



ITS
Institut
Teknologi
Sepuluh Nopember

Buku Ajar

Pemrograman Berbasis Web

Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Dwi Sunaryono SKom, Mkom
Diterbitkan 2010

Daftar Isi

Perkembangan Teknologi Web	1
1.1 Sejarah Internet dan WWW (World Wide Web) Consortium (W3C)	1
1.2 Aplikasi Internet	1
1.3 Cara Kerja WWW	3
1.4 Browsing	3
1.5 TraceRoute	4
1.6 Ping	6
1.7 Web 2.0, Penjelasan fitur-fitur pada web 2.0	7
Dasar-dasar HTML	13
2.1 Standar Dokumen	14
2.2 Warna Halaman	16
2.3 Tautan Halaman	17
2.4 Image	18
2.5 Penampilan teks	18
2.6 Lists	19
2.7 Horizontal Rules	20
2.8 Tabel	20
2.9 Frame	22
2.10 Form	23
XHTML	29
3.1 Versi XHTML	29
3.2 Kategori XHTML	30
3.3 Aturan XHTML	30
3.4 Penggunaan XHTML	32
Cascading Style Sheets (CSS)	36
4.1 Pemakaian Secara Inline	36
4.2 Pemakaian Tag Style	37
4.3 Pemakaian Eksternal Style Sheet	39
4.4 Pemakaian Style Sheet	40
4.5 Pemakaian Elemen Style Sheet	41
JAVA SCRIPT	47
5.1 Memakai JavaScript	47
5.2 Konsep Memori	49
5.3 Operasi Aritmatika Pada JavaScript	49
5.4 Struktur Kontrol Pada JavaScript	50
5.5 Event Pada JavaScript	55
5.6 Fungsi Pada JavaScript	55
5.7 Hirarki Pada JavaScript	57
5.8 Location-obyek Pada JavaScript	60
5.9 Frame dan JavaScript	60
5.10 Frame dan JavaScript	61
5.11 Predefined obyek JavaScript	63
Pengenalan ASP.NET	67
6.1 Sekilas tentang .NET	67
6.2 Framework .NET 4.0	68
6.3 Tujuan ASP.NET	70
6.4 Web Services	71
6.5 Halaman Awal	71
6.6 Bekerja dengan Label, TextBox dan Button	74

Halaman Dinamis dengan ASP.NET	76
7.1 Halaman ASPX secara Dinamis	76
7.2 Bekerja Dengan Visual Studio 2010	78
7.3 Setting IIS pada Windows 7	80
7.4 File ASPX Sebagai Script	82
7.5 Kelas yang dibangkitkan dari File ASPX	85
Web Form pada ASP.NET	87
8.1 Halaman Form HTML	87
8.2 Halaman Form pada ASPX	88
8.3 Notasi Umum pada Kontrol Halaman Web	93
8.4 Keuntungan dari Web Form pada ASPX	93
8.5 Proses Berdasarkan Event	95
Webcontrol pada ASP.NET (1)	98
9.1 Hierarki Webcontrol pada ASP.NET	98
9.2 Kelas kontrol	99
9.3 Kelas Webcontrol	101
9.4 Kelas Button	105
9.5 Kelas TextBox	107
Web Control pada ASP.NET (2)	109
10.1 Kelas CheckBox	109
10.2 Kelas RadioButton	111
10.3 Kelas ListControl	114
DataGrid	124
11.1 Kelas DataGrid	124
11.2 Konfigurasi kelas DataGrid	126
11.3 Method dan events pada Kelas DataGrid	127
Validasi pada ASP.NET	134
13.1 Validasi pada ASP.NET	134
13.2 UserControl	146
State Management	149
13.1 Tiga Jenis State	149
13.2 Manajemen State Secara Lengkap	154
ADO.NET (1)	158
14.1 Arsitektur ADO.NET	160
14.2 Akses Secara Connection Oriented	161
ADO.NET (2)	169
15.1 DataSet	170
15.2 Penentuan DataSet vs DataReader	175
15.3 DataView	175
15.4 DataAdapter	177
Konfigurasi Keamanan	179
16.1 Autorisasi	181
16.2 Autentikasi	182
16.3 Kontrol Login	184

Daftar Gambar

Gambar 1.1 Server-Client	2
Gambar 1.2 Konfigurasi alamat Internet	3
Gambar 1.3 Mencoba koneksi ke halaman web.....	3
Gambar 1.4 Mencoba koneksi ke halaman web.....	4
Gambar 1.5 Cara Melihat Source Web	6
Gambar 1.6 Hasil source script HTML.....	7
Gambar 1.7 Ruang Lingkup Web 2.0	7
Gambar 1.8 Situs Web 1.0 antara Web 2.0.....	8
Gambar 1.9 Content Network.....	9
Gambar 1.10 Tagging clouds.....	10
Gambar 1.11 Arsitektur WebTop	11
Gambar 1.12 Arsitektur WebTop	12
Gambar 2.1 Hasil pengaturan warna halaman	17
Gambar 2.2 Hasil list dengan Penomoran	19
Gambar 2.3 Hasil list tanpa Penomoran.....	19
Gambar 2.4 Hasil Tabel	20
Gambar 2.5 Tabel dengan Pengaturan	21
Gambar 2.6 Tabel dengan Pengaturan lebih kompleks	22
Gambar 3.1 Hasil XHTML	33
Gambar 3.2 Elemen Header	33
Gambar 3.3 Pertautan.....	34
Gambar 3.4 Tabel Sederhana.....	35
Gambar 4.1 Halaman CSS dengan Tag p.....	37
Gambar 4.2 Halaman CSS dengan Tag span.....	37
Gambar 4.3 Halaman CSS dengan Tag style.....	38
Gambar 4.4 Halaman CSS dengan eksternal Tag style	39
Gambar 4.5 Pengaturan absolute dan z-index.....	42
Gambar 4.6 Hasil pengaturan posisi relative	42
Gambar 4.7 Hasil pengaturan dengan latar belakang gambar.....	43
Gambar 4.8 Hasil pengaturan dengan latar belakang gambar.....	44
Gambar 4.9 Model Box	45
Gambar 4.10 Hasil pengaturan dengan Text Mengembang.....	46
Gambar 5.1 Interaksi dengan Pengguna	49
Gambar 5.2 Tulisan Header.....	54
Gambar 5.3 Element Halaman Web.....	57
Gambar 5.4 Hirarki Element Halaman Web	58
Gambar 5.5 Hasil program Java script.....	59
Gambar 5.6 Pembagian Frame Halaman Web.....	60
Gambar 5.7 Pembagian FrameSet Halaman Web	61
Gambar 5.8 Dua Frame	61
Gambar 5.9 Hubungan Browswe ke Frame.....	62
Gambar 5.10 Hubungan Frame ke Browser	63
Gambar 5.11 Hubungan antar Frame	63
Gambar 6.1 Arsitektur Aplikasi lama	67
Gambar 6.2 Arsitektur Aplikasi BerbasisPustaka	67
Gambar 6.3 Arsitektur Aplikasi ASP.NET.....	68
Gambar 6.4 Arsitektur framework .NET.....	69
Gambar 6.5 Garis besar framework	69
Gambar 6.6 Program desktop dan Web versi lama	70
Gambar 6.7 Program desktop dan Web versi ASP.NET	70
Gambar 6.8 Kompabilitas ASP.NET	70
Gambar 6.9 Aplikasi B2B	71
Gambar 6.10 Pilihan Lingkungan Kerja Visual Studio 2010	71
Gambar 6.11 Pilihan Framework dan template VS 2010	72
Gambar 6.12 Halaman Kode Sumber pada VS 2010	72
Gambar 6.13 Halaman Desain VS 2010.....	73

