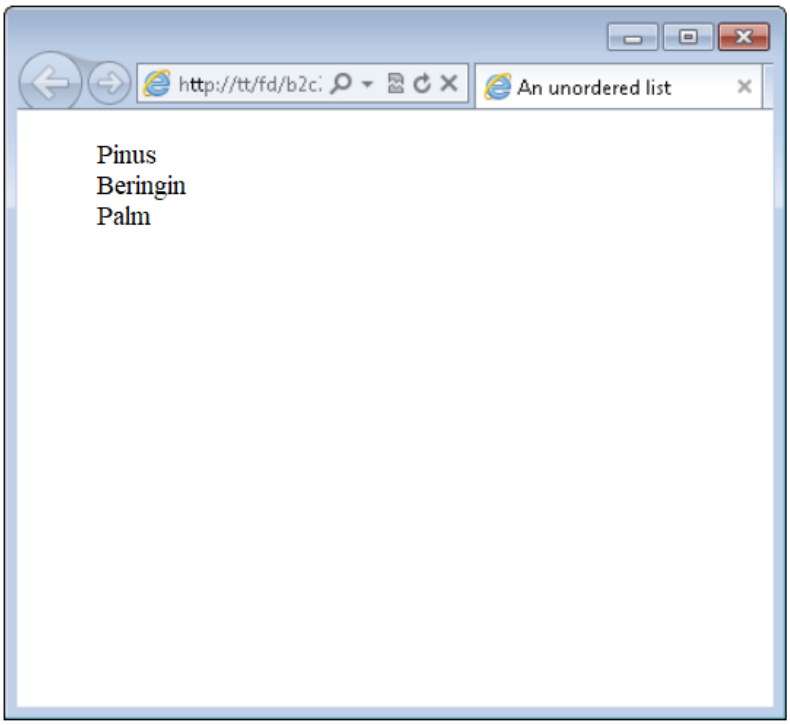
```
ul {
 list-style-type: square;
}
```

6. Simpan file CSSnya.
   Simpan file sebagai ul.css di dokumen root Anda.
7. Buka browser web kita dan lihat halamannya.
   Di browser web, ketik http://localhost/ul.html ke bilah alamat dan tekan Enter. Selanjutnya, kita akan melihat halaman seperti pada gambar dibawah ini.



Gambar 2.21 Mengubah list style.

Dengan properti list style kita dapat bereksperimen untuk menambah atau mengubah gaya peluru.



Gambar 2.22 Menghapus poin dari daftar HTML.

Menghapus poin

Tampilan umum untuk daftar di halaman web tidak menggunakan poin sama sekali. Efek ini dibuat dengan menyetel nilai tipe list style ke none, seperti dalam contoh ini, yang dapat digunakan dalam file ul.css yang baru saja di buat.

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

Ketika diterapkan ke halaman yang dibuat pada latihan sebelumnya, hasilnya terlihat seperti gabar dibawah ini. kita menerapkan properti tipe-daftar ke <ul> atau <ol> dan bukan ke item daftar individual (elemen).

Beberapa aturan lebih setara dari yang lain

Jika dua atau lebih aturan gaya sama persis, hanya aturan yang paling baru diproses yang akan diprioritaskan. Namun, kita dapat memaksakan aturan ke prioritas yang lebih tinggi daripada aturan lain yang setara menggunakan deklarasi !important, seperti ini:

```
p { color:#ff0000 !important; }
```

Saat kita melakukan ini, semua pengaturan sebelumnya yang setara akan ditimpa (termasuk yang menggunakan !important) dan semua aturan yang setara yang diproses nanti akan diabaikan. Jadi, misalnya, yang kedua dari dua aturan berikut biasanya akan didahulukan, tetapi karena penggunaan !important dalam penetapan sebelumnya, yang kedua diabaikan:

```
p { color:#ff0000 !important; }
p { color:#ffff00 }
```

Catatan: Style sheet pengguna dapat dibuat untuk menentukan gaya browser default, dan mereka dapat menggunakan deklarasi !important, dalam hal ini pengaturan user style akan didahulukan dari properti yang sama yang ditentukan di halaman web saat ini. Namun, pada browser yang sangat lama menggunakan CSS 1, fitur ini tidak didukung

2.16 MENAMBAHKAN BACKGROUND

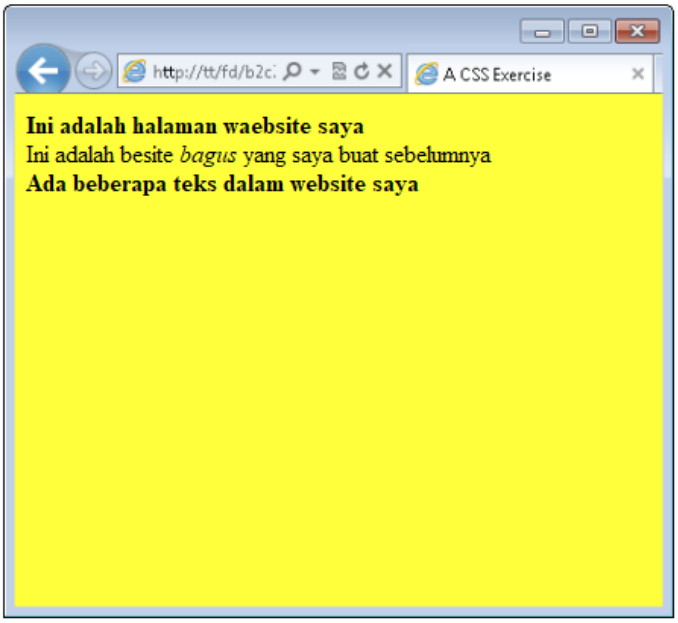
Halaman yang telah kita buat sejauh ini memiliki latar belakang putih, atau lebih tepatnya, mereka memiliki latar belakang default yang dipilih oleh browser. Di browser web versi lama, warna latar belakang itu abu-abu. kita dapat mengubah warna latar belakang menggunakan CSS, atau menggunakan gambar latar belakang. Warna latar belakang dan gambar latar belakang dapat diterapkan ke seluruh halaman atau ke elemen individual. Mengubah warna latar belakang pada elemen individual membantu menambahkan sorotan dan warna ke area halaman tertentu.

Mengubah warna background

Warna latar belakang elemen HTML diubah dengan properti CSS warna latar belakang. Warna latar belakang menggunakan sintaks yang sama (kode hex) sebagai warna font; lihat pembahasan warna font sebelumnya dalam bab ini untuk melihat kode hex untuk warna umum. Berikut adalah contoh yang mengubah warna latar belakang seluruh halaman:

```
body {
 background-color: #FFFF00;
}
```

Gambar dibawah ini menunjukkan halaman yang dihasilkan. Perhatikan bahwa warna kuning tidak akan muncul dengan baik di dalam buku, tetapi ada di sana!



Gambar 2.23 Menambahkan warna latar belakang kuning ke halaman

Seperti yang dinyatakan sebelumnya, elemen individual juga dapat diubah dan kita dapat menggunakan semua selector CSS yang berbeda untuk memfokuskan perubahan warna itu ke kelas, ke elemen individual (menggunakan id), atau ke semua elemen dengan

menggunakan nama elemen. Misalnya, mengubah semua elemen menjadi kuning terlihat seperti ini:

```
div {
 background-color: #FFFF00;
}
```

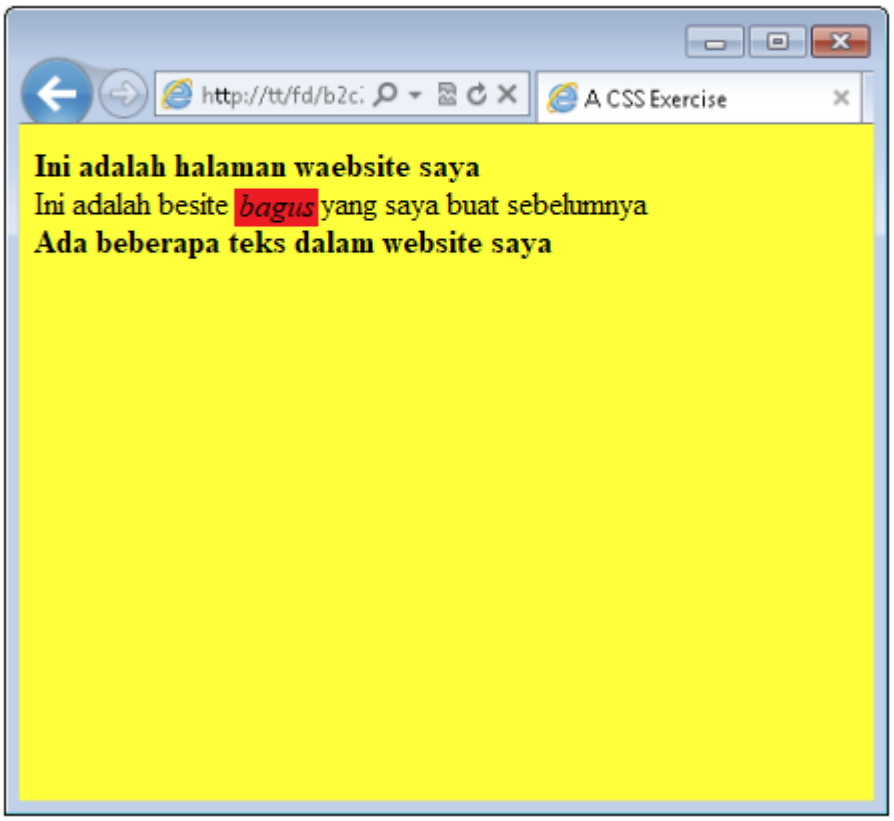
Anda juga dapat menggunakan CSS untuk menargetkan elemen berdasarkan hierarkinya; dengan kata lain, kita dapat menargetkan elemen ketika mereka muncul sebagai anak dari elemen lain. Ini membutuhkan contoh. Banyak contoh dalam buku ini menggunakan HTML yang mirip dengan yang ditunjukkan pada contoh.

