

## BAB 13 DESAIN WEB STATIS DAN HTML

Jika kita sering berselancar di dunia internet kita akan sering menjumpai halaman-halaman web seperti pada Gambar 13.1. di samping ini. Halaman web ini ada yang bersifat statis tetapi juga ada yang dinamis. Tetapi apapun sifatnya, halaman-halaman tersebut hampir pasti melibatkan bahasa HTML.

Bab ini membahas dua standar kompetensi yaitu menerapkan dasar-

dasar pembuatan web statis tingkat dasar dan membuat file HTML sesuai spesifikasi. Penggabungan dua kompetensi ini karena kedekatan isi kompetensi dasar. Standar kompetensi membuat file HTML sesuai spesifikasi terdiri dari empat kompetensi dasar yaitu menetapkan pemakaian struktur file, membuat struktur file HTML, memformat file dan menambahkan obyek. Standar kompetensi menerapkan dasar-dasar pembuatan web statis tingkat dasar juga terdiri dari empat kompetensi dasar yaitu menjelaskan konsep dasar dan teknologi web, mempersiapkan pembuatan web, melakukan pembuatan web, dan menampilkan web di *browser*.

Rangkuman diletakkan pada akhir bab dilanjutkan dengan soal-soal latihan yang disusun dari soal-soal yang mudah hingga soal-soal yang sulit. Latihan soal ini digunakan untuk mengukur kemampuan terhadap kompetensi dasar ini. Sebelum mempelajari kompetensi ini ingatlah kembali dasar sistem komputer, sistem operasi, dan algoritma pemrograman dasar.



Gambar 13.1. Halaman web.

## TUJUAN

Setelah mempelajari bab ini diharapkan kalian akan mampu :

- o Menjelaskan konsep dasar dan teknologi web
- o Menetapkan pemakaian dan struktur dokumen
- o Mempersiapkan pekerjaan pembuatan web
- o Melakukan pembuatan dokumen web baru dan menampilkan dalam web *browser*
- o Membuat struktur dokumen dengan bahasa HTML
- o Memformat dokumen dan menambahkan obyek
- o Membuat tabel
- o Membuat link antar dokumen

### 13.1. KONSEP DASAR DAN TEKNOLOGI WEB

*World Wide Web* secara luas lebih dikenal dengan istilah *Web*. Web pertama kali diperkenalkan pada tahun 1992. Hal ini sebagai hasil usaha pengembangan yang dilakukan CERN di Swiss. Internet dan web adalah dua hal yang berbeda. Internet lebih merupakan perangkat keras, sedangkan web adalah perangkat lunak. Selain itu, protokol yang dipakai oleh keduanya juga berbeda. Internet menggunakan TCP/IP sebagai protokol operasionalnya, sedangkan web menggunakan HTTP (*Hyper Text Transfer Protocol*).

Web disusun dari halaman-halaman yang menggunakan teknologi web dan saling berkaitan satu sama lain. Suatu standar teknologi web saat ini sudah tersusun, meskipun penerapannya belum didukung oleh seluruh pengembang web. Standar ini disusun oleh suatu badan yaitu *World Wide Web Consortium* (W3C). Standar ini dibutuhkan karena semakin banyaknya variasi dalam teknologi web sehingga terkadang satu sama lain tidak kompatibel.

#### 13.1.1 Standar Teknologi Web

Secara umum teknologi desain web terbagi menjadi beberapa layer (lapisan), yaitu structural layer, presentation layer dan behavioral layer.

- **Structural layer**

Layer ini berhubungan dengan struktur dokumen web. Bagaimana sebuah dokumen tersusun, format apa yang dipakai, tanda atau mark up apa yang digunakan merupakan bagian dari layer ini. Standar teknologi yang direkomendasikan saat ini adalah *Extensible Hypertext Markup Language* (XHTML) dan *Extensible Markup Language* (XML). XHTML adalah HTML versi terakhir (4.01) yang ditulis ulang dengan aturan-aturan yang lebih ketat mengacu pada XML. Sedangkan XML adalah sekumpulan aturan untuk menyusun bahasa *markup*.

- **Presentation layer**

Layer ini berhubungan dengan bagaimana mengatur tampilan dokumen pada layar, suara yang keluar, atau bagaimana format pencetakan dokumen. Pada teknologi web lama bagian ini menyatu dengan *structural layer*. Tapi pada standar baru, layer ini disarankan untuk dipisah. Yang termasuk teknologi ini adalah *Cascading Style Sheets* (CSS).

- **Behavioral layer**

Layer ini berhubungan dengan masalah penggunaan bahasa skrip dan pemrogramannya untuk tujuan meningkatkan sisi interaktif dan dinamis halaman *web*. Yang termasuk dalam layer ini adalah *Document Object Model* (DOM) dan JavaScript. DOM memungkinkan suatu dokumen atau skrip untuk mengakses atau meng-update isi, struktur, dan style dari dokumen. JavaScript merupakan teknologi yang cukup lama dan tetap digunakan untuk menambah dokumen menjadi lebih interaktif.

### 13.1.2. *Web* Statis dan *Web* Dinamis.

Halaman *web* dapat digolongkan menjadi *web* statis dan *web* dinamis. Pengertian *web* statis dan *web* dinamis seringkali mengundang perdebatan. Sebagian pengguna internet menyatakan jika pada halaman-halaman *web* dilengkapi dengan animasi yang bergerak maka disebut *web* dinamis sedangkan jika halaman-halaman *web* tersebut hanya berisi teks dan gambar yang tidak bergerak maka disebut *web* statis. Namun berdasarkan kesepakatan maka pengertian statis dan dinamis tidak ditentukan oleh ada atau tidaknya animasi bergerak pada halaman-halaman *web*, tetapi ditentukan oleh isi atau informasi yang ada pada halaman-halaman tersebut.

Data dan informasi yang ada pada *web* statis tidak berubah-ubah. Dokumen *web* yang dikirim kepada *client* akan sama isinya dengan apa yang ada di *web server*. Sedangkan *web* dinamis, memiliki data dan informasi yang berbeda-beda tergantung input apa yang disampaikan *client*. Dokumen yang sampai di *client* akan berbeda dengan dokumen yang ada di *web server*.

Contoh paling mudah untuk membedakan *web* statis dan *web* dinamis adalah bila kalian membuka situs Google. Halaman awal adalah statis karena kita tidak melihat perubahan isi atau informasi. Halaman ini baik di komputer *client* maupun di *web server* akan sama. Namun begitu kita memasukkan kata pada textbox yang tersedia kemudian menekan tombol search maka kita sedang berinteraksi dengan *web server* Google. *Web server* akan mengirimkan halaman *web* sesuai yang diminta oleh *client*. Tampilan di sisi *client* akan berupa daftar alamat dan keterangannya. Sedangkan di sisi *server* isi dokumennya adalah serangkaian kode-kode untuk mencari apa yang diinputkan *client*.

Bab ini secara khusus akan mempelajari tentang pembuatan *web* statis sedangkan *web* dinamis akan kita bahas di bab 14.

## 13.2. PERSIAPAN PEMBUATAN *WEB*

Pembuatan halaman *web* membutuhkan persiapan tidak saja pengetahuan tentang bagaimana disain halaman *web*, namun juga perlu dukungan persiapan perangkat keras, perangkat lunak, dan yang lainnya.

### 13.2.2. Perangkat Keras

Perangkat keras yang dibutuhkan untuk pembuatan halaman *web* tidak berbeda jauh dengan kebutuhan komputasi biasa. Seperangkat komputer lengkap dengan CPU, monitor, keyboard, mouse, printer dan beberapa perangkat tambahan lain sudah dapat digunakan untuk membuat halaman *web*. Spesifikasi tergantung dari perangkat lunak yang akan diinstal pada perangkat komputer tersebut. Jika kita menginstal *web server*, pengolah gambar untuk disain halaman *web*, HTML editor yang komplet, tentu kita membutuhkan spesifikasi yang lebih tinggi.

### 13.2.3. Perangkat Lunak

- **Sistem operasi**

Sistem operasi memegang peranan penting dalam pembuatan *web* karena pada sistem operasi itu akan ditanamkan (diinstal) *web server*, *web editor*, sistem manajemen basis data dan bahasa pemrograman. Artinya pilihan pada sistem operasi tertentu akan menentukan pula pilihan *web server*, perangkat pengembang dan bahasa pemrograman yang akan digunakan. Hal ini dikarenakan adanya masalah kompatibilitas antar perangkat lunak. Sebagai contoh, apabila kita memilih menggunakan sistem operasi Linux maka kita tidak dapat menginstal IIS sebagai *web server*.

Selain masalah kompatibilitas, hal lain yang juga perlu dipertimbangkan dalam penentuan sistem operasi yang akan kita gunakan pada *server* adalah masalah keamanan, stabilitas, kemudahan konfigurasi. Keamanan berhubungan dengan kemampuan sistem operasi untuk melindungi diri dari serangan virus, spam, atau kode-kode jahat yang sengaja disusupkan. Kemampuan ini sangat penting diperhatikan karena lalu lintas data dalam internet sangat rentan terhadap gangguan virus, spam, dan pengganggu lainnya. Stabilitas berhubungan dengan kemampuan sistem operasi untuk bekerja terus-menerus untuk merespon permintaan *client*. Kemudahan konfigurasi berhubungan mudah tidaknya konfigurasi dilakukan terhadap sistem operasi dalam perannya sebagai *server*.