Gambar 6.14 Halaman hasil debugging pada browser	73
Gambar 6.15 Komponen pada ToolBox	74
Gambar 6.16 Rancangan dengan Label, TextBox dan Button	74
Gambar 6.17 Kode Sumber dari Label, TextBox dan Button	75
Gambar 6.18 Tampilan Hasil	75
Gambar 7.2 Hasil Browser dari file My.html	77
Gambar 7.3 Buat Situs Baru	78
Gambar 7.4 Editor Pilihan Situs Baru	78
Gambar 7.5 Jendela Solution dan Properti	79
Gambar 7.6 Menu New File	79
Gambar 7.7 Pembuatan Program My.aspx	79
Gambar 7.8 Menu Start Debugging	80
Gambar 7.9 Hasil Program My.aspx	80
Gambar 7.10 Cara Stop Debugging	80
Gambar 7.11 Setting Windows Features	81
Gambar 7.12 Setting IIS	81
Gambar 7.13 Regsiter FrameWork .NET	82
Gambar 7.14 Cara Kerja Web Server IIS	82
Gambar 7.14 File My.aspx.cs	84
Gambar 7.15 File My.aspx pada VS2010	84
Gambar 7.16 File My.aspx.cs pada VS2010	84
Gambar 7.17 Pembangkitan page Class	85
Gambar 7.18 Pembangkitan dua page Class	86
Gambar 8.1 Hasil Kode Form	88
Gambar 8.2 Menu File Baru	88
Gambar 8.3 Pilihan template	89
Gambar 8.4 Rancangan MyAccount.aspx	89
Gambar 8.5 Jendela Properti Button1	90
Gambar 8.6 Jendela Properti <i>Events</i> Button1	90
Gambar 8.7 Jendela Kode Sumber MyAccount.aspx.cs	91
Gambar 8.8 Hasil program MyAccount	91
Gambar 8.9 Hasil Kode Form Default2.aspx	92
Gambar 8.10 Web Control	94
Gambar 8.11 Hierarki Web Control	94
Gambar 8.12 Proses Event mouse click	95
Gambar 8.13 Proses Event Program Default2.aspx	96
Gambar 8.14 Inisial nilai pada control	96
Gambar 8.15 Inisial nilai pada control	96
Gambar 8.16 Tahap rendering	97
Gambar 9.1 Hierarki Webcontrol	99
Gambar 9.2 Pewarisan WebControl	100
Gambar 9.3 Hubungan antar kontrol	101
Gambar 9.4 Visualisasi BorderStyle	104
Gambar 9.5 Visualisasi Enabled	104
Gambar 9.6 Visualisasi TabIndex	104
Gambar 9.7 Jendela Pembuatan MyCommand.aspx	106
Gambar 9.8 Rancangan MyCommand.aspx	106
Gambar 9.9 Membuka File MyCommand.aspx.cs	107
Gambar 9.11 Visualisasi Pemakaian Kelas TextBox	108
Gambar 10.1 Pembuatan File MyFruits.aspx	110
Gambar 10.2 Rancangan MyFruits.aspx	110
Gambar 10.3 Visualisasi Program MyFruits.aspx	111
Gambar 10.4 Rancangan Program MyPayment.aspx	112
Gambar 10.5 Visualisasi RadioButton	113
Gambar 10.6 Rancangan Program MyCountry.aspx	115
Gambar 10.7 Visualisasi ListBox	116
Gambar 10.8 Rancangan program MyCountry2.aspx	116
Gambar 10.9 Visualisasi ListBox dengan pengisian data dari Server	118
Gambar 10.10 Rancnagan Program MyCountry3.aspx	118
Gambar 10.11 Hasil ListBox dengan Key dan data sama	120

Gambar 10.12 Rancangan program MyCountry4.aspx	120
Gambar 10.13 Rancangan program MyCountry5.aspx	122
Gambar 10.14 Visualisasi DropDownList.....	123
Gambar 11.1 Visualisasi DataGrid	125
Gambar 11.2 Pembuatan File MyDataGrid.aspx	126
Gambar 11.3 Jenis events pada kolom DataGrid.....	128
Gambar 11.4 Hasil Visualisasi datagrid sederhana.....	130
Gambar 11.5 Hasil Visualisasi datagrid sederhana.....	133
Gambar 12.1 Halaman Login	136
Gambar 12.2 Halaman Login dengan pesan kesalahan	137
Gambar 12.3 Validasi Halaman Login.....	140
Gambar 12.4 Validasi Email	140
Gambar 12.5 Validasi Nilai.....	143
Gambar 12.6 Validasi Memakai ValidationSummary	146
Gambar 12.7 Kelompok kontrol dengan satu kendali	147
Gambar 12.8 Dua user control pada satu halaman.....	148
Gambar 13.1 Jenis State.....	150
Gambar 13.2 Visualisasi contoh inputan.....	152
Gambar 13.3 Hasil Visualisasi contoh inputan.....	153
Gambar 13.4 Visualisasi test1.aspx	154
Gambar 13.5 Visualisasi Welcome.aspx	154
Gambar 14.1 Hubungan Aplikasi dan Sumber Data	158
Gambar 14.2 Peran Provider	159
Gambar 14.3 Arsitektur ADO.NET	160
Gambar 14.4 Model ADO.NET Connected.....	161
Gambar 14.5 Kelas interface connection.....	163
Gambar 14.6 Kelas interface command	164
Gambar 14.7 Kelas interface Parameter.....	167
Gambar 14.8 Kelas interface Transaksi.....	167
Gambar 15.1 Arsitektur Microsoft 3-Tier	170
Gambar 15.3 Model DataSet	171
Gambar 15.4 Kelas Diagram DataSet	171
Gambar 15.5 Pemetaan konsep ke dataset.....	172
Gambar 15.6 State DataSet	175
Gambar 15.7 Kelas Diagram DataView.....	176
Gambar 15.8 Kelas Diagram DataAdapter.....	177
Gambar 16.1 Struktur Direktori IIS	179
Gambar 16.2 Pengaturan konfigurasi.....	179
Gambar 16.3 File web.config pada VS2010.....	180
Gambar 16.4 Pembacaan file web.config.....	181
Gambar 16.5 Pesan Error Autorisasi.....	181
Gambar 16.6 Halaman Login.aspx	183
Gambar 16.7 Siklus Cookie pada Login.aspx.....	184
Gambar 16.8 Halaman memakai kontrol login.....	185

Daftar Tabel

Tabel 2.1 Daftar Penampilan Teks.....	15
Tabel 2.2 Daftar Format Keluaran Komputer.....	15
Tabel 2.3 Daftar Karakter Entitas.....	16
Tabel 2.4 Daftar Pemakaian Kode Warna.....	16
Tabel 2.5 Daftar Atribut Horizontal Rule.....	20
Tabel 2.6 Daftar Atribut Tabel.....	21
Tabel 4.1 Daftar Pengaturan Atribut CSS.....	40
Tabel 5.1 Operasi Aritmatika.....	49
Tabel 5.2 Derajat Operasi.....	50
Tabel 8.1 Event Umum.....	95
Tabel 11.1 Setting Tampilan DataGrid.....	127
Tabel 11.2 Setting Tampilan DataGrid.....	127
Tabel 11.3 Jenis Kolom pada Kelas DataGrid.....	128
Tabel 12.1 Validasi Server Control.....	136
Tabel 12.2 Ekspresi Batasan Ekspresi.....	141
Tabel 12.3 Ekspresi Karakter kelas.....	141
Tabel 12.4 Ekspresi Karakter Berbilang.....	142
Tabel 14.1 Perbedaan ADO dan ADO.NET.....	159
Tabel 14.2 Jenis Isolasi.....	168
Tabel 15.1 Ekspresi Karakter Berbilang.....	174

Bab 1

Perkembangan Teknologi Web

1.1 Sejarah Internet dan WWW (World Wide Web) Consortium (W3C)

Internet (Inter-Network) adalah sekumpulan jaringan komputer yang menghubungkan situs akademik, pemerintahan, komersial, organisasi, maupun perorangan. Istilah untuk jaringan global yang memakai protokol-protokol tertentu untuk mengatur pengiriman data. Dalam pengertian global, internet adalah sebuah Public International Network of Network yaitu jaringan yang menghubungkan banyak sekali jaringan lokal yang boleh dipakai secara umum. Untuk tersambung ke jaringan internet, pengguna harus menggunakan layanan khusus seperti :

- ISP - Internet Service Provider
- ISDN - Integrated System Digital Network
- ADSL - Asyemetric Digital Subscriber Line
- VSAT - Very Small Aperture Terminal
- Dan sebagainya

1.2 Aplikasi Internet

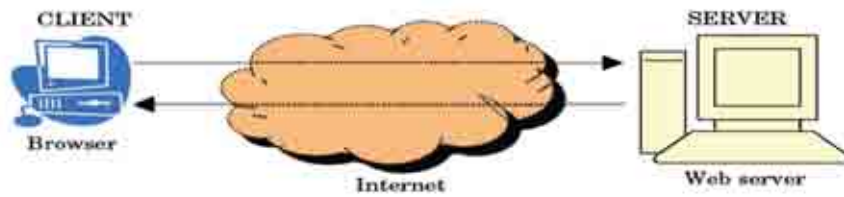
Aplikasi Internet yang terdapat secara umum pada saat ini adalah :

- WWW (World Wide Web)
- Electronic Mail/Email/Messaging
- File Transfer
- Remote Login
- IRC (Internet Relay Chat)

1.2.1 WWW (World Wide Web)

Dewasa ini, WWW atau yang sering disebut sebagai "web" merupakan sebuah sistem dimana informasi dalam bentuk teks, gambar, suara, dan lain-lain yang tersimpan dalam sebuah internet webserver dipresentasikan dalam bentuk hypertext. Informasi di web dalam bentuk teks umumnya ditulis dalam format HTML (Hypertext Markup Language), seperti yang ditunjukkan pada gambar 1.1.