Contoh 4 HTML Digunakan dalam Beberapa Contoh

```
<!doctype html>
<html>
<head>
<title>A CSS Exercise</title>
<link rel="stylesheet" type="text/css" href="style8.css">
</head>
<body>
<div class="boldText">This is my web page.</div>
<div>
 This is the <span>nicest</span> page I've made yet.
</div>
<div class="boldText">Here is some text on my page.</div>
</body>
</html>
```

Fokus pada elemen di dalam kedua dalam HTML ini. Bisa dibilang elemen adalah anak dari . Dengan menggunakan CSS, kita dapat menargetkan span ini berdasarkan posisinya sebagai anak dari . Ini berguna jika kita ingin menerapkan gaya tertentu ke semua elemen dari tipe tertentu tetapi kita tidak (atau tidak bisa) menambahkan kelas ke elemen tersebut. Misalnya, jika kita ingin membuat semua elemen yang muncul dalam memiliki latar belakang merah, kita dapat melakukannya dengan CSS ini:

```
div span {
 background-color: #FF0000;
}
```



Gambar 2.24 Menargetkan elemen untuk menerapkan aturan CSS.

Penargetan CSS ini dapat diterapkan dengan cara apa pun yang kita inginkan, apakah itu menargetkan ID tertentu, kelas tertentu, atau elemen tertentu, seperti contoh. kita dapat membuat kombinasi hierarki CSS yang kuat (dan terkadang membingungkan) untuk menerapkan aturan CSS. kita dapat menggunakan penargetan CSS ini untuk menerapkan aturan CSS apa pun, bukan hanya warna latar belakang.

2.17 WARNA CSS

Anda dapat menerapkan warna ke latar depan dan latar belakang teks dan objek menggunakan properti warna dan warna latar (atau dengan menyediakan satu argumen ke properti latar belakang). Warna yang ditentukan dapat berupa salah satu warna bernama (seperti merah atau biru), warna yang dibuat dari triplet RGB heksadesimal (seperti #ff0000 atau #0000ff), atau warna yang dibuat menggunakan fungsi CSS rgb. 16 nama warna standar seperti yang didefinisikan oleh organisasi standar W3C adalah: aqua, hitam, biru, fuchsia, abu-abu, hijau, kapur, merah marun, biru tua, zaitun, ungu, merah, perak, teal, putih, dan kuning. Aturan berikut menggunakan salah satu dari nama ini untuk mengatur warna latar belakang untuk objek dengan ID objek:

```
#object { background-color:silver; }
```

Dalam aturan ini, warna latar depan teks di semua elemen diatur ke kuning (karena pada tampilan komputer, tingkat heksadesimal ff merah, ditambah ff hijau, ditambah 00 biru menghasilkan warna kuning):

```
div { color:#ffff00; }
```

Atau, jika kita tidak ingin bekerja dalam heksadesimal, kita dapat menentukan triplet warna kita menggunakan fungsi rgb, seperti dalam aturan berikut, yang mengubah warna latar belakang dokumen saat ini menjadi aqua:

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

Catatan: Jika kita memilih untuk tidak bekerja dalam rentang 256 level per warna, kita dapat menggunakan persentase dalam fungsi rgb sebagai gantinya, dengan nilai dari 0 hingga 100 mulai dari terendah (0) hingga tertinggi (100) jumlah primer warna, seperti ini: rgb (58%, 95%, 74%). kita juga dapat menggunakan nilai floating point untuk kontrol warna yang lebih halus, seperti ini: rgb(23,4%, 67,6%, 15,5%).

String Warna Pendek

Ada juga bentuk pendek dari string digit hex di mana hanya yang pertama dari setiap pasangan 2-byte yang digunakan untuk setiap warna. Misalnya, alih-alih menetapkan warna #fe4692, kita malah menggunakan #f49, menghilangkan digit heksagonal kedua dari setiap pasangan, yang setara dengan nilai warna #ff4499. Ini menghasilkan warna yang hampir sama dan berguna di mana warna yang tepat tidak diperlukan. Perbedaan antara string enam digit dan tiga digit adalah bahwa yang pertama mendukung 16 juta warna berbeda, sedangkan yang kedua mendukung empat ribu. Di mana pun kita ingin menggunakan warna seperti #883366, ini adalah padanan langsung dari #836 (karena digit berulang tersirat oleh versi yang lebih pendek), dan kita dapat menggunakan salah satu string untuk membuat warna yang sama persis.

Menambahkan gambar background

Gambar background adalah cara yang baik untuk membuat halaman HTML agar terlihat bagus. Dalam penggunaan gambar background, kita dapat membuat efek gradien, di mana satu bagian halaman berwarna solid dan warnanya memudar atau menjadi lebih terang saat membentang ke sisi lain. Gambar background muncul di belakang elemen lainnya. Ini berarti kita dapat melapisi semua HTML Anda, termasuk gambar lain, di atas gambar latar belakang. kita dapat menemukan banyak gambar gratis melalui *Creative Commons*. Lihat http://search.creativecommons.org untuk informasi lebih lanjut. Pastikan untuk memilih gambar yang masih memungkinkan teks dapat dibaca di halaman; teks hitam pada gambar gelap tidak cocok. Gambar latar belakang ditambahkan dengan properti CSS gambar latar, seperti yang dijelaskan di sini dan di bagian berikut.

```
background-image:url(“myImage.jpg”);
```

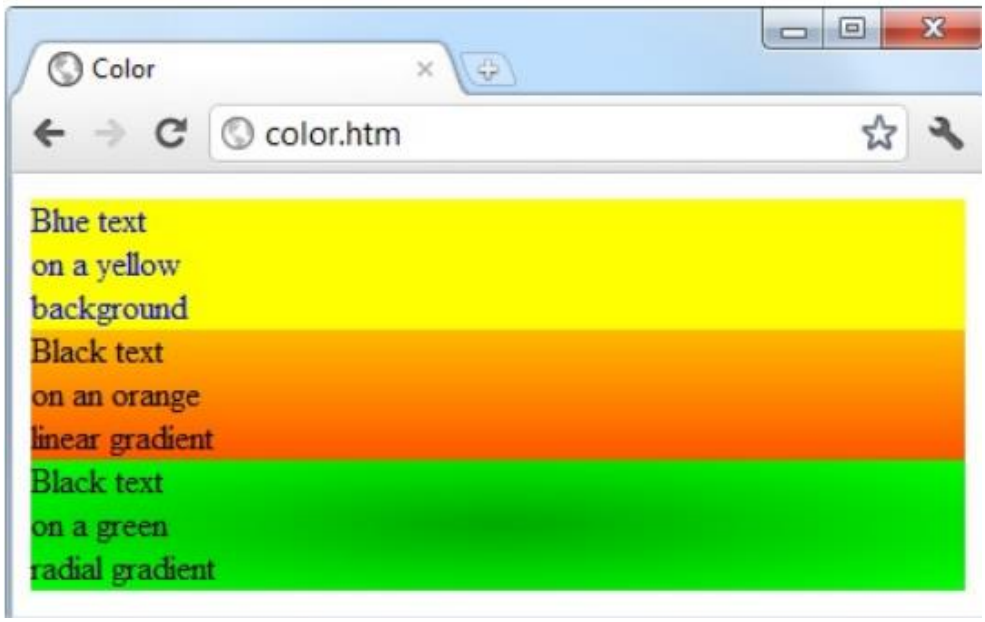
Gradien

Di tempat menggunakan warna latar belakang yang solid, kita dapat memilih untuk menerapkan gradien, yang kemudian akan secara otomatis mengalir dari warna awal yang diberikan ke warna akhir pilihan Anda. Paling baik digunakan bersama dengan aturan warna sederhana sehingga browser yang tidak mendukung gradien setidaknya akan menampilkan warna solid. Contoh 5 menggunakan aturan untuk menampilkan gradien oranye (atau hanya oranye polos pada browser yang tidak mendukung) seperti yang ditunjukkan di bagian tengah Gambar 2.25.

Contoh 5. Membuat gradien linier

```
<!DOCTYPE html>
<html>
<head>
<title>Creating a linear gradient</title>
<style>
.orangegrad {
background:orange;
background:linear-gradient(top, #fb0, #f50);
background:-moz-linear-gradient(top, #fb0, #f50);
background:-webkit-linear-gradient(top, #fb0, #f50);
background:-o-linear-gradient(top, #fb0, #f50);
background:-ms-linear-gradient(top, #fb0, #f50); }
</style>
</head>
<body>
<div class='orangegrad'>Black text<br>
on an orange<br>linear gradient</div>
</body>
</html>
```

Catatan: Seperti yang ditunjukkan pada contoh sebelumnya, banyak aturan CSS memerlukan awalan khusus browser seperti -moz-, -webkit-, -o-, dan -ms- (untuk browser berbasis Mozilla seperti Firefox; browser berbasis WebKit seperti seperti Apple Safari, Google Chrome, dan browser iOS dan Android; dan browser Opera dan Microsoft). http://caniuse.com mencantumkan aturan dan atribut CSS utama, dan apakah versi khusus browser diperlukan.