- **Web Server**

*Web server* adalah perangkat lunak yang bertindak melayani permintaan-permintaan *client* terhadap halaman-halaman *web* tertentu. Ada beberapa nama yang cukup populer dalam dunia *web server*. Diantaranya adalah Apache dan IIS (Internet Information Service). Sampai dengan Desember

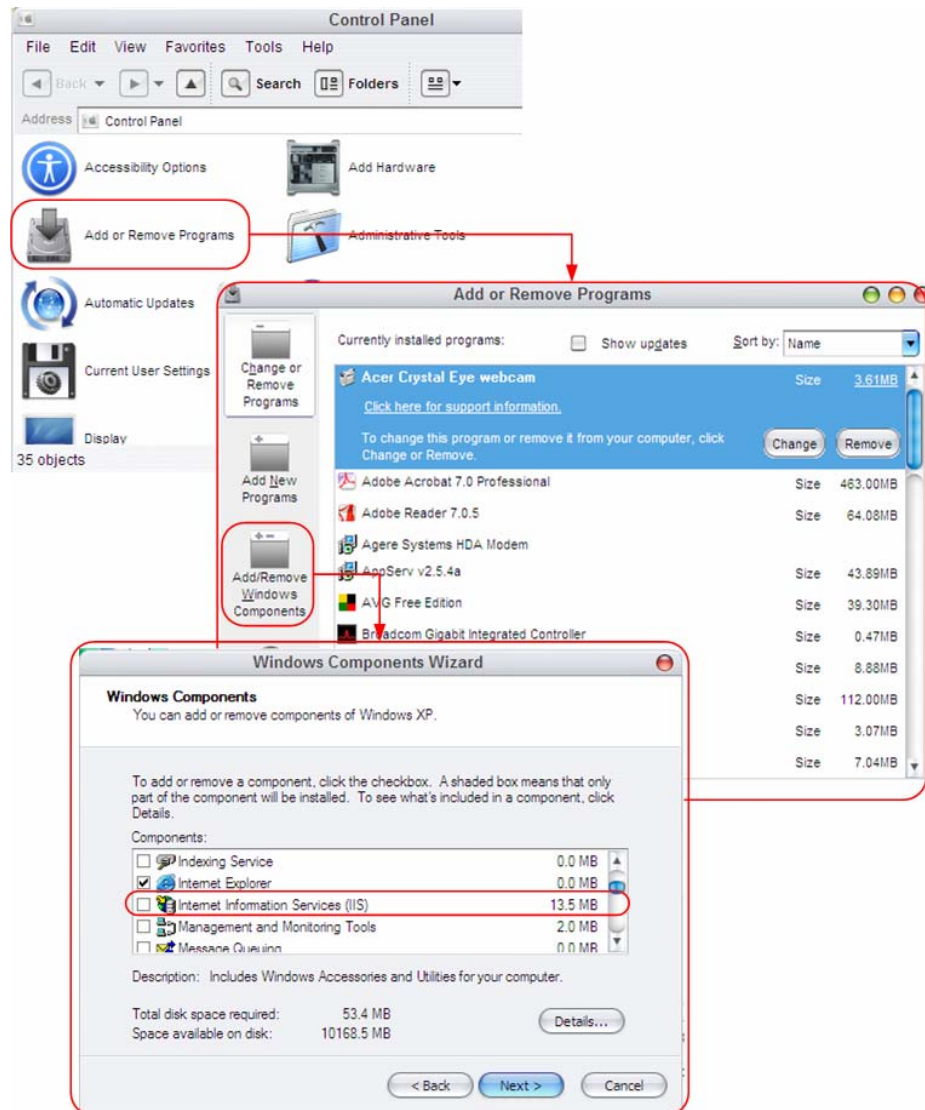
2007, wikipedia mencatat Apache berada dalam posisi pertama sebagai *web server* yang paling banyak digunakan, disusul IIS.

Apache dapat digunakan baik untuk *web* statis maupun *web* dinamis dan mendukung banyak platform sistem operasi dan bahasa pemrograman *server*, antara lain Perl, Python, Java (JSP dengan menggunakan Tomcat) dan tentu saja PHP. Fungsi-fungsi keamanan *web* juga dikendalikan dengan sangat baik. Dukungan pada koneksi dengan berbagai basis data, seperti MySQL, SQL Lite, PostgreSQL, Oracle, DB2 dan lain-lain dapat dilakukan dengan baik. Gambar 13.2 menunjukkan bagaimana *web server* Apache (httpd) digunakan di Linux.



Gambar 13.2. Menjalankan service Apache (httpd) pada Linux.

IIS adalah *web server* keluaran Microsoft. Sebutan *web server* bagi IIS mungkin tidak terlalu tepat, karena selain *web server*, IIS juga memberikan fasilitas *file server*, *email server* dan layanan lain berbasis internet. Oleh karena itu istilah yang tepat mungkin adalah *internet based-service*. Perangkat lunak ini dibundel dalam sistem operasi Microsoft Windows. Namun secara default tidak langsung diinstall. Sehingga kalau kita mau menggunakannya kita harus menginstall lebih dulu. Gambar 13.3 menunjukkan bagaimana IIS diinstall.



Gambar 13.3. Memeriksa dan menginstal IIS.

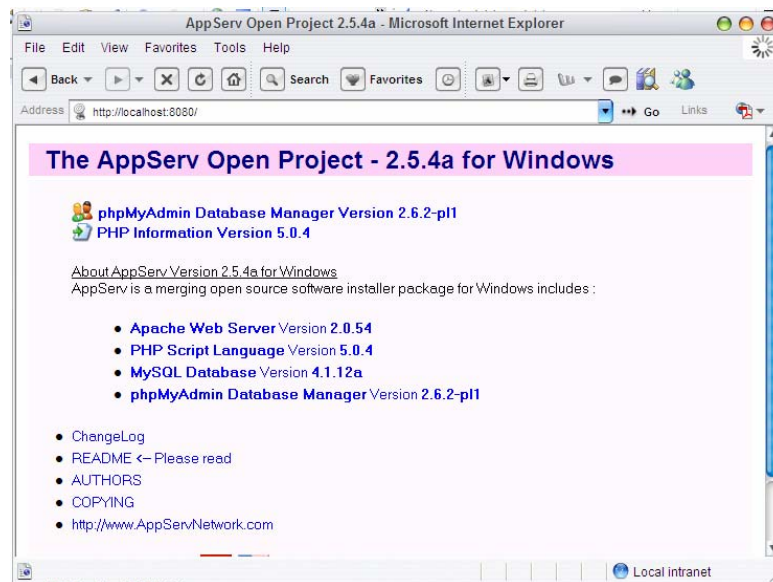
- **Web / HTML Editor**

*Web / HTML Editor* adalah perangkat lunak yang digunakan untuk membuat halaman-halaman *web*, baik yang bersifat statis maupun dinamis. Di pasar perangkat lunak, saat ini tersedia banyak sekali jenis perangkat pengembang *web*, mulai dari yang sederhana sampai yang canggih dan kompleks. Namun sebenarnya untuk membuat halaman *web* baik statis maupun dinamis kita dapat menggunakan teks editor biasa seperti Notepad atau Vi. Hanya saja teks editor tidak menyediakan fasilitas-fasilitas yang memudahkan kita dalam membuat halaman *web*. Pada perangkat pengembang *web* yang lebih kompleks seperti Adobe

Dreamweaver (dulu Macromedia Dreamweaver), Microsoft Visual Studio .Net, dan beberapa yang lainnya, kita akan mendapati fasilitas yang sangat membantu mempercepat pembuatan halaman *web*, antara lain: tampilan berbasis GUI, *automatic code completion* (melengkapi kode secara otomatis), WYSIWYG (*What You See Is What You Get*) HTML Editor, koneksi ke basis data yang lebih mudah, dan banyak lagi fasilitas. Tentu saja perangkat pengembang ini berharga relative mahal. Pada bagian lain dari bab ini kalian akan diajak untuk mengenali sedikit beberapa perangkat lunak ini.

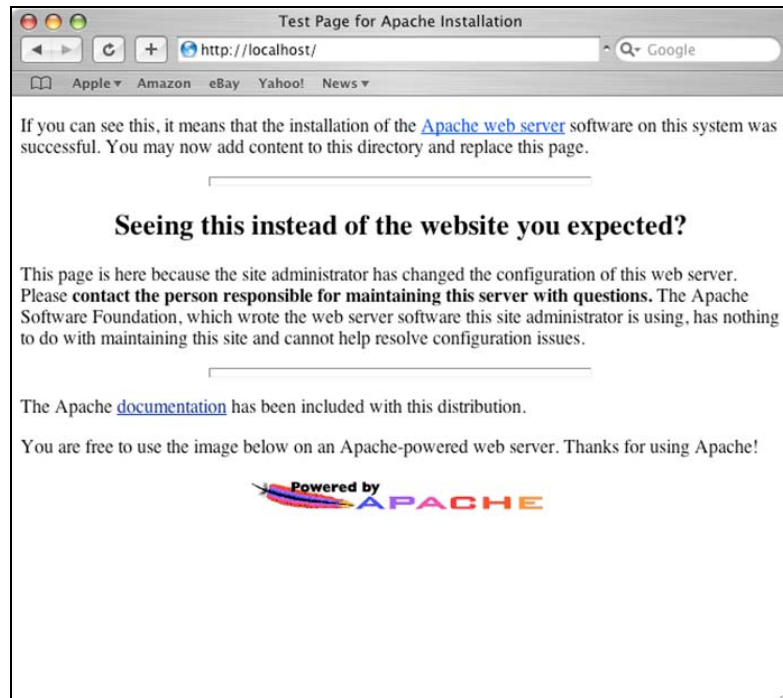
- **Web Browser**

*Web browser* berfungsi menerjemahkan kode-kode HTML menjadi tampilan yang kita kehendaki. Ada banyak *Web Browser* tersedia di internet. Hampir semuanya dapat kita *download* secara bebas. Beberapa nama yang cukup terkenal antara lain Microsoft Internet Explorer, Firefox, Opera atau Safari. Microsoft Internet Explorer adalah default *web browser* pada sistem operasi Microsoft Windows (lihat Gambar 13.4). Firefox adalah default *web browser* pada sebagian besar sistem operasi Linux. Safari adalah default *web browser* pada sistem operasi Mac OS X (lihat Gambar 13.5). Sedangkan Opera adalah *web browser* keluaran Opera Software yang dapat berjalan pada berbagai *platform* sistem operasi (Gambar 13.6).