Web dapat diakses oleh perangkat lunak web client yang secara populer disebut sebagai browser. Beberapa browser yang populer adalah Microsoft Internet Explorer, Netscape Navigator, Opera, Mozilla Firefox



Gambar 1.1 Server-Client

1.2.2 Electronic Mail/Email/Messaging

Email atau kalau dalam istilah Indonesia, surat elektronik, adalah aplikasi yang memungkinkan para pengguna internet untuk saling berkirim pesan melalui alamat elektronik di internet. Layanan email biasanya dikelompokkan dalam dua basis, yaitu email berbasis client dan email berbasis web.

Bagi pengguna email berbasis client, aktifitas per-emailan dilakukan dengan menggunakan perangkat lunak email client, misalnya Eudora atau Outlook Express. Perangkat lunak ini menyediakan fungsi-fungsi penyuntingan dan pembacaan email secara offline. Bagi pengguna email berbasis web, seluruh kegiatan per-emailan harus dilakukan melalui suatu situs web. Dengan demikian, untuk menggunakannya haruslah dalam keadaan online.

1.2.3 File Transfer

Fasilitas ini memungkinkan para pengguna internet untuk mengunggah (*upload*) atau mengunduh (*download*) sebuah file antara komputer lokal dengan komputer lain yang terhubung dalam jaringan internet. Protokol standar yang digunakan untuk keperluan ini disebut sebagai File Transfer Protocol (FTP)

1.2.4 Remote Login

Layanan remote login mengacu pada program atau protokol yang menyediakan fungsi yang memungkinkan seorang pengguna internet untuk mengakses (login) ke sebuah terminal (remote host) dalam lingkungan jaringan internet. Dengan memanfaatkan remote login, seorang pengguna internet dapat mengoperasikan sebuah host dari jarak jauh tanpa harus secara fisik berhadapan dengan host bersangkutan

Protokol yang umum digunakan untuk keperluan remote login adalah Telnet (Telecommunications Network). Namun telah dikembangkan protokol SSH (secure shell) untuk menggantikan Telnet dalam melakukan remote login. Dengan memanfaatkan SSH, maka paket data antar host akan dienkripsi (diacak) sehingga apabila "disadap" tidak akan menghasilkan informasi yang berarti bagi pelakunya.

1.2.5 IRC (Internet Relay Chat)

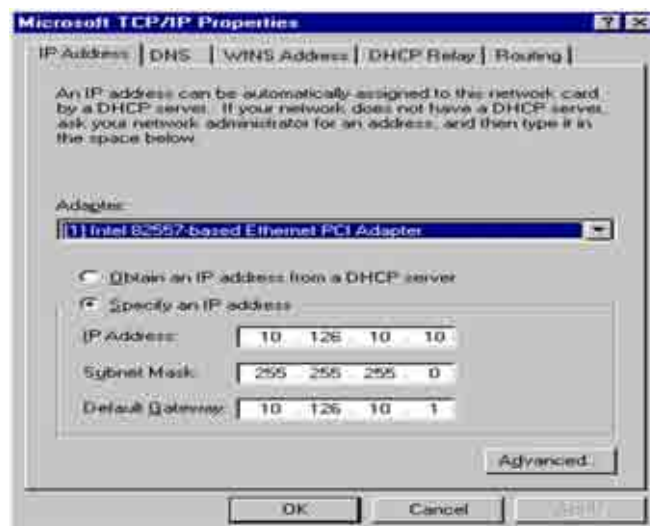
Layanan IRC, atau biasa disebut sebagai "chat" saja adalah sebuah bentuk komunikasi di internet yang menggunakan sarana baris-baris tulisan yang diketikkan melalui keyboard. Contoh software nya adalah software mIRC

1.3 Cara Kerja WWW

Internet merupakan komunikasi global antar komputer di dunia. Informasi yang terekam di sebarakan lewat internet berupa halaman-halaman/ page yang disebut web. Konfigurasi web bisa dilakukan dengan dua cara yaitu :

- Mengetikkan script web (HTML/Hypertext Markup Language)
- Dengan tools/alat bantu spt front page, dream weaver dll

Semua konfigurasi web akan menghasilkan script berupa HTML, sehingga penguasaan script dasar menjadi sangat penting. Mekanisme halaman web, dari komputer client (browser), yang mengetikkan alamat URL (Uniform Request Locator), contohnya www.sciencedirect.com, maka komputer client akan mencoba koneksi dengan mengubah (resolve) alamat tersebut menjadi angka yang di sebut IP (internet protocol) .Koneksi tersebut, melalui gateway (jembatan) yang telah diset di komputer client, sehingga dengan hubungan antar gateway, maka sampai ke tujuan. *Contoh kasus* : Alamat IP client : 10.126.10.1 dengan gateway 10.126.10.1, seperti yang ditunjukkan pada gambar 1.2



Gambar 1.2 Konfigurasi alamat Internet

1.4 Browsing

Internet menyediakan halaman-halaman informasi yang bisa dikunjungi dengan mengetikkan alamat web seperti contoh mengetikkan alamat www.sciencedirect.com, dengan mozilla misalnya, akan muncul halaman web, seperti yang ditunjukkan pada gambar 1.3 :



Gambar 1.3 Mencoba koneksi ke halaman web

Setelah mencoba koneksi, hasil lengkap ditampilkan halaman web secara menyeluruh seperti gambar 1.4. Kecepatan penampilan ini tergantung dari beberapa parameter :

- Jarak lompatan sambungan (*hop*)
- Banyaknya lompatan sambungan
- Kualitas server (komputer penyedia)
- Banyaknya user yang mengakses
- Kecepatan jalur komunikasi



Gambar 1.4 Mencoba koneksi ke halaman web

1.5 TraceRoute

Salah satu cara untuk melihat *hop* dengan perintah di windows : *tracert*, contohnya adalah : (mencoba jalur sambungan/ *trace route*). Dari hasil *tracert* , didapat jumlah hop dan jumlah waktu untuk mencapai koneksi ke situs yang dikehendaki, semakin pendek hop, maka semakin cepat penyampaian data. Sering kali didapat hasil *request time-out*, hal ini berarti waktu yang ditetapkan oleh perintah *tracert* (default 3 detik), tidak mendapatkan respon dari server yang dikehendaki, dan akan dicoba ulang sampai biasanya 3 kali pengulangan. Format lengkap perintah *tracert* adalah :

```
tracert [-d] [-h maximum_hops] [-j komputer-list] [-w timeout]
target_name
```

Kegunaan *tracert* umumnya dipakai untuk alat diagnosa yang dapat menentukan rute/jalur yang dipakai ke tujuan dengan mengirimkan paket - paket sederhana (echo) berupa *Internet Control Message Protocol (ICMP)* dengan mengetahui nilai waktu hidup tujuan atau *Time-To-Live (TTL)* values. Sepanjang jalur yang dipakai, maka nilai TTL akan diturunkan 1 sebelum diteruskan ke router yang lain. Dengan demikian TTL dapat dianggap sebagai hitungan hop

Beberapa Parameter

```
-d :          alamat nomer tidak diubah
             menjadi nama komputer
-h maximum_hops:  menentukan jumlah maksimum hop
                 untuk mencapai target
-w timeout      jumlah waktu habis untuk
                 menunggu jawaban
target_name     : nama komputer[situs]
```

```
D:\>tracert www.yahoo.com
Tracing route to www.yahoo.akadns.net [204.71.200.75]
over a maximum of 30 hops:
  1  <10 ms  <10 ms  <10 ms
      upgw-10.tcnet.its-sby.edu [10.126.10.1]
  2  <10 ms  <10 ms  <10 ms
      gateway.its-sby.edu [203.130.251.17]
  3  171 ms  235 ms  78 ms  192.168.24.13
  4  329 ms  390 ms  344 ms  202.134.3.254
  5  641 ms  750 ms  594 ms
      hssi2-0-gw2.hk.telkom.net.id [202.134.3.10]
  6  1281 ms  812 ms  *  207.176.98.81
  8  1422 ms  1406 ms  *
      paix1.globalcenter.net [198.32.176.29]
  9  2891 ms  1312 ms  969 ms
      sol-0-0-622M.crl.pao2.gambarlx.net
      [208.50.169.141]
 10  *  1235 ms  1515 ms
      pos6-0-2488M.crl.SNV.gambarlx.net
      [208.50.169.62]
 11  1407 ms  1125 ms  1187 ms
      ge0-0-1000M.hr8.SNV.gambarlx.net
      [206.132.254.37]
 12  984 ms  891 ms  968 ms
      bas1r-ge3-0-hr8.snv.yahoo.com [208.178.103.62]
 13  1812 ms  1282 ms  953 ms
      www10.yahoo.com [204.71.200.75]
```

Pertanyaan menarik lainnya adalah bagaimana mengetahui apakah jaringan kita terkoneksi dengan alamat IP/ nomer komputer atau ke situs web secara langsung. Untuk berbagai macam sistem operasi dikenal perintah *ping* khusus di windows, maka perintah defaultnya adalah :

```
ping [-t] [-a] [-l length] [-i ttl] [-w timeout] destination-list
```