Gambar 2.25 Warna latar belakang yang solid, gradien linier, dan gradien radial

Untuk membuat gradien, pilih di mana itu akan dimulai dari atas, bawah, kiri, kanan, dan tengah (atau kombinasi apa pun, seperti kiri atas atau kanan tengah), masukkan warna awal dan akhir yang kita butuhkan, lalu terapkan salah satu gradien linier atau aturan gradien

radial, pastikan kita juga menyediakan aturan untuk semua browser yang kita targetkan. kita juga dapat menggunakan lebih dari sekadar warna awal dan akhir dengan juga menyediakan apa yang disebut warna berhenti di antaranya sebagai argumen tambahan. Dalam kasus ini, misalnya, jika lima argumen diberikan, setiap argumen akan mengontrol perubahan warna pada seperlima area yang diwakili oleh lokasinya dalam daftar argumen.

2.18 ELEMEN PEMOSISIAN

Elemen dalam halaman web berada di tempatnya ditempatkan dalam dokumen, tetapi kita dapat memindahkannya dengan mengubah properti posisi elemen dari default statis menjadi mutlak, relatif, atau tetap.

Pemosisian Absolut

Elemen dengan pemosisian absolut dihapus dari dokumen, dan elemen lain apa pun yang mampu akan mengalir ke ruang bebasnya. kita kemudian dapat memposisikan objek di mana pun kita suka di dalam dokumen menggunakan properti atas, kanan, bawah, dan kiri. Kemudian akan beristirahat di atas (atau di belakang) elemen lainnya. Jadi, misalnya, untuk memindahkan objek dengan ID objek ke lokasi absolut 100 piksel ke bawah dari awal dokumen dan 200 piksel ke dalam dari kiri, kita akan menerapkan aturan berikut (Anda juga dapat menggunakan salah satu dari unit pengukuran lain yang didukung oleh CSS):

```
#object {
position:absolute;
top :100px;
left :200px;
}
```

Pemosisian Relatif

Demikian juga, kita dapat memindahkan objek relatif ke lokasi yang akan ditempatinya dalam aliran dokumen normal. Jadi, misalnya, untuk memindahkan objek 10 piksel ke bawah dan 10 piksel ke kanan dari lokasi normalnya, kita akan menggunakan aturan berikut:

```
#object {
position:relative;
top :10px;
left :10px;
}
```

Pemosisian Tetap

Pengaturan properti pemosisian akhir memungkinkan kita memindahkan objek ke lokasi absolut, tetapi hanya di dalam viewport browser saat ini. Kemudian, saat dokumen di-scroll, objek tetap berada di tempatnya, dengan dokumen utama di-scroll di bawahnya—cara yang bagus untuk membuat dock bar dan perangkat serupa lainnya. Untuk memperbaiki objek ke sudut kiri atas jendela browser, kita akan menggunakan aturan berikut:

```
#object {
position:fixed;
top :0px;
```

```
left :0px;
}
```

Pada Gambar 2.26, Contoh 6 telah dimuat ke dalam browser, dan lebar dan tinggi browser telah dikurangi sehingga kita harus menggulir ke bawah untuk melihat semua halaman web.

Gambar 2.26 Menggunakan nilai pemosisian yang berbeda

Ketika ini selesai, segera terlihat bahwa elemen dengan posisi tetap tetap di tempatnya bahkan melalui pengguliran. kita juga dapat melihat bahwa elemen dengan pemosisian absolut terletak tepat pada 100 piksel ke bawah, dengan 0 offset horizontal, sedangkan elemen dengan pemosisian relatif sebenarnya dipindahkan ke atas sebesar 8 piksel dan kemudian diimbangi dari margin kiri sebesar 110 piksel untuk membuat garis di samping elemen pertama.

Contoh 6 Menerapkan nilai pemosisian yang berbeda

```
<!DOCTYPE html>
<html>
<head>
<title>Positioning</title>
<style>
#object1 {
position :absolute;
background:pink;
width :100px;
height :100px;
top :100px;
left :0px;
}
#object2 {
position :relative;
background:lightgreen;
width :100px;
height :100px;
top :-8px;
left :110px;
```

```
}
#object3 {
position :fixed;
background:yellow;
width :100px;
height :100px;
top :100px;
left :236px;
}
</style>
</head>
<body>
<br><br><br><br><br>
<div id='object1'>Absolute Positioning</div>
<div id='object2'>Relative Positioning</div>
<div id='object3'>Fixed Positioning</div>
</body>
</html>
```

Pada gambar, elemen dengan posisi tetap awalnya sejajar dengan dua elemen lainnya, tetapi tetap berada di tempatnya sementara yang lain telah digulir ke atas halaman, dan sekarang muncul offset di bawahnya.

2.19 PSEUDECLASS

Ada sejumlah selektor dan kelas yang digunakan hanya dalam style sheet dan tidak memiliki tag atau atribut yang cocok dalam HTML apa pun. Tugas mereka adalah mengklasifikasikan elemen menggunakan karakteristik selain nama, atribut, atau kontennya — yaitu, karakteristik yang tidak dapat disimpulkan dari pohon dokumen. Ini termasuk pseudoclasses seperti link dan dikunjungi. Ada juga Pseudo-element yang membuat pilihan, yang mungkin terdiri dari elemen parsial seperti baris pertama atau huruf pertama. *Pseudeclass* dan *Pseudo-element* dipisahkan oleh karakter : (titik dua). Misalnya, untuk membuat kelas yang disebut bigfirst untuk menekankan huruf pertama dari suatu elemen, kita akan menggunakan aturan seperti berikut:

```
.bigfirst:first-letter {
font-size:400%;
float :left;
}
```

Ketika kelas bigfirst diterapkan pada suatu elemen, huruf pertama akan ditampilkan jauh lebih besar, dengan teks yang tersisa ditampilkan pada ukuran normal, mengalir dengan rapi di sekitarnya (karena properti float) seolah-olah huruf pertama adalah gambar atau objek lain . Pseudoclasses termasuk hover, link, aktif, dan dikunjungi, yang semuanya sebagian besar berguna untuk menerapkan elemen jangkar, seperti dalam aturan berikut, yang mengatur warna default semua link ke biru, dan link yang telah dikunjungi ke biru muda:

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

Aturan berikut ini menarik karena mereka menggunakan pseudoclass hover sehingga hanya diterapkan ketika mouse ditempatkan di atas elemen. Dalam contoh ini, mereka mengubah tautan menjadi teks putih dengan latar belakang merah, memberikan efek dinamis yang biasanya hanya kita harapkan dari penggunaan kode JavaScript:

```
a:hover {
color :white;
background:red;
}
```

Di sini saya telah menggunakan properti background dengan satu argumen, alih-alih properti background-color yang lebih panjang. Pseudoclass aktif juga dinamis karena mempengaruhi perubahan pada tautan selama waktu antara tombol mouse diklik dan dilepaskan, seperti aturan ini, yang mengubah warna tautan menjadi biru tua:

```
a:active { color:darkblue; }
```

Pseudoclass dinamis lainnya yang menarik adalah fokus, yang diterapkan hanya ketika sebuah elemen diberikan fokus oleh pengguna yang memilihnya dengan keyboard atau mouse. Aturan berikut menggunakan selektor universal untuk selalu menempatkan batas 2-piksel pertengahan abu-abu, putus-putus di sekitar objek yang saat ini difokuskan:

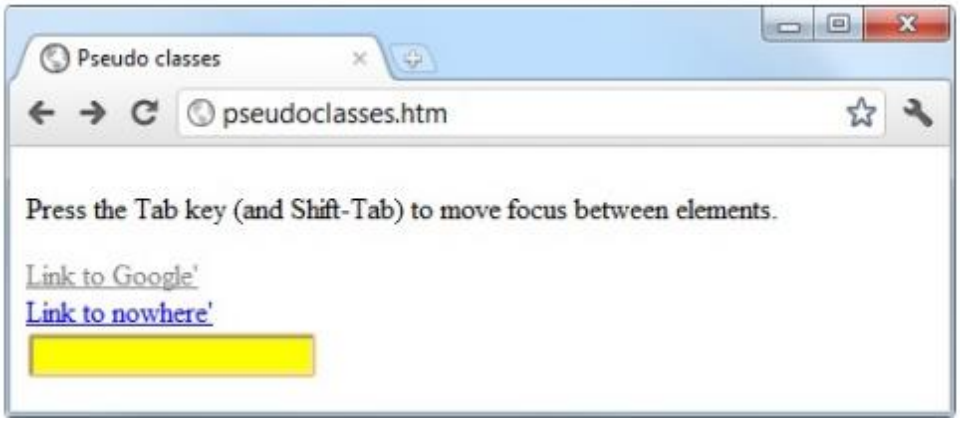
```
*:focus { border:2px dotted #888888; }
```

Contoh 7 menampilkan dua link dan sebuah field input, seperti yang ditunjukkan pada Gambar 2.27. Tautan pertama muncul sebagai abu-abu karena telah dikunjungi di browser ini, tetapi tautan kedua belum dan ditampilkan dengan warna biru. Tombol Tab telah ditekan dan fokus input sekarang menjadi bidang input, sehingga latar belakangnya berubah menjadi kuning. Ketika salah satu tautan diklik, itu akan ditampilkan dalam warna ungu, dan ketika diarahkan ke atasnya akan tampak merah.