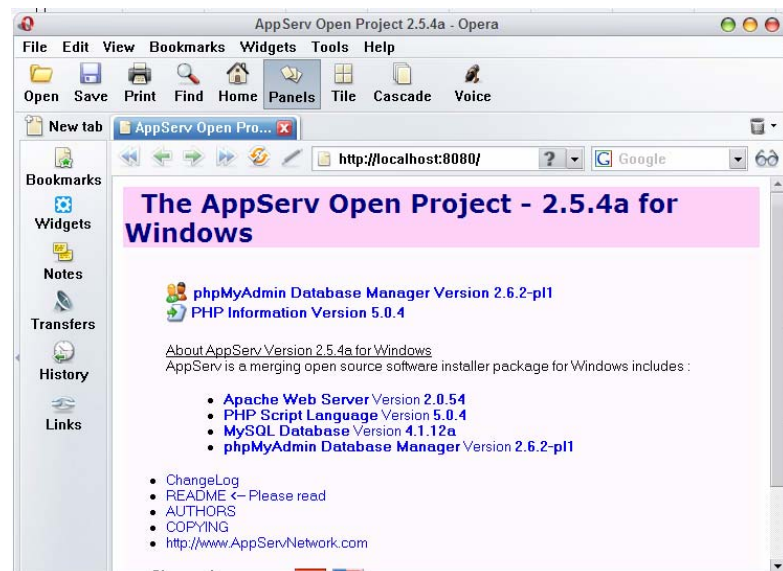


Gambar 13.4. Microsoft Internet Explorer.





Gambar 13.5. Safari.



Gambar 13.6. Opera.

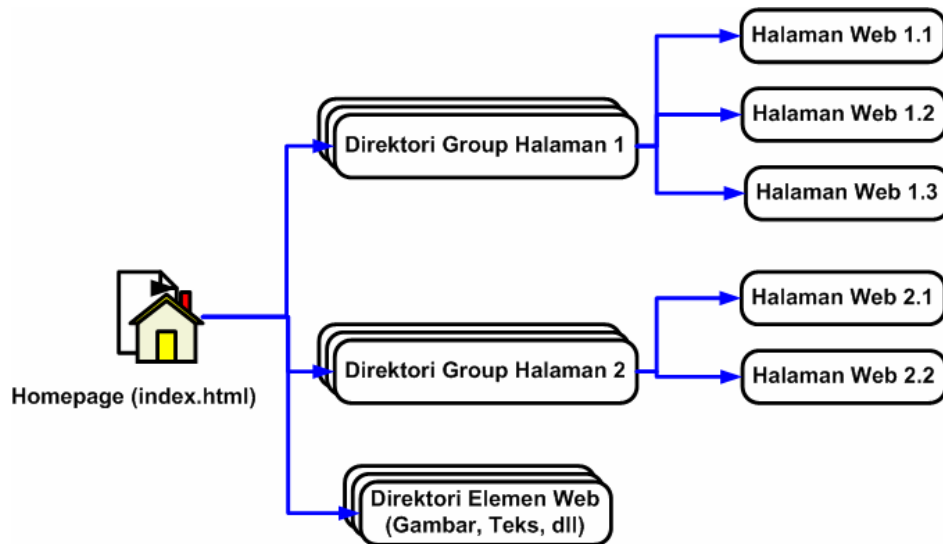


### 13.3. MEMBUAT DAN MENGUJI HALAMAN *WEB*

Ada dua model dalam pembuatan halaman *web* statis. Yang pertama adalah membuat halaman-halaman tersebut pada komputer lokal, kemudian setelah berhasil dipindahkan ke lokasi di *web server*. Model kedua adalah langsung membuat halaman-halaman *web* di lokasi *web server*. Lokasi *web server* dapat berada di satu komputer yang sama dengan tempat pembuatan halaman *web* atau di komputer lain yang berperan sebagai *server*. Cara pertama lebih mudah dilakukan karena tidak membutuhkan kerja *web server* di tahap disain. Pada bagian ini kita akan mencoba cara ini.

Buatlah direktori pada komputer kalian. Jika kalian menggunakan sistem operasi Windows, kalian tinggal buka Windows Explorer kemudian klik kanan pada area Windows eksplorer dan pilih New kemudian pilih Folder. Beri nama misalkan Latihan*Web*. Jika kalian menggunakan Linux, buka file manager yang tersedia misalnya Konqueror, Nautilus, atau Thunar. Lakukan cara yang sama seperti pada Windows Explorer.

Umumnya suatu situs *web* tidak hanya berisi satu halaman *web* tetapi kumpulan dari beberapa halaman *web* yang saling berkait. Biasanya pengembang *web* membuat struktur direktori untuk mempermudah pengaturan halaman. Perhatikan struktur direktori sebuah situs *web* pada Gambar 13.7.



Gambar 13.7. Contoh Struktur direktori situs *web*.

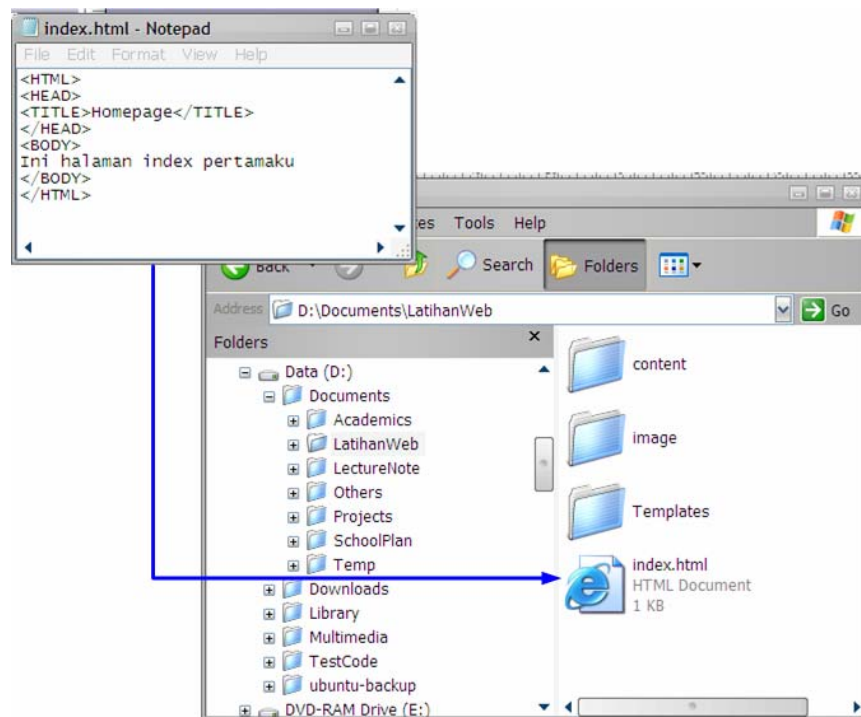
Halaman awal suatu situs *web* biasanya berupa halaman *web* yang diberi nama homepage. Biasanya filenya diberi nama index.html (atau bisa juga index.php, index.jsp, index.asp jika menggunakan bahasa skrip *server*). Di dalam direktori yang sama dengan index.html ini biasanya ada direktori-direktori lain yang berisi halaman-halaman yang dikelompokkan berdasarkan kedekatan

tema atau berdasarkan pengelompokkan lain. Selain itu ada direktori lain yang digunakan untuk menyimpan elemen-elemen yang digunakan dalam halaman *web*. biasanya berisi file-file gambar, teks, audio, video atau elemen-elemen yang lain.

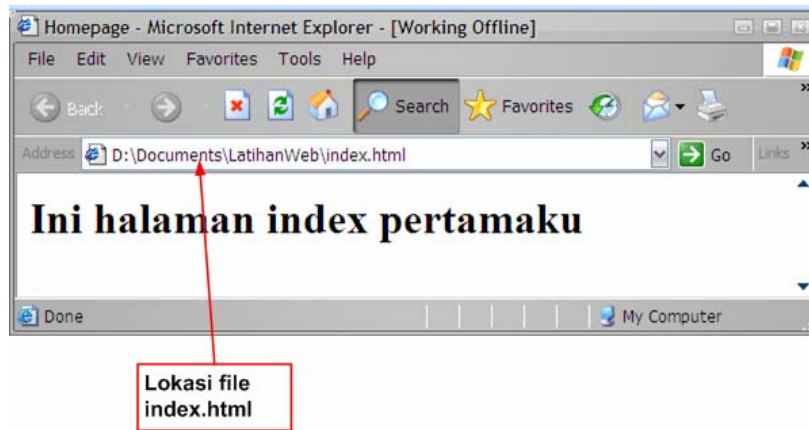
Buatlah struktur direktori seperti di atas dengan cara menambahkan sub direktori di bawah direktori *LatihanWeb* yang telah kalian buat. Beri nama sesuai keinginan kalian, tetapi usahakan nama direktori menunjukkan apa isi dari direktori tersebut. Buka teks editor, misalkan Notepad kemudian ketikkan kode berikut ini.

```
<HTML>
<HEAD>
<TITLE>Homepage</TITLE>
</HEAD>
<BODY>
Ini halaman index pertamaku
</BODY>
</HTML>
```

Simpan dengan nama `index.html` dan letakkan di direktori *LatihanWeb* (Gambar 13.8). Untuk menguji file tersebut, klik ganda pada file `index.html`. kalian akan mendapatkan tampilan seperti pada Gambar 13.9.



Gambar 13.8. File `index.html` dan lokasi penyimpanannya.



Gambar 13.9. Hasil pengujian file index.html.