Perintah *ping* bertugas untuk memastikan hubungan dengan komputer lain atau bahkan komputer dirinya sendiri apakah setting penomoran sudah betul (*TCP/IP*). Perintah ini tersedia hanya jika protokol *TCP/IP* (dasar komunikasi internet adalah penomoran secara *TCP/IP*) telah diinstal

Parameter perintah *ping*

```
-t :          ping terus-menerus sampai di break oleh
              penekanan tombol Ctrl-C
-a :          ubah alamat nomor ke nama komputer
-l length:   kirim paket sederhana (ECHO) yang berisi
              sejumlah data yang ditentukan panjangnya, default adalah 64
              bytes; maksimum is 8192.
-i ttl:      setting waktu hidup paket (TTL)
-w timeout:  penentuan waktu interval timeout dalam
              milli detik
destination-list : penentuan tujuan komputer yang diping
```

Secara teknis perintah *ping* mengirimkan paket sederhana *Internet Control Message Protocol* (*ICMP*) ke komputer dan menunggu jawaban (*reply*) dari komputer lain. *Ping* menunggu tiap 1 detik untuk tiap paket yang dikirim dan menayangkan ke layar berapa paket yang dikirim dan diterima kembali. Tiap paket yang diterima divalidasi ulang terhadap data yang dikirimkan. Secara default, empat paket sederhana berisi 64 bytes dari data yang ditransmisikan.

1.6 Ping

Perintah ping dapat dipakai untuk cek nama komputer ataupun nomernya (IP). *Contoh berikut ini*

```
D:\>ping www.plasa.com

Pinging www.plasa.com [202.134.0.172] with 32 bytes of data:

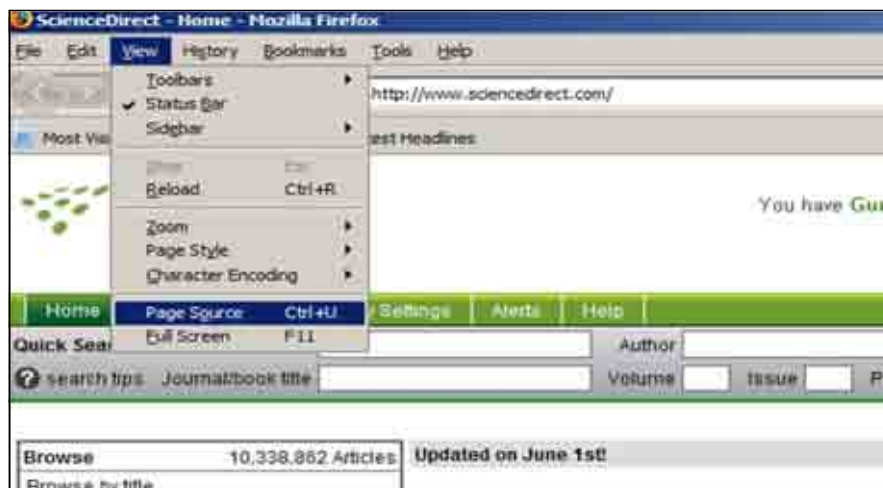
Reply from 202.134.0.172: bytes=32 time=688ms TTL=249
Reply from 202.134.0.172: bytes=32 time=1000ms TTL=249
Reply from 202.134.0.172: bytes=32 time=734ms TTL=249
Reply from 202.134.0.172: bytes=32 time=500ms TTL=249
```

Jika komputer tidak terhubung atau terhubung sangat jauh, maka akan muncul *request time-out*. Contohnya :

```
F:\>ping www.yahoo.com
Pinging www.yahoo.akadns.net [204.71.202.160] with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.
```

Pengertian *browsing* (menampilkan data secara menyeluruh) adalah pihak pemakai mengunjungi situs (alamat web) tertentu dan mendapatkan informasi yang dikehendaki. Halaman web yang ditampilkan berisi kode-kode script HTML. Halaman web dapat dilihat kode scriptnya dengan cara pilih menu View->page source atau Ctrl-U, seperti yang diperlihatkan pada gambar 1.5.



Gambar 1.5 Cara Melihat Source Web

Dari perintah diatas, kemudian muncul source berupa kode-kode HTML seperti yang terlihat pada gambar 1.6

```

Source sh: http://www.sciencedirect.com/ - Mozilla Firefox
File Edit View Help

<html>
<head>
<meta name="verify-v1" content="y1hzga1QU15jtnD+ZERS1JvGn59+WmE+y40NoGeRtU

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<!-- DRAMKEY: 08/22/2010 02:04:38 EDT#2861_008#08797#122.167.12.27 -->
<meta name="robots" content="NOARCHIVE">
<link rel="shortcut icon" href="/servlet/afavicon3D.ico" type="image/x-ico

<!-- scienceHome.tpl -->
<title>ScienceDirect - Home
</title>
<link rel="stylesheet" title="Default Styles" href="http://www.sciencedire

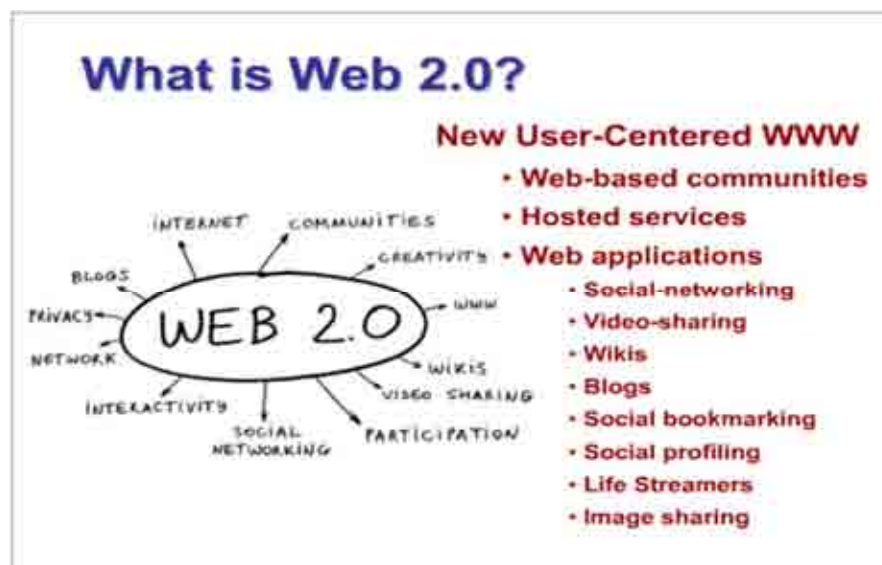
</head>
<body>
<noscript>
<style type="text/css">
.nojs { display: none; }
</style>

```

Gambar 1.6 Hasil source script HTML

1.7 Web 2.0, Penjelasan fitur-fitur pada web 2.0

Web 2.0, adalah sebuah istilah yang dicetuskan pertama kali oleh O'Reilly Media pada tahun 2003, dan dipopulerkan pada konferensi web 2.0 pertama di tahun 2004. Web 2.0, adalah generasi kedua layanan berbasis web, seperti situs jaringan sosial, wiki, perangkat komunikasi, dan folksonomi. Web 2.0, menekankan pada kolaborasi online dan berbagi antar pengguna. Gambar 1.7 menunjukkan komponen dari pembentuk Web 2.0 sebagai ruang lingkup Web 2.0



Gambar 1.7 Ruang Lingkup Web 2.0

1.7.1 Perbedaan Web 1.0 dan Web 2.0

Pada Web 1.0 memiliki keterbatasan dimana seorang user harus datang ke dalam website tersebut dan melihat satu persatu konten di dalamnya. Pada Web 2.0 memungkinkan pengguna internet dapat melihat konten suatu website tanpa harus berkunjung ke alamat situs yang bersangkutan. Kemampuan Web 2.0 dalam melakukan aktivitas drag and drop, auto complete, chat, dan voice seperti layaknya aplikasi desktop, bahkan berlaku seperti sistem operasi, dengan menggunakan dukungan AJAX atau berbagai plug-in (API) yang ada di internet. Perbedaan Web 1.0 dengan Web 2.0 dapat dilihat pada gambar 1.8.

1.7.2 Prinsip-Prinsip Web 2.0

Prinsip-prinsip yang diterapkan pada Web 2.0 antara lain:

- Web sebagai platform
- Partisipasi dari pengguna dalam berkolaborasi pengetahuan
- Data sebagai pengendali utama
- Akhir dari siklus
- Model bisnis yang ringan, yang dikembangkan dengan gabungan isi dan layanan
- Software tidak lagi terbatas pada perangkat tertentu
- Adanya kemajuan inovasi pada antar-muka (interface) di sisi pengguna

Web 1.0	-->	Web 2.0
DoubleClick		Google AdSense
Ofoto		Flickr
Akamai		BitTorrent
mp3.com		Napster
Britannica Online		Wikipedia
personal websites		blogging
evite		upcoming.org and EVDB
domain name speculation		search engine optimization
page views		cost per click
screen scraping		web services
publishing		participation
content management systems		wikis
directories (taxonomy)		tagging ("folksonomy")

Gambar 1.8 Situs Web 1.0 antara Web 2.0

1.7.3 Layanan pada Web 2.0

Layanan layanan yang disediakan oleh Web 2.0 beserta perangkat pendukungnya, disarikan pada gambar 1.9.