Contoh 7 Tautkan dan fokuskan pseudoclass

```
<!DOCTYPE html>
<html>
<head>
<title>Pseudoclasses</title>
<style>
a:link { color:blue; }
a:visited { color:gray; }
a:hover { color:red; }
a:active { color:purple; }
*:focus { background:yellow; }
</style>
</head>
```

```
<body>
<a href='http://google.com'>Link to Google'</a><br>
<a href='nowhere'>Link to nowhere'</a><br>
<input type='text'>
</body>
</html>
```



Gambar 2.27 Pseudoclass diterapkan pada pilihan elemen

Pseudeclass lainnya juga tersedia, dan kita bisa mendapatkan informasi lebih lanjut tentangnya di http://tinyurl.com/pseudoclasses.

Catatan: Hati-hati dalam menerapkan pseudoclass fokus ke selektor universal, *, seperti yang ditunjukkan dalam contoh ini; Internet Explorer menganggap dokumen yang tidak fokus sebagai benar-benar memiliki fokus yang diterapkan ke seluruh halaman web, dan (dalam hal ini) seluruh halaman akan berubah menjadi kuning hingga Tab ditekan atau fokus diterapkan ke salah satu elemen usia.

Menambahkan satu gambar background

Salah satu fitur gambar background adalah kita dapat memasang atau mengulanginya dalam satu halaman. Artinya, seberapa besar ukuran layar pengunjung website, gambar backgrounf akan selalu muncul. Sebalikny, kita juga dapat memilih untuk tidak mengulangi gambar background. Bagian ini menunjukkan cara menambahkan gambar tunggal yang tidak berulang. Untuk menyelesaikan latihan ini, kita memerlukan gambar. Gambar sebaiknya paling sedikit 800 piksel kali 600 piksel. kita dapat mengetahui resolusi gambar dengan mengklik kanan gambar dan memilih Properties di Windows atau memilih Get Info dari jendela Finder di Mac.

1. Buka text editor
   Buat dokumen teks kosong baru di text editor.
2. Mmasukkan HTML berikut

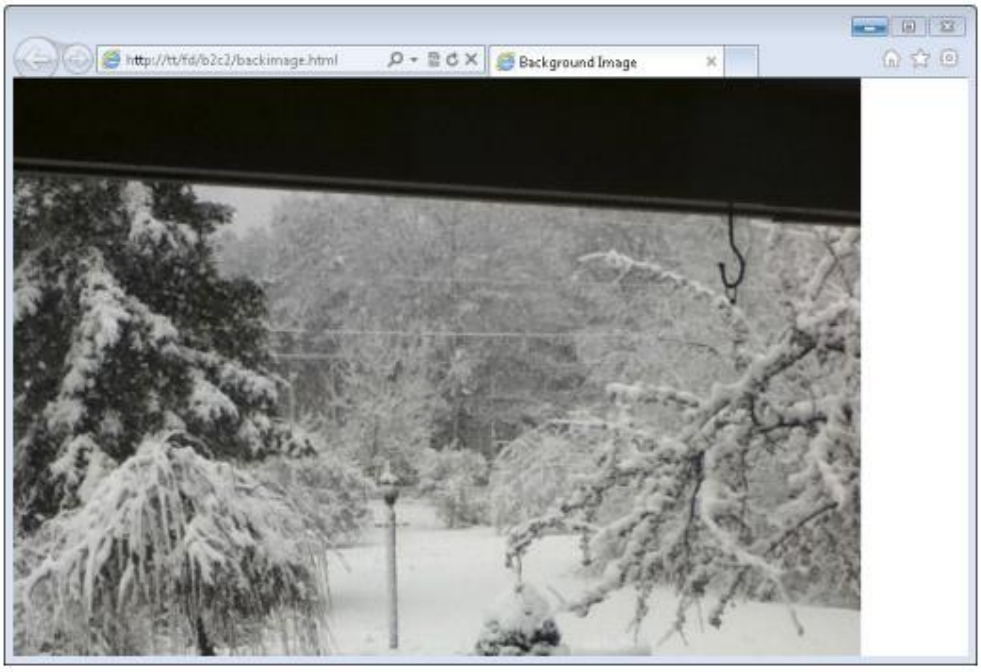
```
<!doctype html>
<html>
<head>
<title>Background Image</title>
<link rel=”stylesheet” type=”text/css”
href=”image-style.css”>
</head>
```

```
<body>
</body>
</html>
```

3. Simpan file.
   Simpan file sebagai backimage.html di root dokumen Anda.
4. Buat dokumen teks baru.
   Buat dokumen teks kosong baru dengan text editor.
5. Tempatkan CSS berikut di dokumen baru.
   Pastikan untuk menggunakan nama gambar Anda. Dalam contoh ini, kami menggunakan gambar yang disebut large-snow.jpg. Gambar harus disimpan di dalam root dokumen Anda.

```
body {
 background-image:url("large-snow.jpg");
 background-repeat: no-repeat;
}
```

6. Simpan file CSS.
   Simpan file sebagai image-style.css dan pastikan itu disimpan di dalam root dokumen Anda.
7. Buka browser web kita dan lihat halamannya.
   Buka browser web kita dan navigasikan ke halaman di http://localhost/backimage.html. nda akan melihat halaman dengan gambar latar belakang. kita dapat melihat tangkapan layar halaman kami, dengan gambar large-snow.jpg, pada gambar berikut.



Gambar 2.28 Sebuah gambar latar belakang tunggal.

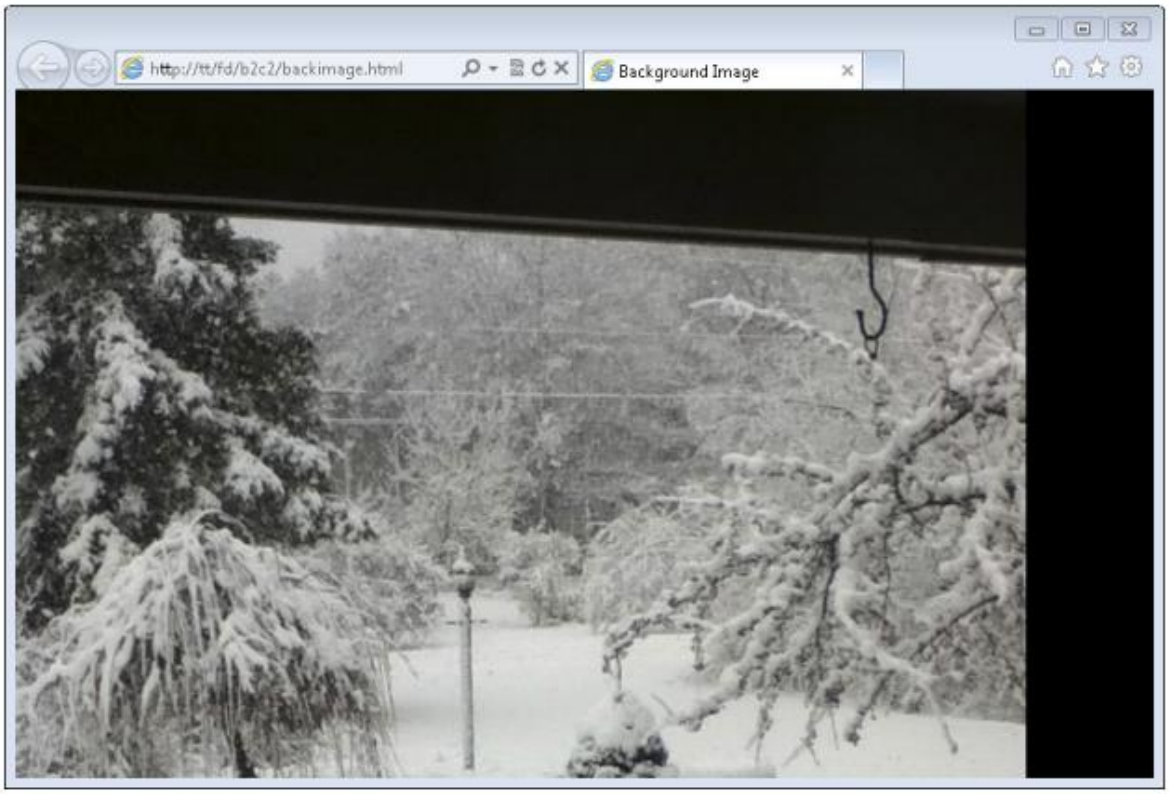
Tergantung pada ukuran gambar dan layar Anda, kita mungkin memperhatikan bahwa gambar berakhir, seperti halnya di sepanjang sisi kanan diatas. Selain itu, kita mungkin memperhatikan bahwa gambar tidak berada di tengah. Baca terus untuk solusinya.

Meningkatkan halaman gambar latar belakang tunggal

Pendekatan umum yang digunakan untuk membuat halaman yang tampak lebih baik adalah dengan menambahkan warna latar belakang yang cocok dengan tepi gambar. Dalam kasus gambar kami, bagian atas dan bawah berwarna hitam. Oleh karena itu, kita dapat menambahkan aturan ke CSS untuk membuat warna latar belakang default menjadi hitam. Ini tidak akan berpengaruh apa pun di mana gambar berada — gambar didahulukan — tetapi itu akan menjadi masalah di sepanjang bagian bawah di mana gambar berakhir. CSS untuk tampilan ini adalah sebagai berikut:

```
body {
 background-image:url(“large-snow.jpg”);
 background-repeat: no-repeat;
 background-color: #000000;
}
```

Dengan aturan itu, gambar akan tetap berakhir tetapi penampilannya tidak akan terlalu mengejutkan atau mencolok karena cocok dengan warna tepi gambar, seperti yang ditunjukkan pada gmbar berikut.