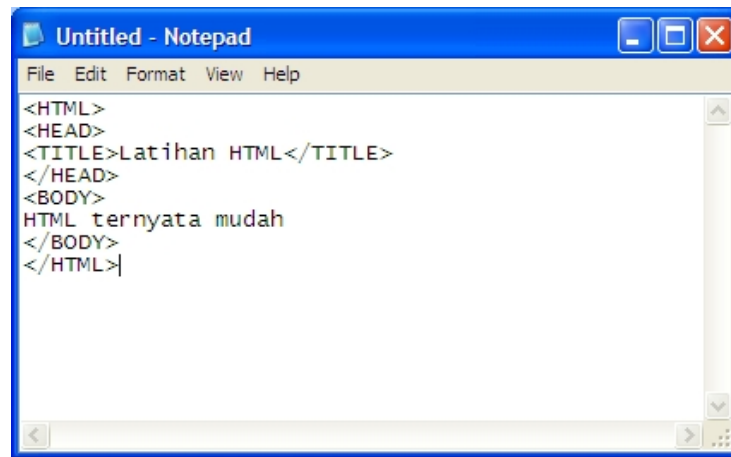
## 13.4. HTML

### 13.4.1. Pengertian HTML

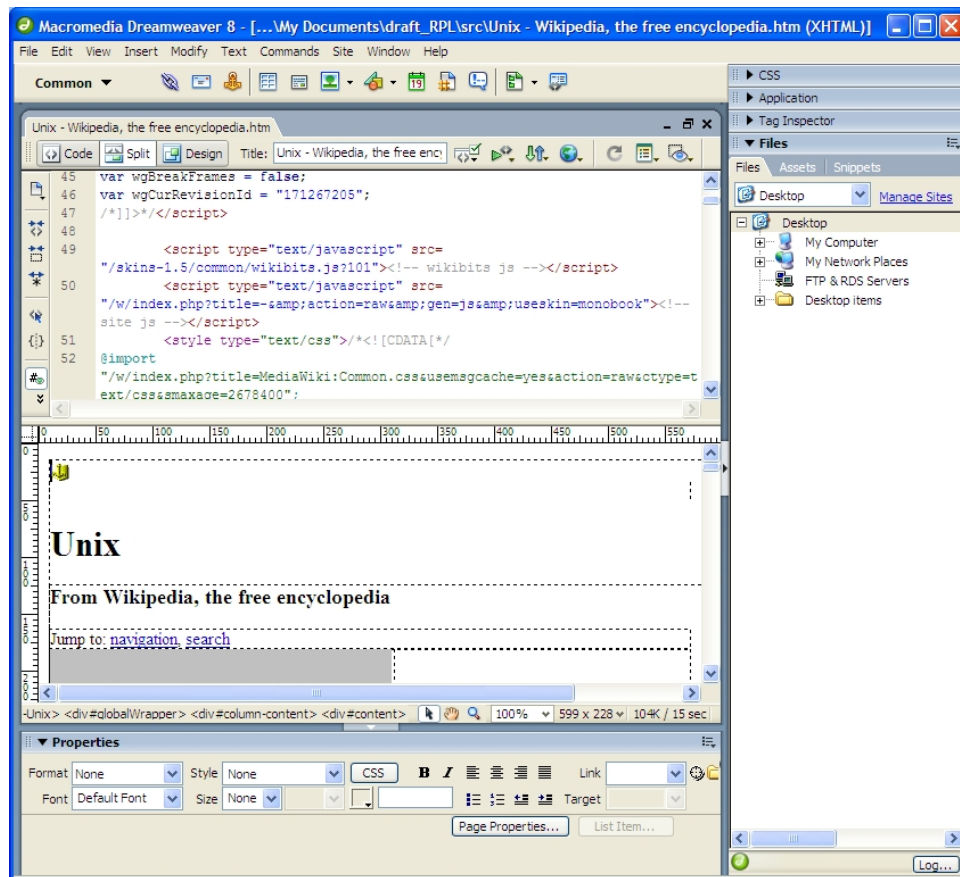
Gambar 13.8 dan 13.9 menunjukkan pada kalian bagaimana membuat halaman *web* sederhana. Halaman *web* yang kalian buat ini menggunakan bahasa yang disebut HTML (*Hypertext Markup Language*). HTML merupakan pengembangan dari standar pemformatan dokumen teks yaitu *Standard Generalized Markup Language (SGML)*. HTML sebenarnya adalah dokumen ASCII atau teks biasa yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu.

HTML dibuat oleh Tim Berners-Lee ketika masih bekerja untuk CERN. HTML dipopulerkan pertama kali oleh *browser Mosaic*. Selama awal tahun 90'an, HTML mengalami perkembangan yang sangat pesat. Setiap pengembangan HTML pasti akan menambahkan kemampuan dan fasilitas yang lebih baik daripada versi sebelumnya. Perkembangan yang pesat tersebut tidak sampai merubah cara kerja HTML.

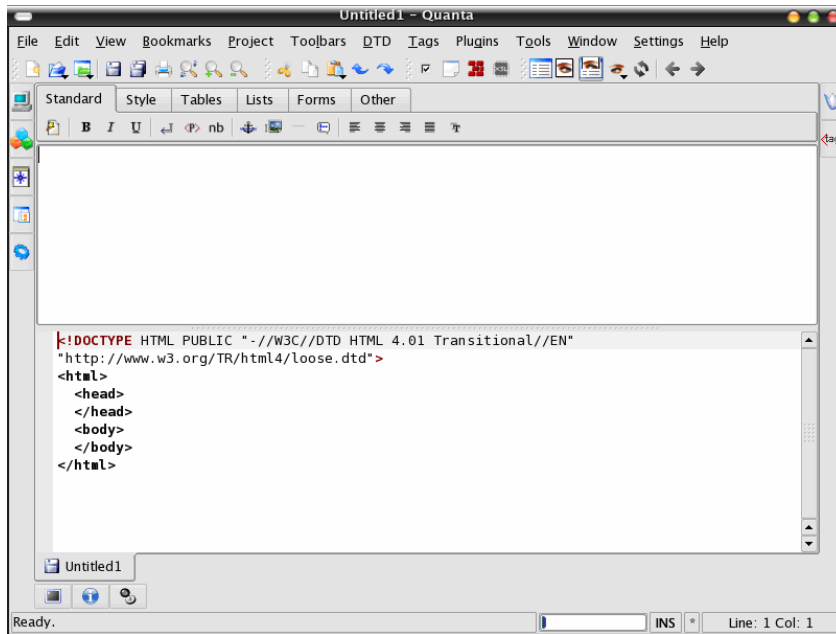
Sebuah dokumen atau file HTML agar dapat dibaca langsung oleh *browser* disimpan dalam ekstensi `.htm` atau `.html`. Untuk menulis HTML tidak dibutuhkan perangkat lunak yang spesifik, cukup dengan text editor sederhana seperti *Notepad* (pada Microsoft Windows) atau beragam text editor yang ada di platform Linux dan Apple Mac OS, seperti *vi*, *nano*, *joe*, *gedit*, *leafpad* dan lain-lain. Beberapa editor menyediakan fitur-fitur tambahan seperti *syntax coloring* (memberi warna pada kode-kode HTML) dan *code completion* (melengkapi secara otomatis kode yang akan dituliskan). Saat ini telah banyak perangkat lunak berbasis GUI yang sangat membantu dalam pembuatan halaman-halaman HTML. *Macromedia Dreamweaver* dan *Microsoft Frontpage* merupakan dua nama yang cukup populer di platform Microsoft Windows. Sedangkan di Linux tersedia *Quanta+*, *Bluefish* dan *Nvu*.



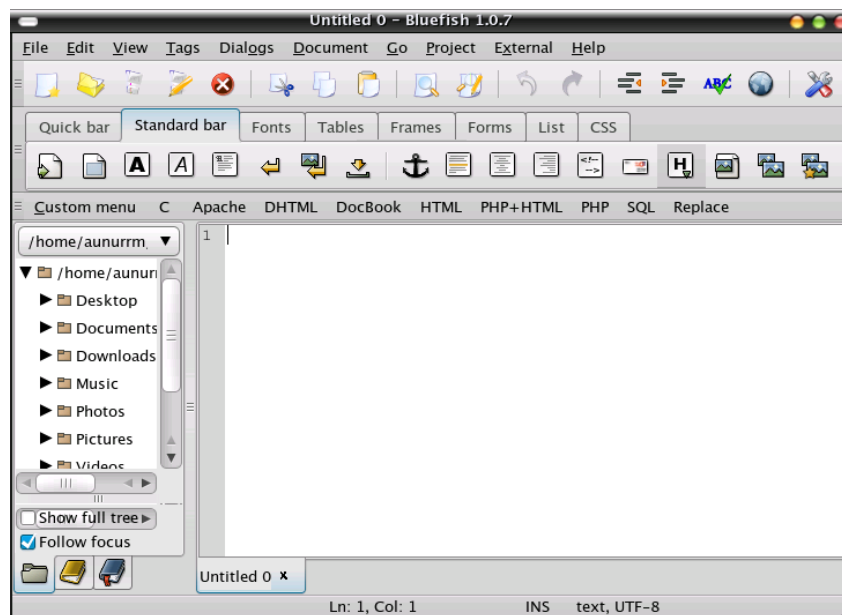
Gambar 13.10. Teks editor Notepad.



Gambar 13.11. Macromedia Dreamweaver.



Gambar 13.12. Quanta pada sistem operasi Linux



Gambar 13.13. Bluefish pada sistem operasi Linux

### 13.4.2. Struktur Umum File dengan Bahasa HTML

HTML adalah bahasa yang disisipkan (*embedded language*) pada dokumen dengan memberi tanda tertentu yang disebut **tag**. Tag merupakan aturan penulisan kode yang ditulis dengan diawali tanda lebih kecil dan di akhiri dengan tanda lebih besar (**<tag>**). *Browser* akan menentukan tampilan teks atau dokumen berdasarkan tag yang digunakan.

Sintaks penulisan tag mengikuti aturan-aturan umum berikut ini:

- a) Setiap tag mempunyai nama yang spesifik. Kadang-kadang diikuti opsi-opsi yang disebut atribut. Baik nama maupun opsi harus berada dalam tanda **<...>**.

Contoh:

```
<a href="/wiki/PHP" title="PHP">PHP</a>
```

Pada contoh ini tagnya memiliki nama **<a>** sedangkan atribut untuk tag **<a>** adalah **href** dan **title**. Sehingga baik nama tag dan atributnya harus berada di dalam tanda **<...>** seperti pada contoh.

- b) Sebagian besar tag berpasangan. Penulisan untuk tag yang berpasangan adalah sebagai berikut : **<namatag>...</namatag>**

Contoh:

```
<TITLE>Paragraf</TITLE>
<strong>Cetak Tebal</strong>
```

Pada tag yang berpasangan seperti pada contoh ini, **<TITLE>** adalah tag awal dan **</TITLE>** adalah tag akhir. Perhatikan tanda **/** pada tag akhir.