Gambar 1.9 Content Network

1.7.4 Search Engine

Fungsi pencarian merupakan salah satu layanan pada web 2.0, sebagai contoh situs yang memanfaatkan layanan Search :Google Search, Yahoo!, MSN, Ask, Vertical Search, Location-Based Search, Attention Economy, Creating Customized Search Engines, Search Engine Optimization (SEO), Link Building, Search Engine Marketing (SEM).

1.7.5 Content Network

Sebuah Content Network adalah situs Web atau kumpulan situs yang menyediakan informasi dalam berbagai bentuk (seperti artikel, wiki, blog, dll). Ini menyediakan cara lain untuk menyaring sejumlah besar informasi di Internet, dengan memungkinkan pengguna untuk pergi ke terpercaya situs yang telah disortir melalui banyak sumber untuk mencari yang terbaik telah menyediakan konten atau konten sendiri.

1.7.6 User Generated Content

User Generated Content (UGC), juga dikenal sebagai konsumen generated media (CGM) atau user created content (UCC), mengacu pada berbagai jenis isi media, tersedia untuk umum, yang dihasilkan oleh pengguna akhir. Penggunaannya untuk berbagai aplikasi termasuk pengolahan masalah, berita, gosip dan penelitian mencerminkan media perluasan produksi melalui teknologi baru yang dapat diakses dan terjangkau kepada masyarakat umum.

1.7.7 Social Networking

Pada Web 2.0, jaringan sosial menyediakan forum yang unik inovatif baik untuk menggambarkan kolaboratif dan aplikasi yang terbuka, dan berbagi data skenario, serta teknologi dan metodologi baru untuk membangun dan mengelola aplikasi tersebut. Aplikasi tersebut banyak dipergunakan antara lain Friendster, MySpace, Facebook, LinkedIn, Xing, Second Life, Gaia Online, Mobile Social Networking

1.7.8 Social Media

Sebuah Social media adalah media yang dirancang untuk disebarkan melalui interaksi sosial, yang dibuat menggunakan scalable sangat mudah diakses dan teknik penerbitan. Social media menggunakan internet dan teknologi berbasis web untuk mengubah media penyiaran (satu ke banyak) ke dalam dialog media sosial (banyak ke banyak).

Media sosial dapat dikatakan memiliki tiga komponen ;

- Konsep (seni, informasi, atau media).
- Media (fisik, elektronik, atau verbal).
- Sosial antarmuka (intimate direct, keterlibatan masyarakat, social viral, elektronik menyiarkan atau sindikasi, atau media fisik lainnya seperti cetak).

Contoh-contoh situ yang berbasis layanan media social adalah YouTube, Internet TV, Digg, Last.fm, Digital Rights Management(DRM), Podcasting

1.7.9 Tagging - History

Sebuah Tagging – History adalah bagian dari sifat kolaboratif Web 2.0 yaitu user generated setiap kata atau frasa yang membantu mengatur web konten dan label dalam cara yang lebih manusiawi. Meskipun set label standar memungkinkan pengguna untuk menandai konten secara umum, penandaan item dengan label yang dipilih sendiri menciptakan identifikasi yang lebih kuat dari konten.

1.7.10 Tagging - Clouds

Sebuah Tag Clouds menampilkan visual dari tag dilihat oleh popularitas. Banyak situs Web 2.0 termasuk representasi grafis populer tags (popularitas tag ditandai dengan ukuran dari teks, seperti yang ditunjukkan pada gambar 1.10)



Gambar 1.10 Tagging clouds

1.7.11 Tagging - Folksonomies

Sebuah Folksonomies adalah klasifikasi berdasarkan tag. Folksonomies terbentuk pada situs-situs seperti Flickr, Technorati dan del.icio.us. del.icio.us. Pengguna dapat mencari konten oleh tag, yang mengidentifikasi konten yang berbeda-beda (dan kadang-kadang lebih bermakna) dari pada menggunakan kata kunci tradisional yang digunakan oleh mesin pencari

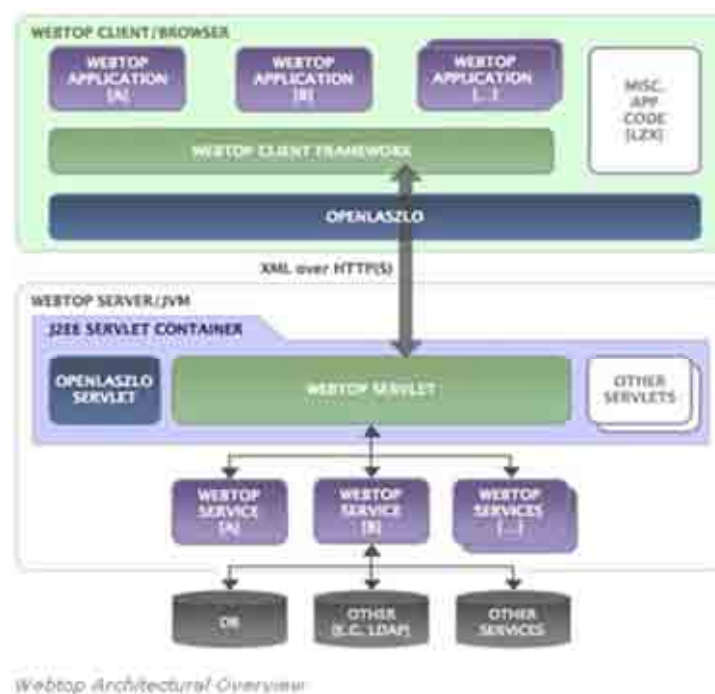
- **Tagging - Flickr**, adalah kunci Penandaan konten situs., Dimaksudkan sebagai suatu cara untuk mengorganisasikan koleksi foto pribadi, penandaan di situs populer karena masyarakat menjadi tertarik pada "pandangan global dari tagscape" (bagaimana orang lain yang tag foto). Pengguna dapat mencari foto menurut tag bermakna. Tag juga mendorong loyalitas ke situs, karena tag foto yang hilang jika pindah ke situs lain.
- **Tagging - Technorati**, sebuah mesin pencari media sosial, menggunakan tag untuk mencari blog yang relevan dan bentuk-bentuk media social
- **Social Bookmarking**, adalah metode untuk internet pengguna untuk berbagi, mengatur, pencarian, dan mengelola bookmark sumber daya web. Tidak seperti file sharing, sumber daya sendiri tidak dibagi, hanya penanda bahwa referensi

1.7.12 Software Development

Kunci untuk pengembangan software Web 2.0 adalah KIS (keep it simple; keep it small). To-the-Point dan orang bisa mengetahui isi situs anda dalam 10 detik.

1.7.13 WebTop

WebTop Memungkinkan untuk menjalankan aplikasi web seperti dalam sebuah lingkungan desktop di web browser. Menggunakan web sebagai platform adalah bagian dari kemajuan menuju operating-system-independent applications. Penghapusan hambatan OS memungkinkan potensi penonton untuk setiap satu produk untuk menjadi lebih besar. Gambaran arsitektur webtop dilukiskan pada gambar 1.11



Gambar 1.11 Arsitektur WebTop

Webtop SDK adalah cara tercepat bagi pengembang untuk menciptakan solusi Webtop dan memanfaatkan pengembangan platform OpenLaszlo. Webtop berisi framework client-server untuk mendukung aplikasi ajax sisi server, diimplementasikan sebagai Java servlet sisi

klien, yang termasuk libraries OpenLaszlo The OpenLaszlo servlet biasanya digunakan untuk mengkompilasi selama pengembangan aplikasi .Contoh aplikasi Webtop yang telah ada, dapat dilihat pada gambar 1.12



Gambar 1.12 Arsitektur WebTop

Bab 2

Dasar-dasar HTML

HTML adalah bahasa pemrograman dasar yang digunakan untuk membangun sebuah situs. Sebuah HTML (*Hypert Text Markup Language*) merupakan bahasa pemrograman yang digunakan dalam pembuatan halaman web. *HTML sebagai HyperText Markup Language* bukanlah bahasa procedural seperti C, Fortran, Cobol atau Pascal. *Markup language* menerjemahkan elemen dari suatu halaman sehingga browser dapat me-render halaman tersebut dan menampilkannya dalam layer. Sebuah HTML (*Hyper Text Markup Language*) merupakan bahasa standar untuk membuat suatu dokumen HTML (halaman web) yang terdiri dari kode-kode singkat tertentu, dimana dengan kode-kode tersebut akan memerintahkan Web browser bagaimana untuk menampilkan halaman Web yang terdiri dari berbagai macam format file seperti teks, grafik, animasi link maupun audio-video.