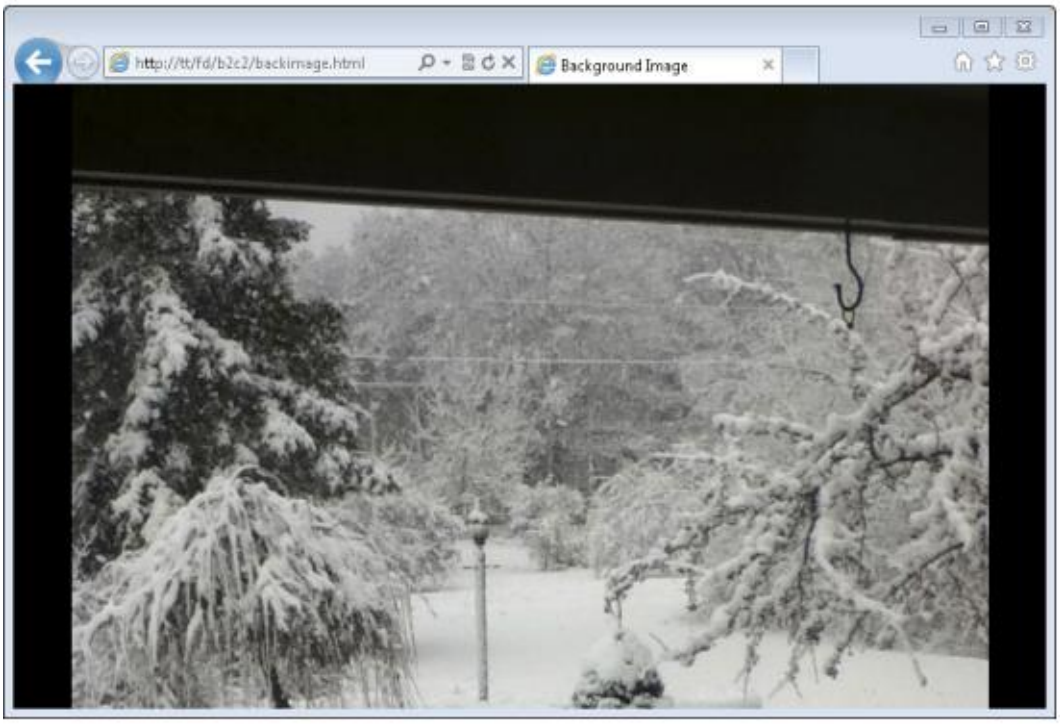
Gambar 2.29 Menambahkan warna latar belakang dan gambar background.

Sementara trik warna latar belakang memecahkan masalah dengan tepi gambar, itu tidak menyelesaikan masalah pemusatan. Gambar latar belakang saat ini diterapkan ke badan

— dengan kata lain, seluruh halaman. Untuk memusatkan gambar latar belakang, properti CSS lain perlu ditambahkan, seperti yang ditunjukkan dalam CSS ini:

```
body {
 background-image:url(“large-snow.jpg”);
 background-repeat: no-repeat;
 background-color: #000000;
 background-position: center top;
}
```

CSS ini menambahkan aturan posisi latar belakang dan menempatkannya di tengah di bagian atas halaman. Nilai lainnya termasuk kiri, kanan, dan bawah, dan kita dapat menggabungkannya sehingga gambar latar belakang akan muncul di kanan bawah, misalnya. CSS yang ditampilkan di sini menempatkan gambar di tengah halaman dan di atas. Ini menghasilkan halaman yang ditunjukkan pada Gambar berikut.



Gambar 2.30 Gambar latar tengah di bagian atas, dengan warna latar belakang.

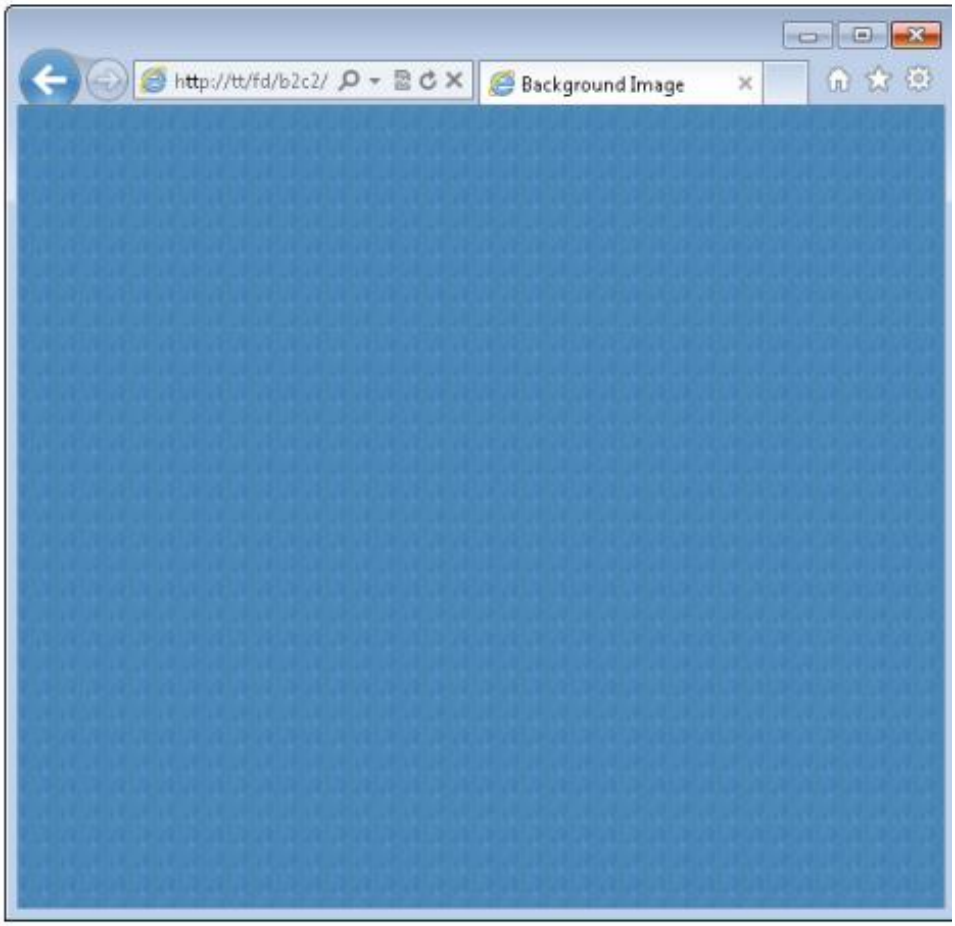
Dengan gambar itu di tempatnya, kita kemudian dapat menambahkan HTML apa pun ke halaman yang kita inginkan. Perhatikan dengan gambar seperti ini (bagian atas gelap dan tengah terang) kita perlu menyesuaikan warna font agar teks terlihat di halaman.

2.20 MENAMBAHKAN GAMBAR BERULANG

Anda dapat menambahkan gambar yang berulang. Ini adalah skenario umum untuk halaman web karena gambar tidak berakhir di sepanjang sisi, tidak peduli seberapa besar resolusi Anda. Ini juga mengurangi kebutuhan akan posisi latar belakang karena gambar latar berlaku untuk seluruh elemen. Saat diterapkan ke seluruh halaman, seperti pada contoh yang ditampilkan, kita juga dapat mengabaikan aturan pengulangan latar belakang dan warna latar belakang karena gambar berlanjut di seluruh halaman. Gambar berulang yang ideal adalah

gambar yang tidak memiliki batas yang terlihat karena batas tersebut akan muncul saat gambar dipetakan atau diulang pada halaman. Gambar 2.31 menunjukkan gambar kecil (15 piksel x 15 piksel) yang digunakan sebagai gambar berulang dengan CSS berikut:

```
body {
 background-image:url(“back.jpg”);
}
```



Gambar 2.31 Gambar latar berulang.

Seperti pada contoh untuk latar belakang gambar tunggal, kita sekarang dapat menambahkan HTML di atas latar belakang, sekali lagi memilih warna font yang mengimbangi gambar sehingga pengunjung dapat dengan mudah membaca teks.

2.21 MEMBUAT LAYOUT HALAMAN

Anda sekarang telah mempelajari banyak CSS untuk mengubah perilaku dan tampilan setiap item, menambahkan warna latar belakang, daftar gaya, dan sebagainya. Semua ini mengarah pada pembuatan halaman dengan menggunakan CSS. CSS digunakan untuk membuat tampilan halaman web yang lebih kompleks daripada yang kita lihat sejauh ini. Misalnya, kita dapat membuat efek kolom, di mana ada menu di sisi kiri atau kanan dan konten di kolom lain, dan kami memberi tahu kita cara melakukannya di sini.

Saat bekerja dengan perataan dan layout kolom, terkadang berguna untuk menambahkan batas ke elemen untuk melihat di mana itu dimulai dan berakhir sehingga kita bisa melihat tampilan layout.