- c) Nama tag dan atribut-nya tidak bersifat case sensitive. Penulisan **<strong>Cetak Tebal</strong>** memberikan hasil yang sama dengan **<STRONG>Cetak Tebal</STRONG>**.
- d) Penulisan atribut suatu tag diletakkan setelah nama tag. Jika ada lebih dari satu atribut maka digunakan spasi untuk memisahkan. Urutan atribut tidak penting.

Contoh:

```
<FONT SIZE=3>Teks Baru</FONT>
<FONT SIZE=5 FACE="verdana">Teks Baru </FONT>
```

- e) Nilai dari atribut ditulis setelah tanda sama dengan (=). Pada contoh sebelumnya (lihat bagian d) terlihat bahwa atribut **SIZE** dari tag **FONT** memiliki nilai 5 sedangkan atribut **FACE** memiliki nilai **"verdana"**.
- f) Jika nilai dari atribut hanya tunggal, maka kita langsung menuliskan setelah tanda =. Jika lebih dari satu maka dapat digunakan tanda **'... '** atau **"..."**. Pada contoh bagian d, tampak bahwa penulisan nilai untuk atribut **SIZE** tanpa menggunakan tanda **"..."**, sedangkan pada atribut **FACE** tanda **"..."** untuk menandai kata **verdana**.

Dokumen HTML secara umum akan terdiri dari dua bagian yaitu Header dan Body (Gambar 13.14)

```
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1" /
>
<title>Untitled Document</title>
</head>

<body>
<strong>Cetak Tebal</strong>
</body>
</html>
```

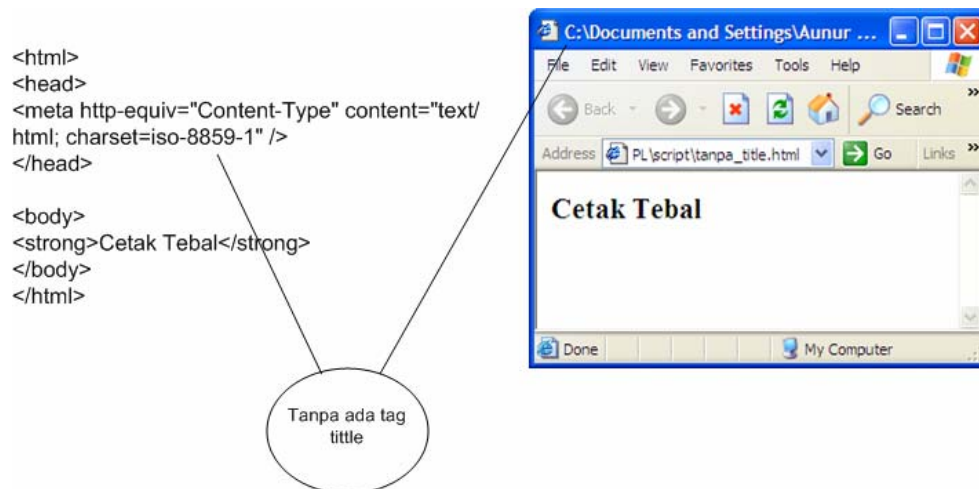
Header

Body

Gambar 13.14. Struktur umum dokumen HTML.

- **Header**

Bagian ini biasanya berisi berbagai macam keterangan tentang dokumen termasuk title (judul dokumen), posisinya dalam sekumpulan halaman *web* dan hubungannya dengan dokumen lain. Bagian ini ditandai dengan tag `<head> ... </head>`. Tag ini tidak mempunyai atribut. Di dalam tag ini kita dapat meletakkan beberapa tag lain seperti tag title dan tag link. Lihat Gambar 13.15 dan 13.16.



Gambar 13.15. Header dokumen HTML tanpa tag title.

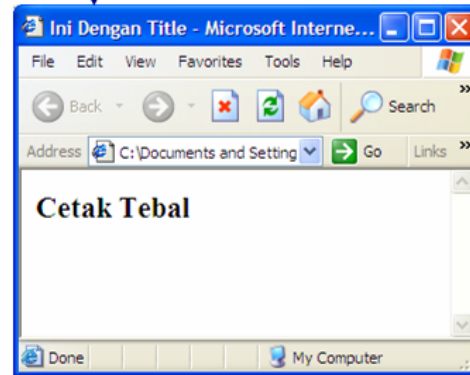


```

<html>
<head>
<meta http-equiv="Content-Type" content="text/
html; charset=iso-8859-1" />
<title>Ini Dengan Title</title>
</head>

<body>
<strong>Cetak Tebal</strong>
</body>
</html>

```



Gambar 13.16. Header dokumen HTML dengan tag title.

- **Body**

Body adalah bagian dari dokumen HTML tempat dimana kita meletakkan isi dari dokumen. Bagian ini ditandai dengan tag `<body>` dan diakhiri dengan `</body>`. Apapun yang berada diantara dua tanda ini disebut sebagai *body content*. Dokumen HTML yang paling sederhana mungkin hanya berisi sebaris atau dua baris teks saja tanpa format apapun (Gambar 13.17). Pada dokumen yang lebih kompleks, *body content* bisa berisi teks yang terformat, gambar, tabel atau bahkan animasi yang rumit (Gambar 13.18).

```

<BODY>
HTML ternyata mudah
</BODY>

```



Gambar 13.17. Dokumen HTML dengan *body content* sederhana.

```

<BODY>
<p class="style1">HTML ternyata mudah</p>
<p>Bisa menampilkan gambar</p>
<p class="style1"> </p>
</BODY>

```



Gambar 13.18. Dokumen HTML dengan *body content* yang lebih kompleks.

### 13.4.3.Format Dokumen

Ada banyak sekali tag HTML yang tersedia, baik itu berhubungan dengan deskripsi dokumen atau yang berhubungan dengan format tampilan dokumen. Tidak semuanya akan dibahas dalam buku ini. Pada bagian berikut ini akan dibahas beberapa tag yang biasa dipakai dalam format tampilan dokumen.

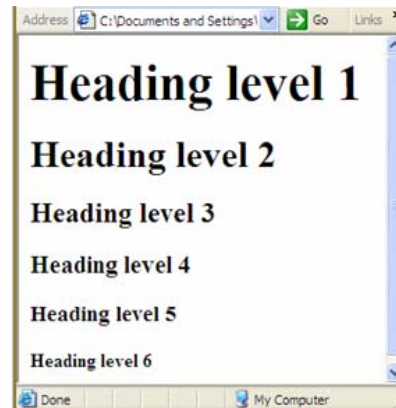
- **Heading**

*Heading* adalah sekumpulan kata yang menjadi judul atau subjudul dalam sebuah dokumen HTML. *Heading* berbeda dengan tag **<TITLE>**. HTML menyediakan enam tingkatan *heading*, dimana *heading* level 1 (**<H1>**) adalah yang terbesar dan *heading* level 6 (**<H6>**) adalah yang terkecil. Gambar 13.19 menunjukkan bagaimana penggunaan *heading*.

```

<HTML>
<HEAD>
  <TITLE>Heading</TITLE>
</HEAD>
<BODY>
  <H1>Heading level 1</H1>
  <H2>Heading level 2</H2>
  <H3>Heading level 3</H3>
  <H4>Heading level 4</H4>
  <H5>Heading level 5</H5>
  <H6>Heading level 6</H6>
</BODY>
</HTML>

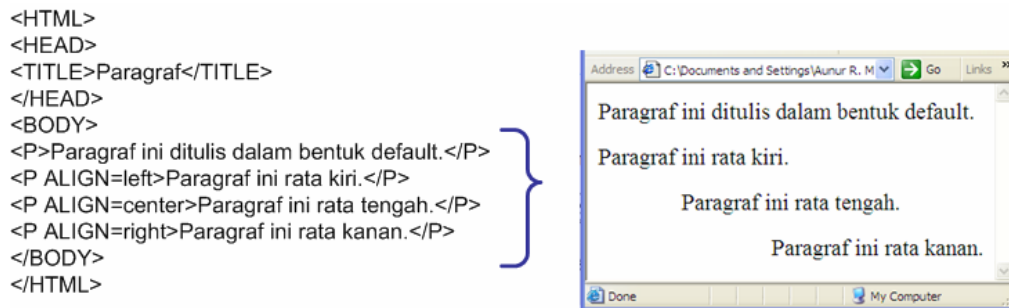
```



Gambar 13.19. Penggunaan heading.

- **Paragraph**

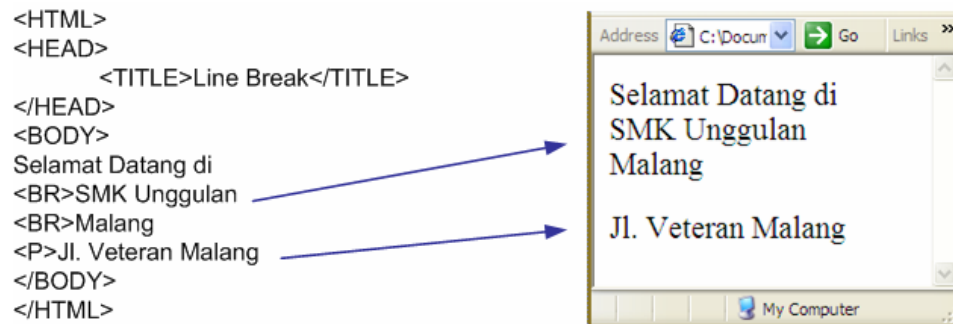
Paragraf dalam HTML dibuat dengan tag `<P>`. Tag ini akan membuat baris baru dengan menyisipi satu baris kosong. Penulisan isi paragraf diapit oleh `<P>` dan `</P>`. Pengaturan posisi paragraf dapat dilakukan dengan atribut **ALIGN** yang diikuti dengan posisi yang diinginkan, yaitu `left` untuk rata kiri, `center` untuk rata tengah horizontal dan `right` untuk rata kanan (Lihat Gambar 13.20).



Gambar 13.20. Penggunaan paragraph.