Web browser adalah sebuah program yang dapat menterjemahkan kode perintah dari dokumen HTML tersebut sehingga dapat kita lihat, baca dan dengar. Contoh dari Web Browser adalah Internet Explorer, Netscape Navigator, Mozilla Firefox, Opera, Safari dll. Kode HTML mempunyai file perluasan (extension) htm atau html. Dimana kedua perluasan tersebut adalah sama, jadi anda boleh menyimpan file dokumen HTML dengan *extension* ".htm" atau ".html". Perintah didalam dokumen HTML terdiri dari berbagai struktur penyusun seperti :

- tag, elemen
- atribut
- Nilai (*value*).

Tag terdiri dari lambang-lambang khusus seperti: "<", ">" dan "/", untuk menuliskannya di dalam dokumen HTML dimulai dengan tag pembuka "<...>" dan diakhiri dengan tag penutup "</...>". Kemudian di dalam tag tersebut terdapat teks yaitu html, head, title, body dan yang lainnya ini disebut dengan Elemen HTML. Elemen HTML kemudian ada yang memiliki atribut dan nilai tertentu, seperti contoh untuk membuat warna latar belakang (background): `<body bgcolor="0000ff">`, Body merupakan elemen, bgcolor adalah atribut dan 0000ff merupakan nilai (value).

Elemen HTML mempunyai tiga hal penting yaitu: tag pembuka, isi, dan tag penutup. Sebagai contoh: Elemen html mempunyai tag pembuka "<html>" dan tag penutup "</html>" dan yang berada diantaranya merupakan isi dari elemen html. Untuk menuliskan Elemen HTML bisa menggunakan huruf besar maupun huruf kecil. Contoh: `<HTML>`, `<HtMl>`, `<HTml>`, `<html>`, semuanya adalah sama.

2.1 Standar Dokumen

Komponen dasar dari suatu dokumen HTML terdiri dari tag elemen HTML, HEAD dan BODY. Semua HTML harus dimulai dengan label <HTML>. Berikut ini contoh sederhana sebuah dokumen HTML standar

```
<HTML>
<HEAD>
  <TITLE>
</TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>
```

Tag dapat memiliki atribut-atribut. Atribut-atribut tersebut menyediakan informasi tambahan dari suatu elemen HTML. Atribut-atribut memiliki format penulisan seperti berikut : name="value", dimana name merupakan suatu atribut, dan value merupakan nilai dari atribut tersebut. Atribut-atribut selalu dispesifikasikan atau ditempatkan pada tag pembuka dari suatu elemen HTML. Atribut-atribut dan nilai atribut juga case-sensitive dalam penulisannya, dianjurkan penulisan tag adalah huruf kecil (*lowercase*) untuk penulisan atribut dan nilai atribut pada HTML 4 dan XHTML

2.1.1 Penulisan Nilai Atribut

Nilai atribut selalu ditempatkan didalam tanda kutip. Biasanya tanda kutip ganda yang digunakan, namun tanda kutip tunggal juga diperkenankan. Dalam situasi tertentu, seperti penggunaan tanda kutip didalam atribut, maka diperlukan tanda kutip tunggal. Kode tag <HTML> adalah awal dari sebuah dokumen HTML standard, diakhir dari sebuah dokumen HTML harus pula dituliskan tag penutup yakni </HTML>. Seperti halnya satu artikel majalah, dokumen HTML memiliki dua bagian berbeda yaitu kepala dan tubuh. Diantara kode tag <HTML> dapat disisipkan kode tag <HEAD>, yang berarti kepala dari dokumen tersebut, kemudian dapat pula diberikan kode tag <TITLE> yang berarti judul halaman. Baik <HEAD> maupun <TITLE> harus ditutup dengan kode tag yang bersangkutan. Bagian lain dari dokumen HTML adalah badan dokumen yang ditandai dengan kode tag <BODY>. Bagian ini mengandung semua bagian yang tampak di halaman. Ini meliputi teks, . gifs, hubungan terkait, tabel, bentuk, dan sebagainya.

Tag pertama pada dokumen HTML diatas adalah " <html>" tag ini merupakan awal dari suatu dokumen HTML, dan tag " </html>" merupakan akhir dari dokumen HTML. Isi keseluruhan dari suatu dokumen HTML selalu berada diantara tag <html>...</html>. Kemudian tag " <head>...</head>" (header) adalah informasi dari dokumen HTML. Informasi di dalam header meliputi: title, meta, style, script dll. Selain judul informasi di dalam header ini nantinya tidak akan di tampilkan di web browser. Tag berikutnya adalah " <title>...</title>" merupakan bagian dari Head, tag ini adalah sebagai judul dari dokumen HTML, judul ini akan tampil pada tab browser. Tag "<body>...</body>" merupakan isi dari suatu dokumen HTML yang akan ditampilkan di web browser, yang terdiri dari berbagai macam format file berupa teks, grafik, link, animasi maupun audio-video (multimedia).

2.1.2 Tag HTML Dasar

Tag HTML dasar adalah sebuah kode HTML sederhana yang dikenali oleh browser standar, terdiri antara lain:

1. Headings

Suatu tag Headings didefinisikan dengan tag <h1> hingga tag <h6>. Tag <h1> mendefinisikan heading yang terbesar, dan <h6> akan mendefinisikan heading yang terkecil. Biasanya tag heading dipakai pada suatu web untuk mendefinisikan suatu judul atau sub judul, dan biasanya mesin pencari seperti google atau yahoo banyak mempergunakan tag heading sebagai patokan untuk mencari kata kunci (*keyword*) pada halaman web yang bersangkutan. Kode tag heading secara otomatis akan menambahkan extra baris kosong sebelum dan sesudah heading ketika diterjemahkan oleh browser.

2. Paragraph

Paragraph didefinisikan dengan tag <p> dipakai untuk mengatur tata letak tulisan. Kode tag paragraf secara otomatis akan menambahkan extra baris kosong sebelum dan sesudah paragraph ketika diterjemahkan oleh browser.

3. Line Breaks

Pergantian baris atau pindah baris tanpa ingin membuat sebuah paragraph, memakai line break, yakni memakai tag
. Kode Tag
 akan menghasilkan perpindahan baris dimanapun dia diletakkan. Tag
 merupakan tag yang tidak memiliki tag penutup atau tanpa tag penutup.

4. Komentar pada HTML

Kode Tag comment digunakan untuk membuat comment pada kode HTML. Comment tersebut tidak akan ditampilkan oleh browser. Penggunaan comment adalah untuk membantu kita dalam dokumentasi source code HTML yang kita buat. Kode tag comment adalah <!-- dan diakhir dengan tag -->

5. Format Text HTML

Kode Tag yang dipakai untuk memformat penulisan terangkum pada tabel 2.1

Tabel 2.1 Daftar Penampilan Teks.

Kode Tag	Penjelasan Format Text
	Tebal
<big>	Besar
	Menekan (Emphasized)
<i>	Miring (Italic)
<small>	Kecil
	Kuat/Tebal
<sub>	subscripted
<sup>	superscripted
<ins>	inserted
	deleted

6. Format Keluaran Komputer

Kode Tag yang dipakai untuk memformat jenis keluaran di komputer terangkum pada tabel 2.2

Tabel 2.2 Daftar Format Keluaran Komputer.

Kode Tag	Penjelasan Format Keluaran
<code>	Mendefinisikan kode teks komputer
<pre>	Mendefinisikan teks terformat

7. Format Kutipan Tag

Kode Tag yang dipakai untuk menuliskan kutipan terdiri dari dua jenis yaitu kutipan panjang memakai kode tag <blockquote> dan kutipan pendek memakai <q>.

8. Karakter Entitas

Beberapa karakter memiliki arti khusus pada HTML, seperti tanda lebih kecil (<) yang mendefinisikan awal dari suatu tag HTML. Jika ingin menampilkan tanda tersebut pada browser, harus menggunakan karakter entitas pada kode HTML. Suatu karakter entitas terdiri dari tiga bagian yaitu :

- tanda & (ampersand)
- nama entitas atau suatu tanda # dan nomor entitas
- tanda ; (semicolon)

Karakter entitas yang sering digunakan diperlihatkan pada tabel 2.3 dibawah ini:

Tabel 2.3 Daftar Karakter Entitas.