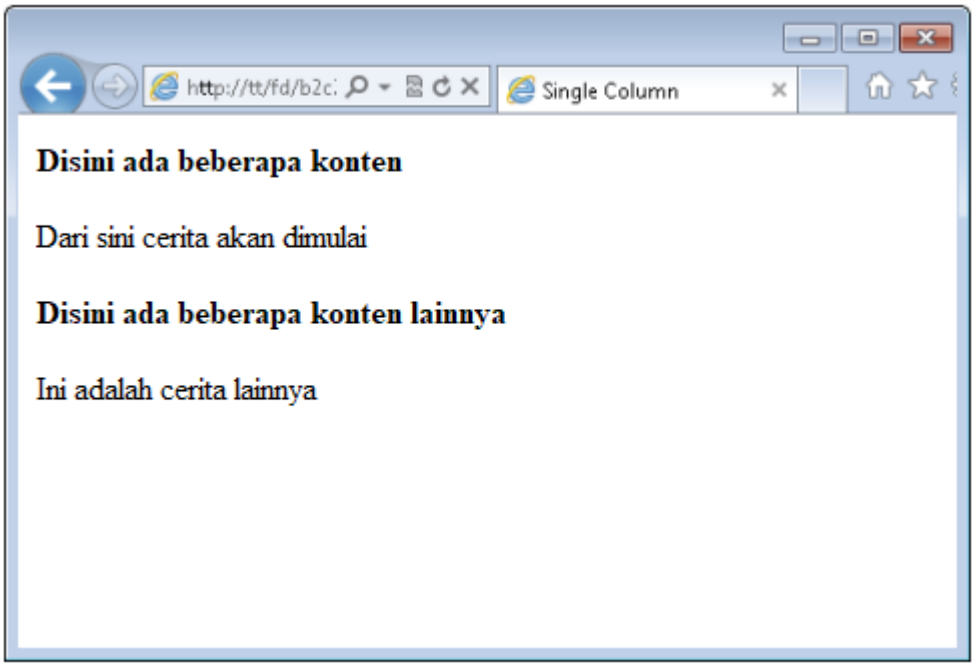
Membuat layout satu kolom

Semua yang kita lihat sejauh ini adalah layout satu kolom. Hanya ada satu kolom, sejajar di sebelah kiri halaman. Namun, kita dapat mengontrol penyelarasan itu dengan CSS. Melakukannya berarti membuat HTML yang lebih kompleks daripada yang kita lihat sejauh ini, tetapi tidak ada yang baru di HTML; hanya akan ada lebih banyak HTML daripada sebelumnya.

1. Buka text editor
2. Di dalam dokumen kosong, masukkan HTML berikut:

```
<!doctype html>
<html>
<head>
<title>Single Column</title>
<link rel="stylesheet" type="text/css" href="single.
css">
</head>
<body>
<div id="container">
 <div id="content">
 <h2>Disini ada beberapa konten</h2>
 <p>Dari sini cerita akan dimulai go</p>
 <h2>Disini ada beberapa konten lainnya</h2>
 <p>Ini adalah cerita lainnya</p>
 </div> <!-- end content -->
</div> <!-- end container -->
</body>
</html>
```

3. Simpan file
   Simpan file sebagai single.html pada dokumen root.
4. Buka browser untuk menampilkan halaman
   Lihat halaman dengan membuka http://localhost/single.html di browser Anda. kita akan melihat halaman seperti gambar berikut.

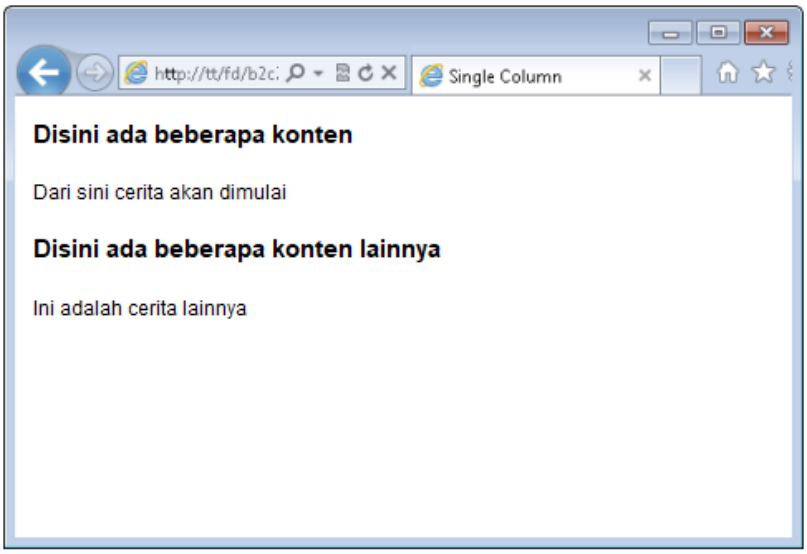


Gambar 2.32 Halaman dasar sebelum menambahkan CSS

5. Buat dokumen teks baru.
   Buat dokumen teks kosong baru di text editor.
6. Dalam dokumen, tempatkan CSS berikut

```
body {
 font-family: arial,helvetica,sans-serif;
}
#container {
 margin: 0 auto;
 width: 600px;
 background: #FFFFFF;
}
#content {
 clear: left;
 padding: 20px;
}
```

7. Simpan file CSS.
   Simpan file sebagai single.css di root dokumen Anda.
8. Buka browser web Anda.
   Arahkan ke http://localhost/single.html di browser Anda. Jika browser kita masih terbuka, muat ulang halaman dengan Ctrl+R di Windows atau Command+R di Mac. kita akan melihat halaman seperti itu pada Gambar dibawah ini. Lihat paragraf berikut untuk informasi lebih lanjut tentang modifikasi spesifik apa yang kita buat di Langkah 6.



Gambar 2.33 Satu layout kolom.

Untuk layout HTML ini menggunakan elemen sebagai wadah. Wadah membantu membuat layout dan tidak menyimpan konten teks apa pun sendiri. CSS untuk latihan ini menggunakan tiga properti CSS width, margin, dan clear. Cara kerjanya adalah sebagai berikut:

- **width**: Mengatur lebar horizontal dari sebuah elemen. Dalam hal ini, wadah diatur ke lebar 600px (piksel). Tidak peduli seberapa kecil jendela browser, HTML kita tidak akan pernah lebih kecil dari 600px.
- **margin**: Ini adalah pelengkap dari properti padding yang ditunjukkan sebelumnya dalam bab ini, di bagian "Adding Border". Properti margin mendefinisikan jarak di luar elemen. Dalam kasus yang ditunjukkan di sini (margin: 0 auto;), metode pintasan digunakan. Lihat bilah sisi untuk informasi lebih lanjut. Nilai "auto" berarti browser akan memilih nilainya.
- **clear**: Membuatnya agar tidak ada elemen yang muncul di sisi elemen yang aturannya berlaku. Dalam contoh, clear left digunakan pada dengan id #content. Ini berarti tidak ada yang bisa muncul di sisi kiri elemen itu. Nilai lain untuk clear termasuk "both", "none", "right", dan "inherent".

Dari sini kita dapat bereksperimen dengan margin jendela browser untuk melihat bagaimana layout yang dibuat dapat bereaksi atau bergerak seiring dengan pergerakan browser. Layout yang dibuat di bagian ini disebut *layout single-column fixed-width*. Pilihan lainnya adalah layout *single-column liquid*. Layout *single-column liquid* dapat bekerja lebih baik di perangkat tertentu. Layout lebar tetap yang ditampilkan terkadang dapat menghasilkan bilah gulir horizontal di bagian bawah halaman. Untuk mengubah layout menjadi layout *single-column liquid*, hanya perlu mengubah sedikit CSS di #container, seperti yang ditunjukkan di sini:

```
body {
 font-family: arial,helvetica,sans-serif;
}
#container {
 margin: 0 30px;
 background: #FFFFFF;
```

```
}
#content {
 clear: left;
 padding: 20px;
}
```

Perhatikan satu-satunya perubahan adalah menghapus properti lebar di dalam #container dan juga mengubah margin dari "0 auto" menjadi "0 30px." Dengan itu, layout menjadi liquid layout dan berfungsi lebih baik, terutama di perangkat seluler.

Aturan Singkatan

Untuk menghemat ruang, grup properti CSS terkait dapat digabungkan menjadi satu tugas singkatan. Misalnya, saya telah menggunakan singkatan untuk membuat batas beberapa kali, seperti dalam aturan fokus di bagian sebelumnya:

```
*:focus { border:2px dotted #ff8800; }
```

Ini sebenarnya adalah rangkaian singkatan dari kumpulan aturan berikut:

```
*:focus {
border-width:2px;
border-style:dotted;
border-color:#ff8800;
}
```

Saat menggunakan aturan singkatan, kita hanya perlu menerapkan properti hingga titik di mana kita ingin mengubah nilai. Jadi kita dapat menggunakan yang berikut ini untuk hanya mengatur lebar dan gaya batas, memilih untuk tidak mengatur warnanya:

```
*:focus { border:2px dotted; }
```

Model dan Layout Kotak

Properti CSS yang memengaruhi layout halaman didasarkan pada model kotak, kumpulan properti bersarang yang mengelilingi elemen. Hampir semua elemen memiliki (atau dapat memiliki) properti ini, termasuk badan dokumen, yang marginnya dapat Anda, misalnya, hapus dengan aturan berikut:

```
body { margin:0px; }
```

Model kotak suatu objek dimulai dari luar, dengan margin objek. Di dalam ini adalah border, lalu ada padding antara border dan isi bagian dalam, dan akhirnya ada isi objek. Setelah kita memahami model kotak, kita akan siap untuk membuat halaman yang ditata secara profesional, karena properti ini saja yang akan membuat banyak gaya halaman Anda.