- **Line Break**

Line break digunakan untuk menuliskan teks pada baris berikutnya. Line break dibuat dengan tag tunggal `<BR>`. Tag ini akan membuat baris baru tanpa memberi baris kosong sebagaimana pada tag `<P>` (Lihat Gambar 13.21).



Gambar 13.21. Tag `<BR>` dan `<P>`.

- **List**

HTML menyediakan 3 cara untuk membuat daftar atau list, yaitu

- *Ordered list*

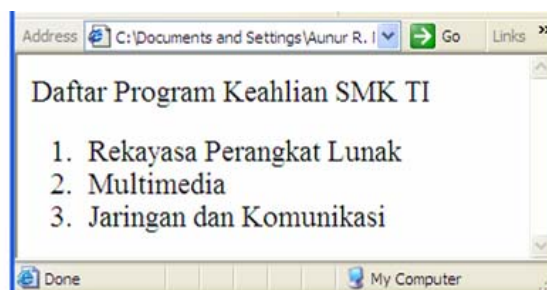
*Ordered list* digunakan untuk membuat daftar dimana setiap bagian memiliki nomor secara berurutan. *Ordered list* dimulai dengan tag `<OL>` dan diakhiri dengan tag `</OL>`, sedangkan setiap bagiannya digunakan tag `<LI>` tanpa tag penutup. Tag ini menggunakan angka sebagai urutan secara default, tapi bisa dirubah dengan merubah nilai atribut **TYPE**. Nilai-nilai atribut yang diijinkan dapat dilihat pada Tabel 13.1.

Tabel 13.1. Daftar atribut TYPE untuk *Ordered list* dan *Unordered list*.

Jenis List	Atribut	Fungsi
<i>Ordered list</i>	<b>TYPE=1</b>	Daftar berurutan dengan angka arab (1,2,3,...)
	<b>TYPE=I</b>	Daftar berurutan dengan angka romawi besar (I,II,III,...)
	<b>TYPE=i</b>	Daftar berurutan dengan angka romawi kecil (i,ii,iii,...)
	<b>TYPE=A</b>	Daftar berurutan dengan abjad besar (A,B,C,...)
	<b>TYPE=a</b>	Daftar berurutan dengan abjad kecil (a,b,c,...)
<i>Unordered List</i>	<b>TYPE=circle</b>	Daftar dengan tanda lingkaran
	<b>TYPE=square</b>	Daftar dengan tanda kotak
	<b>TYPE=disk</b>	Daftar dengan tanda cakram

Contoh penggunaan *Ordered list* dapat dilihat pada Gambar 13.22.

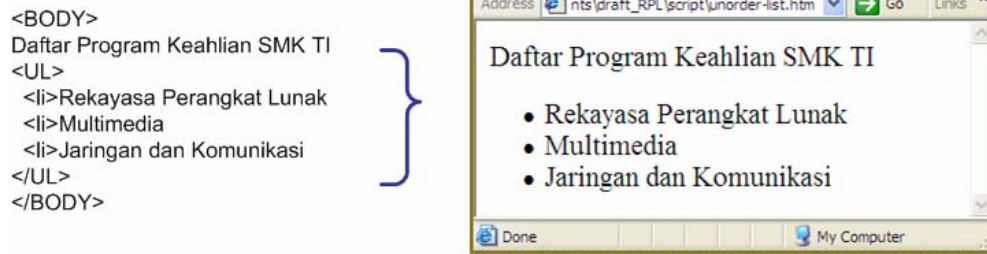
```
<BODY>
Daftar Program Keahlian SMK TI
<OL>
<li>Rekayasa Perangkat Lunak
<li>Multimedia
<li>Jaringan dan Komunikasi
</OL>
</BODY>
```



Gambar 13.22. Penggunaan *Ordered List*.

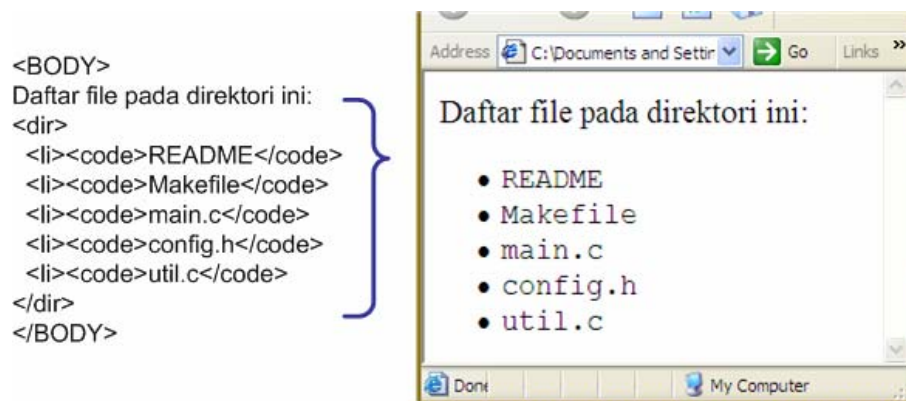
- *Unordered list*

*Unordered list* digunakan untuk membuat daftar yang disajikan tanpa nomor urut, melainkan dengan secara default dengan tanda bulatan utuh (bullet). Tanda ini bisa dirubah dengan merubah nilai atribut **TYPE** untuk unordered list. Nilai-nilai atribut yang diijinkan dapat dilihat pada tabel 13.1. *Unordered list* dimulai dengan tag `<UL>` dan diakhiri dengan tag `</UL>`, sedangkan setiap bagiannya digunakan tag `<LI>` tanpa tag penutup (lihat contoh pada Gambar 13.23).

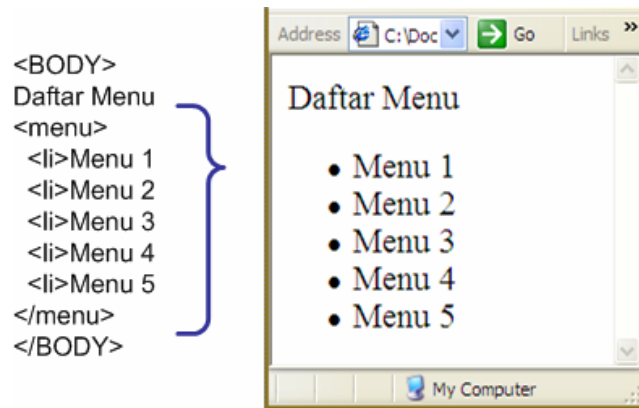


Gambar 13.23. Penggunaan *Unordered List*.

Tipe lain dari list yang termasuk Unordered list adalah *Directory list* dan *Menu list*. *Directory list* merupakan daftar tak bernomor yang digunakan untuk menangani direktori. Tag yang digunakan adalah `<DIR> ... </DIR>`. *Menu list* umumnya digunakan untuk menu pilihan. Tag yang digunakan adalah `<MENU> ... </MENU>`. Baik *Direktori list* maupun *Menu list* menghasilkan tampilan yang sama dengan *Unordered list* (Gambar 13.24 dan Gambar 13.24).



Gambar 13.24. Penggunaan *Direktori List*



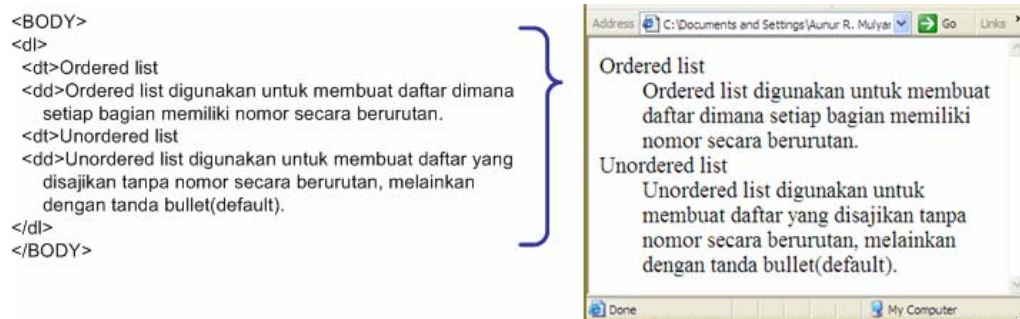
Gambar 13.25. Penggunaan *Menu List*

o *Definition list*

*Definition list* membuat daftar definisi mirip seperti tampilan pada kamus, dengan definisi suatu istilah agak menjorok ke kanan. Tiga buah pasang tag yang terkait dengan definition list adalah:

- `<DL> ... </DL>` untuk menyatakan tempat bagi daftar definisi.
- `<DT> ... </DT>` untuk menyatakan tempat bagi istilah yang akan didefinisikan.
- `<DD> ... </DD>` untuk menyatakan tempat bagi definisi dari istilah.

Contoh penggunaan *Definition list* dapat dilihat pada Gambar 13.26.



Gambar 13.26. Penggunaan *Definition List*.

- **Font**

HTML menyediakan fasilitas pengaturan huruf yang akan ditampilkan dalam dokumen. Pengaturan ini dilakukan dengan tag berpasangan **<FONT>** dan **</FONT>**. Tag ini memiliki beberapa atribut untuk mengatur ukuran, jenis dan warna huruf yang digunakan.

- Atribut **SIZE** untuk mengatur ukuran huruf, dimana nilai 1 untuk huruf terkecil dan nilai 7 untuk huruf terbesar.
- Atribut **FACE** untuk mengatur jenis huruf yang diinginkan, dimana nilainya berupa *string* nama font seperti Arial, Tahoma dan sebagainya.
- Atribut **COLOR** untuk mengatur warna teks yang dikehendaki, dimana nilainya dapat diisi dengan dua cara dengan menyebut nama warna dalam bahasa Inggris seperti *red*, *blue* dan *green* atau dengan menggunakan nilai RGB (*Red Green Blue*) seperti **FF0000** untuk merah.