Kode Tag	arti
 	Space kosong
<	Tanda lebih kecil '<'
>	Tanda lebih besar '>'
&	Ampersand (dan) '&'
"	Tanda kutipan "
'	'Apostrophe '

2.2 Warna Halaman

Menambahkan berwarna ke halaman web secara relatif sederhana. Masing-masing warna punya satu dari enam kode digit. Kode ini adalah kode Hexadecimal . Dua hal pertama itu angka dari kode adalah sejumlah merah. #FF0000 adalah merah . Digit kedua angka adalah sejumlah hijau. #00FF00 adalah hijau . Dua angka terakhir mewakili sejumlah biru. #0000FF adalah biru . Apapun kombinasi dari kode ini biasanya menciptakan berbagaimacam warna. Untuk mengubah warna baku pada keseluruhan halaman, kode warna dapat disisipkan pada atribut <body>. Tabel 2.4 berikut memperlihatkan beberapa daftar atribut yang dapat dipakai dalam label tubuh:

Tabel 2.4 Daftar Pemakaian Kode Warna.

Kode Tag	arti
bgcolor="..."	warna latar belakang dari halaman.
text="..."	warna dari teks.
link="..."	Warna dari hubungan terkait.
vlink="..."	Warna penanda link
alink="..."	Warna penanda link yang telah aktif

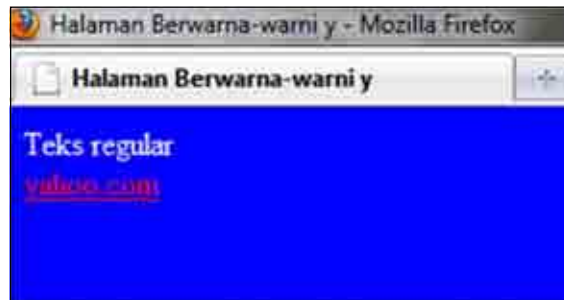
Satu contoh dari satu halaman dengan latar belakang biru, putih teks dan dengan hubungan terkait merah

```
<html> <head>
<title> Halaman Berwarna-warni y</title> </head>
<body bgcolor="#0000FF" text="#FFFFFF" link="#FF0000">
  <p> Teks regular <br>
  <a href="www.yahoo.com"> yahoo.com</a></p>
</body>
```



```
</html>
```

Hasil dari contoh diatas digambarkan pada gambar 2.1:



Gambar 2.1 Hasil pengaturan warna halaman

2.3 Tautan Halaman

Tautan (*link*) halaman atau hubungan terkait merupakan sebuah penghubung antara dua halaman HTML. Halaman HTML dapat di hubungkan dengan label `<a>` dan di akhiri dengan ``. Misalnya terdapat "page1. html" dan "page2. html" (pastikan halaman tersebut berada pada map yang sama). Pada "page1. html" letakkan kode tag `<a>` diantara tag `<body>` dan `</body>`. Kode tag `<a>` dituliskan sebagai halaman tautan : ` umur 2`

Dengan tautan itu `` berarti membuat satu penghubung untuk page2.html ketika tulisan yang berada di antara `<a>` dan `` diklik, maka halaman akan berganti ke halaman tautan. Terdapat dua macam tautan yaitu, relatif dan absolut. Kebanyakan dari hubungan terkait adalah bertipe relatif. Tautan Relatif pathnames menunjuk arsipkan berlandaskan lokasi mereka sehubungan dengan file saat ini, sementara absolut pathnames menunjuk arsipkan berlandaskan lokasi absolut mereka pada sistem berkas. Pembuatan absolut penghubung dapat dilakukan dengan mengubah menjadi lokasi file ` umur 2`. Untuk menghubungkan ke halaman lain pada web, yang tidak berlokasi sama dengan halaman web saat ini, kode tag tautan dituliskan pada alamat yang dituju di dalam href="...". Contoh halaman lain yang ditautkan dengan kode tag sebagai berikut ini : ` yahoo! .`

Selain halaman yang dapat ditautkan, tautan dapat dipakai untuk membuat satu penghubung pada alamat email. Contohnya adalah: `here`

Kode tag tautan `<a>` dapat pula di pakai sebagai jangkar (anchor) pada halaman yang sama. Contoh berikut akan punya penghubung dibawah halaman: `To top`, ketika di klik, tautan tersebut menuju pada sebuah label yang telah didefinisikan yaitu "top". Pada sebuah halaman, tautan jangkar (anchor) dapat dibuat sebanyak mungkin, bergantung kebutuhan, biasanya dipakai untuk mempercepat navigasi halaman.

```
<html><head>
<title>Anchors</title>
</head><body>
<a name="top">
<p>Paragraph 1</p> <p>Paragraph 2</p> <p>Paragraph 3</p>
<p>Paragraph 4</p> <p>Paragraph 5</p><p>Paragraph 6</p>
<p>Paragraph 7</p>
<a href="#top">To top</a></body>
</html>
```

2.4 Image

Gambar (*image*) adalah gambar yang dipasang pada halaman web. Meletakkan satu image ke atas satu halaman secara relatif sangat mudah. Kode tag image adalah:

```

```

Sebuah gambar yang berupa file `person.jpg` di muat kedalam halaman dengan ukuran lebar 50 dan tinggi 50, batas antara gambar dengan sisi sekelilingnya adalah 0 yang berarti tanpa diberi batas. Penempatan gambar adalah sebelah kiri dan ketika gambar gagal dimuat, maka tulisan "person" dimunculkan sebagai alternative. Untuk membuat gambar diatas, tertaut ke dalam satu penghubung, opsi tautan dapat ditambahkan pada gambar di dalam kode penghubung:

```
<a href="somefile.html" src= img "orang. jpg" width= "50" height= "50" border= "1 " alt= "Orang"></a>
```

Ketika gambar akan dipakai didalam halaman web, pertimbangan utama adalah ukuran dari grafis. Pemilihan grafis sekecil mungkin, dimaksudkan untuk mempercepat pemuatan gambar, sehingga halaman lebih cepat tertayangkan. Sebuah halaman web, direkomendasikan untuk pemuatan gambar pada satu halaman tidak lebih dari 30KB . Ada dua jenis utama dari image pada web: JPG dan GIF. Jenis ini adalah format kompresi yang sangat tinggi. Sebuah file format BMP, sebaiknya dikonversi menjadi jpg atau gif. JPG memampatkan foto atau gambar besar terbaik, sementara GIF adalah berguna bagi gambar kecil dengan beberapa warna, transparansi atau animasi. Pada gambar dengan format GIF, pastikan angka paling sedikit dari warna pada gambar. Hal ini dapat dilakukan dengan menyusut ukuran dari pelet atau penurunan angka dari warna dari 256 ke 16.

2.5 Penampilan teks

Tag HTML untuk font adalah `...`, dengan tag ini kita bisa menentukan jenis font (face), warna (color), dan ukuran (size). Sesuai dengan rekomendasi dari World Web Consortium (W3C), Untuk HTML 4.0 keatas, tag `` tidak dipergunakan lagi, sebagai gantinya dibuat tag `<style>` atau disebut dengan Cascading Style Sheets (CSS). Dengan CSS penggunaannya akan lebih kompleks dan lebih luas lagi karena dapat dikombinasikan dengan tag elemen-elemen lainnya di dalam dokumen HTML. Ada banyak kode tag yang mempunyai efek pada perubahan penampilan dari teks. Penampilan secara umum dari text diatur dengan tag ` ... `. Contoh dibawah ini menunjukkan penampilan teks dengan pengaturan bentuk lainnya: ` ...`

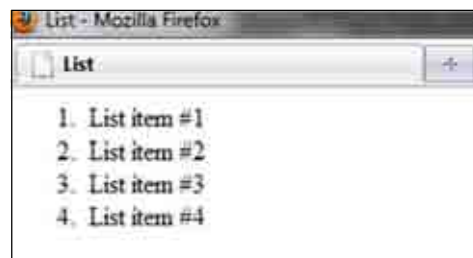
Kode tag `` dapat mengubah ukuran, warna, dan tampilan (huruf). Ukuran dapat sejumlah dari 1 relatif 7 atau satu angka seperti + 1 (satu lebih besar ukuran dibandingkan nilai standar). Kode tag color dipakai untuk memberi warna pada teks, dimana warna dapat dipilih pada sekumpulan color set. •Kode tag face adalah tampilan dari jenis huruf yang dipakai. Berbagai macam jenis huruf tersedia secara standar yang disediakan oleh komputer antara lain Arial, Times New Roman, Helvetica, Tahoma, Courier. Kombinasi berbagai macam huruf dapat ditambahkan seperti Contoh: `<font face="verdana, arial black, arial".` Dengan huruf yang berbeda beda, maka komputer akan mencoba hingga menemukan jenis huruf yang sesuai. Perintah tag base font dapat dipakai untuk Menetapkan nilai default pada sebuah halaman, contohnya `<BASEFONT size="2" color="#FFFF00" face="arial">`