Pintasan untuk margin dan padding

Daripada menentukan aturan untuk setiap elemen margin atau padding atas, bawah, kanan, dan kiri, kita dapat menggunakan metode pintasan yang mendefinisikan semuanya dalam satu baris. Sebagai contoh:

```
margin: 0px 50px 200px 300px;
setara dengan ini:
margin-top: 0px;
margin-right: 50px;
margin-bottom: 200px;
margin-left: 300px;
```

Ketika empat angka muncul dalam aturan, urutannya adalah atas, kanan, bawah, dan kiri. Untuk membantu mengingat urutannya, gunakan mnemonic "TrouBLe," yang mengambil huruf pertama dari masing-masing Atas, Kanan, Bawah, Kiri, dan membuatnya menjadi sebuah kata untuk mengingatkan kita betapa sulitnya mengingat urutannya. Alih-alih keempat nilai, terkadang kita melihat satu, dua, atau tiga nilai yang ada untuk margin atau padding, seperti pada contoh yang ditunjukkan sebelumnya:

```
margin: 0 auto;
```

Ketika dua nilai digunakan, nilai pertama sesuai dengan atas dan bawah dan nilai kedua sesuai dengan kanan dan kiri. Ketika tiga nilai digunakan, yang pertama adalah bagian atas, yang kedua adalah kiri dan kanan, dan yang terakhir adalah bagian bawah. Akhirnya, ketika satu nilai digunakan, itu berlaku sama untuk atas, kanan, bawah, dan kiri.

Mengatur Margin

Margin adalah tingkat terluar dari model kotak. Ini memisahkan elemen satu sama lain dan penggunaannya cukup cerdas. Misalnya, asumsikan kita telah memilih untuk memberi sejumlah elemen margin default 10 piksel di sekitar masing-masing elemen. Ketika mereka ditempatkan di atas satu sama lain, ini akan menciptakan celah 20 piksel (total lebar border yang berdekatan). CSS mengatasi masalah potensial ini, namun: ketika dua elemen dengan batas diposisikan langsung satu di atas yang lain, hanya yang lebih besar dari dua margin yang digunakan untuk memisahkannya. Jika kedua margin memiliki lebar yang sama, hanya salah satu lebar yang digunakan. Dengan cara ini, kita jauh lebih mungkin untuk mendapatkan hasil yang kita inginkan. Tetapi kita harus memperhatikan bahwa margin elemen yang diposisikan secara mutlak atau sebaris tidak runtuh.

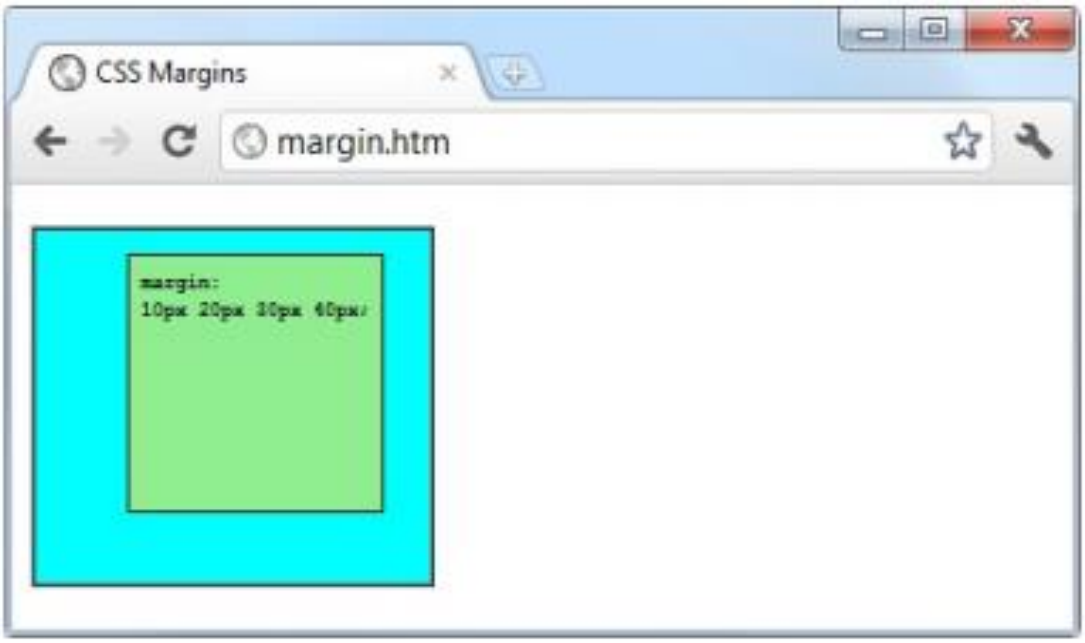
Margin suatu elemen dapat diubah secara massal dengan properti margin, atau secara individual dengan margin-kiri, margin-atas, margin-kanan, dan margin bawah. Saat mengatur properti margin, kita bisa menyediakan satu, dua, tiga, atau empat argumen, yang memiliki efek yang dikomentari dalam aturan berikut:

```
/* Set all margins to 1 pixel
margin:1px;
Set top and bottom to 1 pixel, and left and right to 2
margin:1px 2px;
Set top to 1 pixel, left and right to 2, and bottom to 3
margin:1px 2px 3px;
Set top to 1 pixel, right to 2, bottom to 3, and left to 4 */
margin:1px 2px 3px 4px;
```

Gambar 2.34 menunjukkan Contoh 8 dimuat ke dalam browser, dengan aturan properti margin (disorot dalam huruf tebal) diterapkan pada elemen persegi yang telah ditempatkan di dalam elemen tabel. Tabel tidak diberi dimensi, sehingga hanya akan membungkus elemen bagian dalam sedekat mungkin. Akibatnya, ada margin 10 piksel di atasnya, 20 piksel di sebelah kanan, 30 piksel di bawahnya, dan 40 piksel di sebelah kirinya.

Contoh 8. Bagaimana margin diterapkan

```
<!DOCTYPE html>
<html>
<head>
<title>Margins</title>
<style>
#object1 {
background :lightgreen;
border-style:solid;
border-width:1px;
font-family :"Courier New";
font-size :9px;
width :100px;
height :100px;
padding :5px;
margin :10px 20px 30px 40px;
}
table {
padding :0;
border :1px solid black;
background :cyan;
}
</style>
</head>
<body>
<table>
<tr>
<td>
<div id='object1'>margin:<br>10px 20px 30px 40px;</div>
</td>
</tr>
</table>
</body>
</html>
```



Gambar 2.34 Tabel luar mengembang sesuai dengan lebar margin

Membuat layout dua kolom

Layout dua kolom menggunakan sedikit lebih banyak HTML untuk mencapai efek beberapa kolom. Ini sering dilakukan untuk menambahkan menu di sepanjang sisi halaman atau tautan ke cerita atau konten lain. Daftar berikut menunjukkan HTML yang terlibat untuk layout lebar tetap dua kolom.

Contoh 9 Layout Lebar Tetap Dua Kolom

```
<!doctype html>
<html>
<head>
<title>Two Column</title>
<link rel=”stylesheet” type=”text/css” href=”double.css”>
</head>
<body>
<div id=”container”>
 <div id=”mainContainer”>
 <div id=”content”>
 <h2>Disini ada beberapa konten</h2>
 <p>Dari sini cerita akan dimulai go</p>
 <h2>Disini ada beberapa konten lainnya</h2>
 <p>Ini adalah cerita lainnya</p>
 </div> <!-- end content -->
 <div id=”sidebar”>
 <h3>Menu</h3>
 <ul>
 <li>Menu item 1</li>
 <li>Menu item 2</li>
 <li>Menu item 3</li>
```

```
 </ul>
 </div> <!-- end sidebar -->
 </div> <!-- end mainContainer -->
</div> <!-- end container -->
</body>
</html>
```

HTML ini menggunakan wadah dari layout kolom tunggal dan menambahkan wadah lain untuk menampung konten. ini disebut sebagai mainContainer, menampung konten dan bilah sisi. Tambahan lainnya adalah sidebar itu sendiri, tepat berjudul sidebar. Bilah sisi itu menyimpan menu dengan daftar tidak berurutan (<ul>) di dalamnya. CSS untuk layout dua kolom ditunjukkan pada daftar berikut.