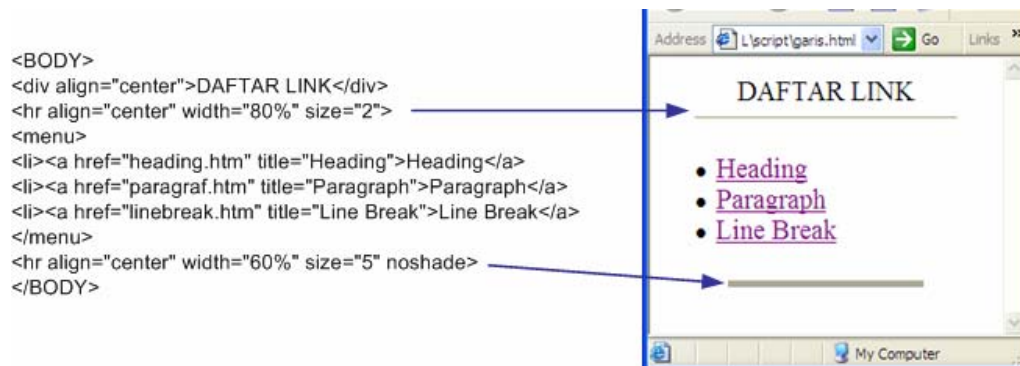
Gambar 13.27. Penggunaan tag Font.

#### 13.4.4. Penambahan Obyek

- **Horizontal Line**

Untuk mempercantik tampilan halaman HTML, kita dapat menambahkan garis horizontal dengan tag **<HR>**. Tag **<HR>** mempunyai atribut **SIZE** untuk menentukan ketebalan garis, atribut **WIDTH** untuk menentukan lebar garis, Atribut **ALIGN** untuk menentukan letak teks dalam garis, dan atribut **NOSHADOW** untuk mengatur agar garis tidak disertai bayangan. Gambar 13.28 menunjukkan bagaimana tag **<HR>** digunakan.





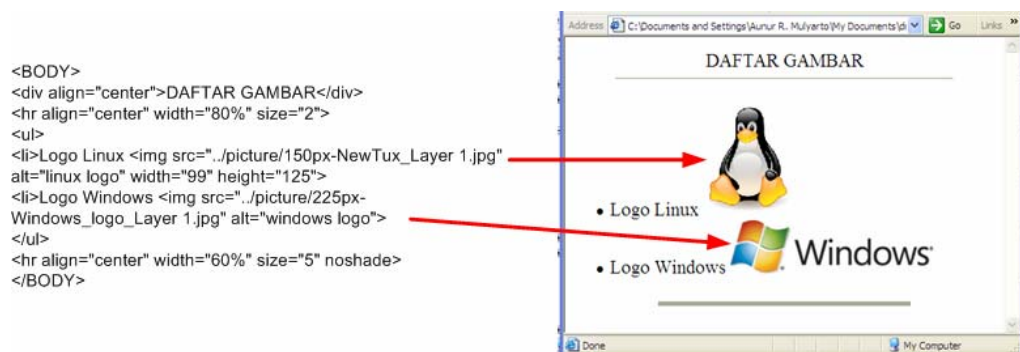
Gambar 13.28. Penggunaan garis.

## • Image

Dokumen HTML dapat diperindah dengan menyertakan gambar pada halaman *web* yang dibuat. Tag **<IMG>** dapat digunakan untuk memanggil dan menampilkan gambar pada halaman *web*. Sintaks penulisan tag **<IMG>** adalah:

```
<IMG SRC="file_gambar" ALT="nama_alternatif">
```

Atribut **SRC** digunakan untuk menentukan sumber file gambar yang akan ditampilkan. Atribut **ALT** berfungsi untuk memberi tulisan pengganti, apabila gambar tidak ditampilkan.



Gambar 13.29. Penggunaan tag image.

Untuk pengaturan gambar yang lebih baik, tag **IMG** menyediakan beberapa atribut, antara lain:

- Atribut **ALIGN** untuk mengatur penempatan teks pada gambar.
- Atribut **BORDER** untuk memberi bingkai pada gambar.
- Atribut **HEIGHT** dan **WIDTH** untuk mengatur tinggi dan lebar gambar.

Contoh berikut ini memperlihatkan penggunaan atribut-atribut tersebut.

```
<BODY>
<div align="center">DAFTAR
GAMBAR</div>
<hr align="center" width="80%"
size="2">
<ul>
<li>Logo Linux 
<li>Logo Windows 
</ul>
<hr align="center" width="60%"
size="5" noshade>
</BODY>
```



Gambar 13.30. Penggunaan atribut-atribut tag IMG.

Pada Gambar 13.30 terlihat perbedaan tampilan dibandingkan dengan Gambar 13.29. Tulisan Logo Linux terletak ditengah gambar karena kita menggunakan atribut **align** dengan nilai **middle**. Sedangkan tulisan logo Windows terletak di bagian atas karena kita menggunakan **align** dengan nilai **top**. Gambar logo windows diberi garis bingkai dengan menggunakan atribut **border** dengan nilai 2.

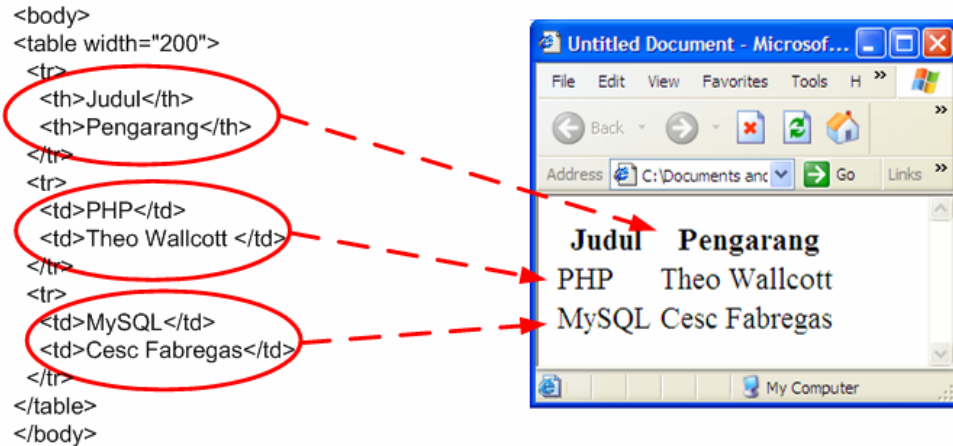
#### 13.4.5. Tabel

Tabel dalam HTML dibuat dengan menggunakan tag awal **<TABLE>** dan tag penutup **</TABLE>**. Tag ini memiliki beberapa bagian penting, seperti dapat dilihat pada Tabel berikut ini.

Tabel 13.2. Bagian-bagian pada tag Table

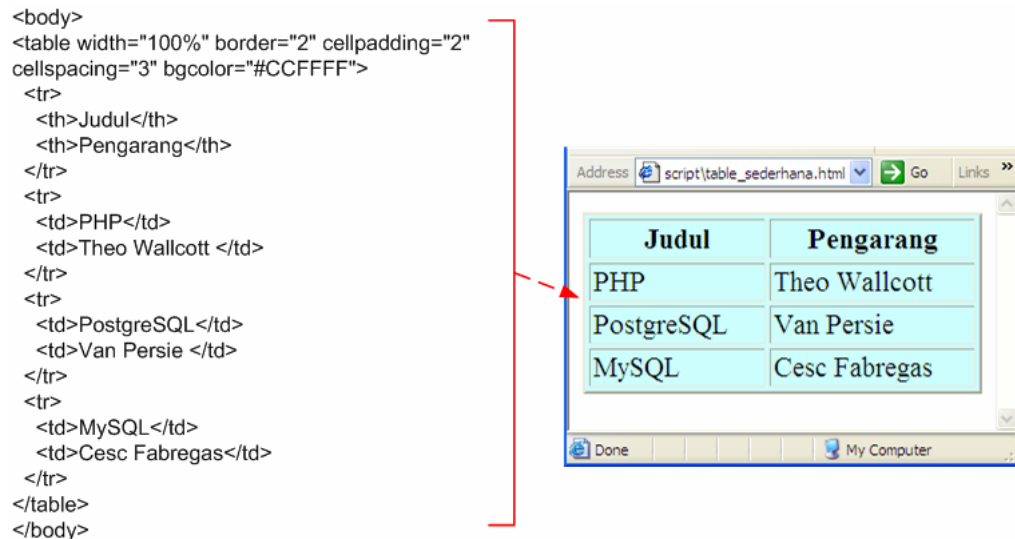
Tag	Fungsi
<b>&lt;CAPTION&gt;...&lt;/CAPTION&gt;</b>	Membentuk judul tabel
<b>&lt;TH&gt;...&lt;/TH&gt;</b>	Membuat judul kolom
<b>&lt;TR&gt;...&lt;/TR&gt;</b>	Membentuk baris pada suatu tabel
<b>&lt;TD&gt; . . . &lt;/TD&gt;</b>	Membuat sebuah sel data

Contoh-contoh penggunaan tabel adalah sebagai berikut:



Gambar 13.31. Tabel sederhana.

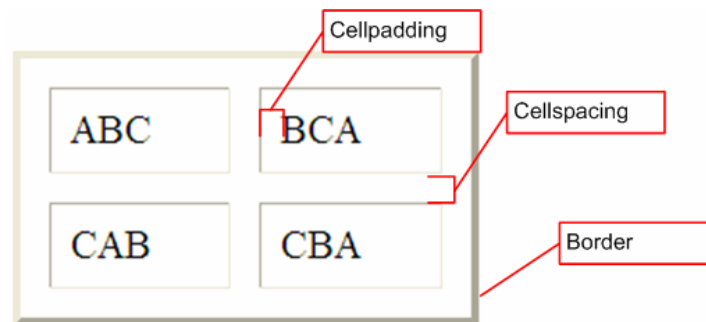
Pada Gambar 13.31, tabel yang kita buat adalah tabel sederhana dengan dua buah kolom dan 3 buah baris (perhatikan ada 3 buah pasangan tag `<TR>` ... `</TR>`). Secara default tabel ditampilkan tanpa ada garis pada tabel tersebut. Kita dapat menambahkan garis dengan menggunakan atribut `border` pada tabel (lihat Gambar 13.32).



Gambar 13.32. Tabel dengan format yang lebih kompleks.