Contoh 10 CSS untuk Layout Lebar Tetap Dua Kolom

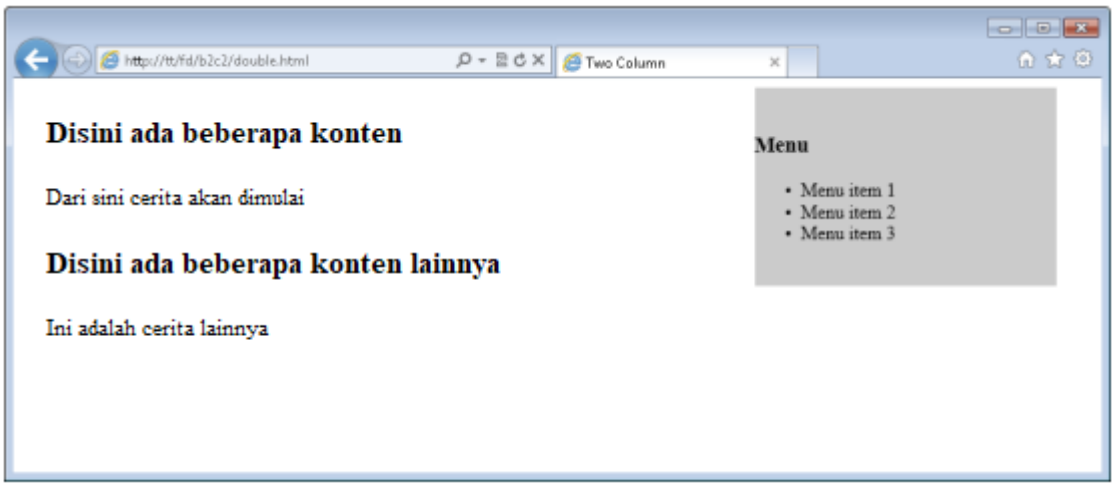
```
#container {
 margin: 0 auto;
 width: 900px;
}
#mainContainer {
 float: left;
 width: 900px;
}
#content {
 clear: left;
 float: left;
 width: 500px;
 padding: 20px 0;
 margin: 0 0 0 30px;
 display: inline;
}
#sidebar {
 float: right;
 width: 260px;
 padding: 20px 0;
 margin: 0 20px 0 0;
 display: inline;
 background-color: #CCCCCC;
}
```

CSS ini menggunakan beberapa item yang telah kita lihat, antara lain margin, padding, clear, dan background-color. Di CSS ini hal yang baru adalah float dan properti display. Properti float menentukan apakah sebuah elemen akan bergerak atau mengambang di dalam sebuah layout, baik ke kiri atau ke kanan atau apakah elemen tersebut tidak akan mengambang sama sekali (tidak ada), seperti defaultnya. Namun, karena kita ingin membuat dua kolom bersebelahan, kita perlu melayangkan wadah konten ke kiri dan bilah sisi ke kanan. Oleh

karena itu, jika kita ingin sidebar muncul di sebelah kanan, kita hanya perlu menukar float: left di #content CSS dengan float: right yang terdapat di CSS #sidebar.

Properti display mengatur bagaimana elemen harus ditampilkan. Elemen tertentu dikenal sebagai elemen level blok dan menampilkan seluruh lebar halaman. Elemen adalah contoh yang bagus untuk ini. Karena kita ingin membuat kolom muncul bersebelahan, kita perlu mengubah perilaku tampilan blok ini menjadi sebaris (kami memperkenalkan elemen sebaris di bab sebelumnya), sehingga elemen tidak memperpanjang lebar penuh halaman.

Tiga nilai yang sering digunakan untuk properti tampilan adalah blok (untuk memperluas elemen dengan lebar penuh), inline (untuk membuat elemen hanya menggunakan lebarnya sendiri untuk tampilan), dan none (yang menghilangkan elemen dari tampilan seluruhnya).



Gambar 2.35 Layout lebar tetap dua kolom.

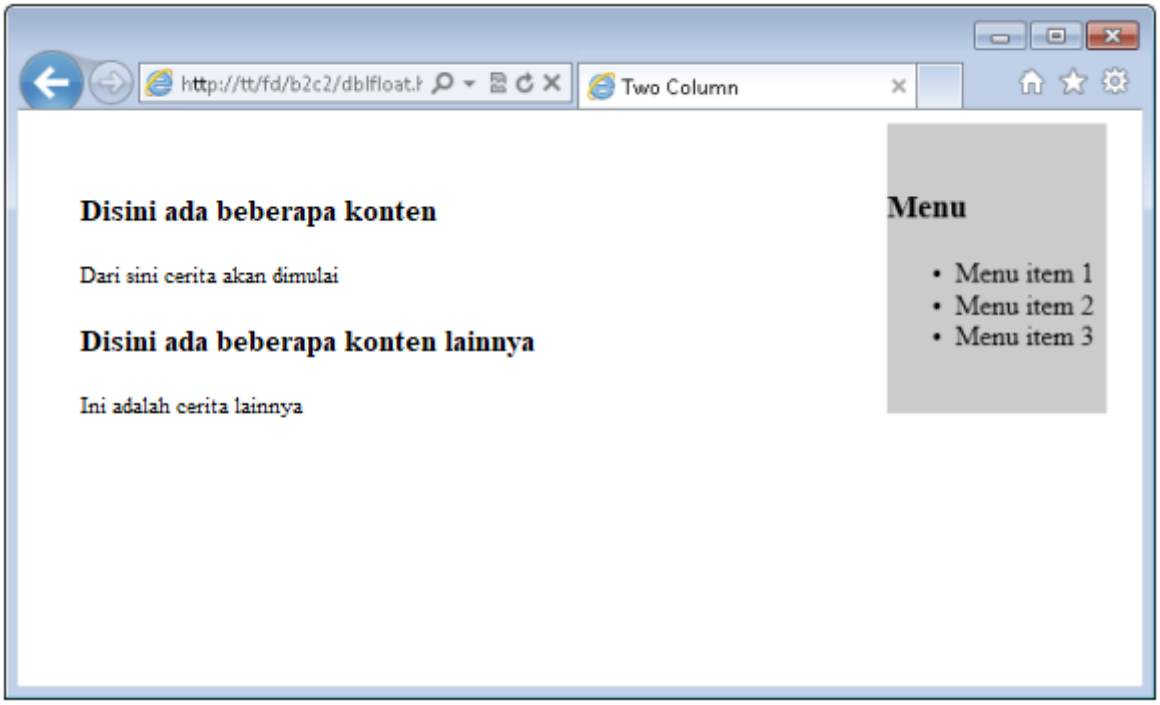
Layout yang ditunjukkan pada gambar diatas adalah layout lebar tetap. Mengonversi ini ke liquid layout berarti mengubah nilai width dan margin dalam CSS dari piksel (px) menjadi persentase (%). CSS untuk diubah menjadi liquid layout ditunjukkan pada daftar berikut.

Contoh 11 Mengonversi ke Liquid layoutan Dua Kolom.

```
#container {
 margin: 0 auto;
width: 100%;
}
#mainContainer {
 float: left;
 width: 100%;
}
#content {
 clear: left;
 float: left;
 width: 65%;
 padding: 20px 0;
 margin: 0 0 0 5%;
```

```
 display: inline;
}
#sidebar {
 float: right;
 width: 20%;
 padding: 20px 0;
 margin: 0 2% 0 0;
 display: inline;
 background-color: #CCCCCC;
}
```

Perubahan terjadi di bagian #container, #mainContainer, #content, dan #sidebar, untuk mengubah nilai sebelumnya yang menggunakan piksel menjadi persentase. Layout ini sekarang berubah dengan lebar browser, seperti yang ditunjukkan pada Gambar 2.36, di mana kita akan melihat bahwa lebar browser jauh lebih kecil.



Gambar 2.36 Membuat liquid layout dengan dua kolom.

Menyembunyikan elemen

Menyetel properti tampilan CSS ke none akan menyembunyikan elemen dari halaman. Saat kita melakukannya, elemen tersebut akan dihapus seluruhnya dari halaman. kita juga dapat menggunakan properti CSS lain, visibility, untuk menyembunyikan elemen. Saat menyembunyikan elemen dengan properti visibility (visibility: hidden;), kotak atau area pada halaman tetap berada di tempatnya tetapi elemen tersebut menjadi tidak terlihat. Membuat elemen terlihat lagi (visibility: terlihat;) menunjukkan elemen tersebut.

2.22 MENAMBAHKAN HEADER DAN FOOTER KE HALAMAN

Header biasanya digunakan untuk menyampaikan informasi seperti nama situs atau untuk menyediakan menu; footer digunakan untuk memberikan informasi tambahan seperti

hak cipta dan juga digunakan untuk menyediakan peta tautan dalam sebuah situs, yang dikenal sebagai *site map*. Selain itu, kami memberi tahu kita cara membuat menu di dalam header.

Membuat header, menu header, dan footer

Anda telah melihat cara membuat layout multi-kolom dengan area konten utama dan bilah sisi. Untuk membuat layout ini, kita menambahkan elemen untuk menahan konten bilah sisi. kita kemudian menerapkan aturan CSS ke untuk mengatur lebar dan posisinya. Membuat header dan footer sebagian besar dilakukan dengan cara yang sama. tambahan dibuat untuk menampung konten untuk masing-masing dan kemudian aturan diterapkan ke elemen tersebut untuk memposisikannya. Ini menjadi contoh terakhir dalam bab ini, ini berfungsi sebagai latihan batu penjuru.

1. Buka text editor
2. Masukkan HTML ini pada text dokumen :

```
<!doctype html>
<html>
<head>
<title>Two Column With Header and Footer</title>
<link rel=”stylesheet” type=”text/css” href=”final.
css”>
</head>
<body>
<div id=”container”>
 <div id=”header”>
 <h1>This is my site!</h1>
 </div> <!-- end header -->
 <div id=”menu”>
 <ul>
 <li><a href=”#”>Home</a></li>
 <li><A href=”#”>Services</a></li
<li><a href=”#”>About Me</a></li>
 <li><a href=”#”>Contact Me</a></li>
 </ul>
 </div> <!-- end menu -->
 <div id=”mainContainer”>
 <div id=”content”>
 <h2>Disini ada beberapa konten</h2>
 <p>Dari sini cerita akan dimulai go</p>
 <h2>Disini ada beberapa konten lainnya</h2>
 <p>Ini adalah cerita lainnya</p>
 </div> <!-- end content -->
 <div id=”sidebar”>
 <h3>Menu</h3>
 <ul>
 <li>Menu item 1</li>
 <li>Menu item 2</li>
```