Pada Gambar 13.32, terlihat tabel yang tampilannya lebih baik daripada Gambar sebelumnya. Ada beberapa atribut yang kita tambahkan pada tabel yaitu :

- Atribut **WIDTH** untuk mengatur lebar tabel pada halaman. Kita dapat menggunakan satuan persen (%) atau pixel (px).
- Atribut **BORDER** untuk memberikan garis pada tabel. Nilai untuk atribut ini dari dimulai dari 0 yang berarti tidak ada garis. Semakin besar angka semakin tebal garis.
- Atribut **BGCOLOR** untuk menambahkan warna latar belakang pada tabel.
- Atribut **CELLPADDING** untuk menentukan jarak antara teks dan tepi kiri sebuah sel (lihat gambar 13.33 untuk lebih jelasnya).
- Atribut **CELLSPACING** untuk menentukan jarak bagian sel terhadap tepi dalam bingkai tabel (lihat gambar 13.33 untuk lebih jelasnya).



Gambar 13.33. Cellpadding, cellspacing dan border.

Seperti halnya perangkat lunak *word-processor*, pada HTML kita dapat menggabungkan dua atau lebih sel menjadi satu buah sel. Untuk menggabungkan baris dapat digunakan atribut **ROWSPAN** dan untuk menggabungkan kolom dapat digunakan atribut **COLSPAN**. Contoh penggunaannya dapat dilihat pada Gambar 13.34 dan 13.35.

```

<body>
<table width="100%" border="2"
cellspacing="2" cellpadding="2">
  <tr>
    <td>Propinsi</td>
    <td>Kota</td>
  </tr>
  <tr>
    <td rowspan="2">Jawa Timur </td>
    <td>Surabaya</td>
  </tr>
  <tr>
    <td>Malang</td>
  </tr>
  <tr>
    <td rowspan="2">Jawa Tengah </td>
    <td>Semarang</td>
  </tr>
  <tr>
    <td>Solo</td>
  </tr>
</table>
</body>

```

Propinsi	Kota
Jawa Timur	Surabaya
	Malang
Jawa Tengah	Semarang
	Solo

Gambar 13.34. Rowspan.

```

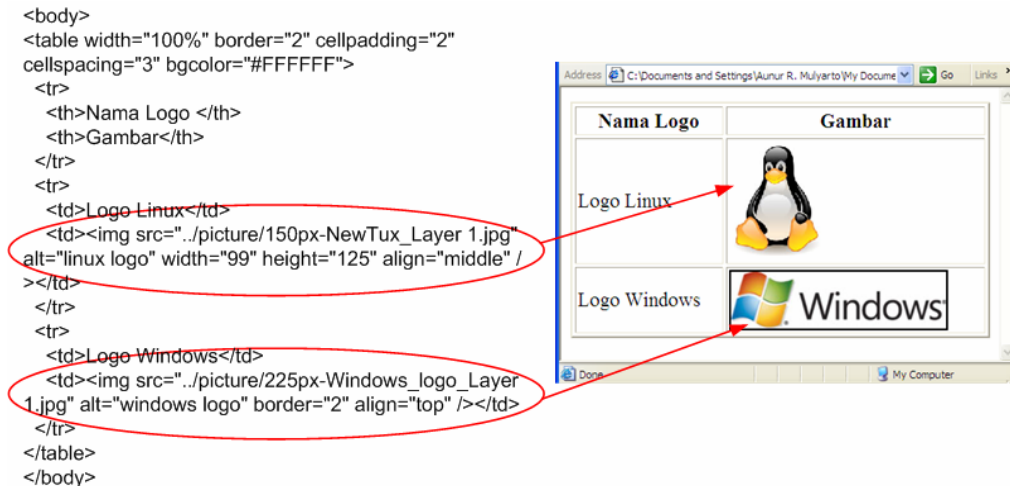
<body>
<table width="100%" border="2"
cellspacing="2" cellpadding="2">
  <tr>
    <td colspan="2">LUAS AREA KOTA </td>
  </tr>
  <tr>
    <td>Surabaya</td>
    <td>12000</td>
  </tr>
  <tr>
    <td>Malang</td>
    <td>11000</td>
  </tr>
  <tr>
    <td>Semarang</td>
    <td>13000</td>
  </tr>
</table>
</body>

```

LUAS AREA KOTA	
Surabaya	12000
Malang	11000
Semarang	13000

Gambar 13.35. Colspan.

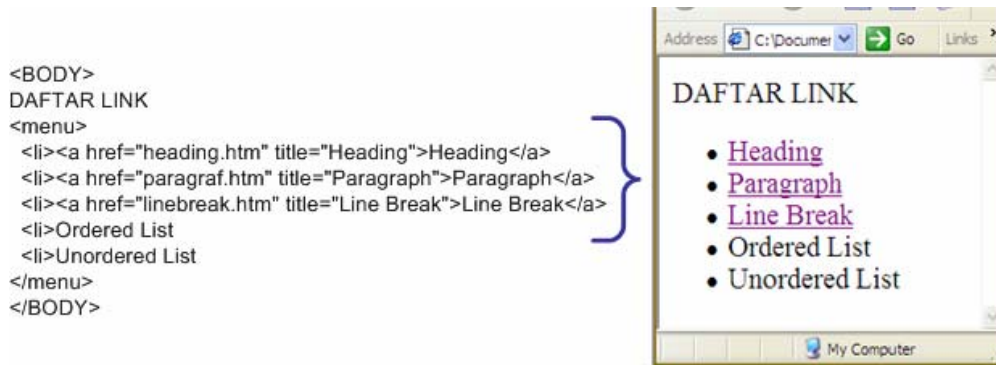
Sel pada tabel tidak selalu harus berisi teks namun dapat juga berisi gambar seperti terlihat pada Gambar 13.36.



Gambar 13.36. Tabel dengan sel berisi gambar.

#### 13.4.6.Link antar Dokumen

*Link* merupakan pautan untuk membuka atau memanggil halaman *web* atau file tertentu. *Link* merupakan tag yang sangat penting dalam penggunaan HTML, karena disinilah letak perbedaan antara dokumen HTML dengan dokumen teks yang lain. *Link* dapat dibuat dengan memberi perintah tag *anchor* <A>. *Anchor* memiliki beberapa atribut, diantaranya **HREF** yang berfungsi untuk membuat link ke dokumen HTML tertentu dan **NAME** yang berfungsi untuk memberi tanda/nama titik tertentu pada dokumen HTML yang sama. Contoh penggunaan tag *anchor* dapat dilihat pada Gambar 13.37.



Gambar 13.37. Penggunaan tag anchor.

Atribut **HREF** dapat digunakan untuk memanggil halaman *web* pada sistem yang sama (pada satu komputer) seperti ditunjukkan pada Gambar 13.37. Pada kasus ini kita tinggal menuliskan lokasi dimana halaman yang akan kita

panggil berada. **HREF** dapat juga kita gunakan untuk memanggil halaman lain diluar sistem kita atau memanggil situs-situs lain di internet. Caranya dengan mengetikkan alamat URL situs yang akan kita panggil. Cobalah ketikkan kode HTML berikut ini dengan teks editor kemudian simpan file dengan ekstensi .htm.

#### DAFTAR ALAMAT MESIN PENCARI

```
<menu>
  <li><a href="http://www.google.com/">Google</a>
  <li><a href="http://www.yahoo.com/">Yahoo</a>
  <li><a
href="http://www.altavista.com/">Altavista</a>
</menu>
```

### 13.5. RINGKASAN

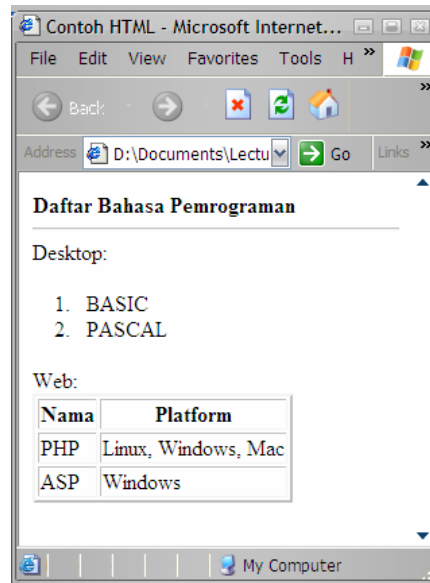
- Teknologi disain *web* terbagi menjadi beberapa layer (lapisan), yaitu structural layer, presentation layer dan behavioral layer.
- *Web* statis adalah halaman *web* yang isi data dan informasinya tidak berubah-ubah. Sedangkan *web* dinamis, memiliki isi data dan informasi yang berbeda-beda tergantung input apa yang disampaikan *client*.
- Pembuatan halaman *web* membutuhkan dukungan persiapan perangkat keras, perangkat lunak dan pemahaman teknologi pembuatan *web*.
- HTML (*Hypertext Markup Language*) merupakan bahasa yang digunakan untuk pembuatan halaman *web* dilakukan dengan cara disisipkan (*embedded language*) pada dokumen dengan memberi tanda tertentu yang disebut **tag**.
- Dokumen HTML secara umum akan terdiri dari dua bagian yaitu Header dan Body.
- HTML menyediakan tag-tag untuk pendeskripsian dokumen dan format dokumen yang lengkap.
- Pada halaman HTML dapat ditambahkan gambar, tabel, suara, atau file-file lain.

### 13.6. SOAL-SOAL LATIHAN

1. Apakah perbedaan antara *web* statis dan *web* dinamis?
2. Kunjungilah beberapa situs, kemudian cermatilah apakah halaman-halaman *web* yang tersedia termasuk halaman *web* statis ataukah dinamis?
3. Bukalah sebuah situs tertentu kemudian dari menu bar *web browser* kalian, pilih View kemudian pilih Source. Pada jendela yang muncul, cermatilah kode-kode HTML yang muncul. Cari bagian mana yang merupakan kode-kode untuk format dokumen, tabel, gambar, dan elemen-elemen HTML yang lain.



4. Buatlah halaman HTML seperti berikut.



5. Buat lagi halaman *web* seperti berikut.