2.6 Lists

Sebuah halaman html menyediakan satu cara yang mudah untuk menciptakan daftar, diorder (diberi nomor) atau tidak diorder (tidak diberi nomor). Daftar diorder adalah ` ...`. Di dalam label ini, setiap elemen harus mendefinisikan individu yang termuat pada tag ` ...`. Contoh:

```
<html><head>
<title>List</title>
</head><body>
<ol>
<li>List item #1</li><li>List item #2</li>
<li>List item #3</li><li>List item #4</li>
</ol>
</body>
</html>
```

Hasil dari list yang diperlihatkan pada gambar 2.2



Gambar 2.2 Hasil list dengan Penomoran

Selain dengan penomoran, pembuatan Lists tanpa order yang berarti menghilangkan penomoran masing-masing elemen. Kode tag List tersebut adalah ` .. `. Daftar tidak diorder pasang dengan ` ...` adalah sangat serupa, mengecuali akhir hasil akan punya "bullet" sebagai ganti angka. Berikut ini kode program dari list tanpa penomoran:

```
<html><head>
<title>List</title>
</head><body>
<ul>
<li>List item #1</li><li>List item #2</li>
<li>List item #3</li><li>List item #4</li>
</ul>
</body></html>
```

Hasil yang tampak di browser diperlihatkan pada gambar 2.3:



Gambar 2.3 Hasil list tanpa Penomoran

2.7 Horizontal Rules

Sebuah halaman html juga mempunyai satu cara sederhana membuat sebuah ketentuan horisontal (atau garis) untuk membagi satu halaman atas secara terpisah. Ketentuan horisontal `<hr>` punya beberapa atribut yang dapat didefinisikan pada tabel 2.5

Tabel 2.5 Daftar Atribut Horizontal Rule.

<code>align=".."</code>	Mengubah letak dapat berupa left, right atau center.
<code>size=".."</code>	Ukuran atau ketinggian dari ketentuan.
<code>width=".."</code>	Lebar dari garis. Ini dapat ditulis sebagai angka dari titik lebar(<code><hr width="400"></code> atau sebagai satu persen dari lebar dari layar(<code><hr width="75%"></code> .
<code>color=".."</code>	Setting pilihan warna .

Contoh pemakaian adalah `<hr align="center" size="5" width="80%">` . Hasil yang tampak di browser adalah berupa horizontal rule seperti dibawah ini:

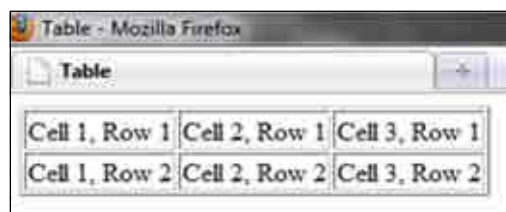


2.8 Tabel

Tabel adalah alat yang paling efektif untuk mengatur tata letak isi sebuah halaman. Kode tag untuk membuat tabel adalah `<table> ...</table>`. Tiap baris dari isi tabel diberikan kode tag `<tr> ...</tr>`. Tiap kolom dari isi tabel diberikan kode tag `<td> ...</td>`

```
<html><head>
<title>Tabel</title>
</head><body><table border="1">
<tr> <td>Cell 1, Row 1</td><td>Cell 2, Row 1</td><td>Cell 3, Row 1</td></tr>
<tr> <td>Cell 1, Row 2</td><td>Cell 2, Row 2</td><td>Cell 3, Row 2</td></tr>
</table>
</body>
</html>
```

Hasil dari kode HTML diatas, diperlihatkan pada gambar 2.4:



Gambar 2.4 Hasil Tabel

Setting `border=0` berarti batas antar kolom dan baris tidak tampak sebagai garis pemisah. Berikut ini daftar beberapa setting dari tabel, diperlihatkan pada tabel 2.6

Tabel 2.6 Daftar Atribut Tabel

<code><TABEL></code>	
<code>width="..."</code>	Lebar dari tabel, dapat berupa nilai persentase atau nilai pixel (contoh 90%) atau pixel (contoh 500)
<code>border="..."</code>	Lebar di titik dari satu perbatasan sekeliling tabel
<code>cellspacing="..."</code>	Mengatur jarak di antara sel individu
<code>cellpadding="..."</code>	Mengatur jarak di dalam sel di antara data dan tepi dari sel
<code>align="..."</code>	Pilihan perataan (left, right, center, justify)
<code>bgcolor="..."</code>	Menetapkan warna latar belakang tabel
<code>bordercolor="..."</code>	Menetapkan warna border
<code>bordercolorlight="..."</code>	Menetapkan warna highlight
<code>bordercolordark="..."</code>	Menetapkan warna shadow
<code><TR> = Baris tabel</code>	
<code>align="..."</code>	Pilihan perataan secara horisontal pada isi dari sel di dalam baris (left, center, right, justify)
<code>valign="..."</code>	Pilihan perataan secara tegak lurus pada isi dari sel di dalam baris (top, middle, bottom).
<code>bgcolor="..."</code>	Tetapkan warna latar belakang dari baris (sel)
<code><TD> = Data tabel</code>	
<code>rowspan="..."</code>	Jumlah baris yang di span dalam satu sel.
<code>colspan="..."</code>	Jumlah kolom yang di span dalam satu sel.
<code>align="..."</code>	Pengaturan perataan dari (left, center, right, justify)
<code>valign="..."</code>	Pilihan perataan secara tegak lurus pada isi dari sel di dalam kolom (top, middle, bottom).

Kode Rowspan dan colspan sangat kompleks sebagai alat yang berguna untuk mengatur isi dari sebuah tabel. Contoh untuk menampilkan seperti pada gambar 2.5. Kode HTML untuk membuat tata letak seperti diatas, dengan memakai rowspan dan colspan



Gambar 2.5 Tabel dengan Pengaturan

Kode untuk tabel pada gambar 2.5 adalah sebagai berikut:

```
<html><head>
<title>Tabel</title>
</head><body>
<tabel border="1">
<tr> <td colspan="3">Title</td></tr>
<tr> <td rowspan="2">Left</td><td>A</td> <td>B</td> </tr>
<tr> <td>C</td> <td>D</td> </tr>
</tabel>
</body>
</html>
```

Tabel dapat dibuat lebih kompleks dengan menambahkan di warna latar belakang, perataan, dan lebar tabel. Penambahan kode adalah memberi ruang satu spasi. Tampilan tabel yang lebih kompleks diperlihatkan pada gambar 2.6



Gambar 2.6 Tabel dengan Pengaturan lebih kompleks

Kode HTML dari gambar 2.6:

```
<html><head><title>Tabel</title>
</head><body>
<tabel width="75%" border="4" cellspacing="0">
<tr><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td></tr>
<tr><td>&nbsp;</td>
<td>
  <tabel align="center" width="90%" border="1" bordercolor="#FF00FF">
  <tr><td>&nbsp;</td><td>&nbsp;</td> </tr>
  <tr><td>&nbsp;</td><td>&nbsp;</td> </tr>
  </tabel>
</td>
<td>&nbsp;</td></tr>
<tr><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td></tr>
</tabel>
</html>
```

Penulisan diatas, tabel yang memiliki lebar 90% tapi tidak di isi sepenuhnya. Isi tersebut dapat berupa teks, gambar, atau sebuah tautan.

2.9 Frame

Tag frame berfungsi untuk membagi sebuah halaman web ke dalam beberapa bagian sehingga penampilan lebih dari 1 dokumen HTML pada browser yang sama. Setiap dokumen HTML disebut dengan frame, dan setiap frame terpisah dengan yang lainnya. Kode tag frame adalah <frameset> digunakan untuk membagi window menjadi frame-frame. Kode <frameset> mendefenisikan lebar dari kolom atau baris, didalam kode <frameset> terdapat Tag Frame yang mendefinisikan file HTML mana yang akan ditampilkan pada frame. Contoh :

```
<frameset rows="15%,85%">
<frame src="file1.html">
<frame src="file2.html">
</frameset>
```

Pada contoh diatas, frameset memiliki 2 kolom. Kolom pertama diset dengan nilai 15% dari lebar browser, kolom yang kedua diset 85% dari lebar kolom. Frame pertama akan menampilkan dokumen file1.html dan frame yang kedua akan menampilkan dokumen file2.html. Selain menggunakan persen, kita juga dapat meng-set nilai kolom dengan pixel (cols="100,500") dan dapat juga meng-set salah satu kolom saja, dan kolom berikutnya menyesuaikan (cols="15%,*")

Jika border pada suatu frame ditampilkan dibrowser, maka user dapat mengganti ukuran kolom hanya dengan click and drag . Untuk mencegah hal tersebut tambahkan `noresize="noresize"` pada tag frame. Tambahkan tag `<noframes>` untuk browser yang tidak mensupport browser. Namun kekurangan dari penggunaan frame bahwas halaman web yang memiliki frame biasanya akan susah di print / cetak

2.10 Form

Salah satu tugas penting dari internet adalah mendapatkan infomasi/input dari pemakai. Inputan ini bisa berupa :

- Text boxes (input text)
- Button (input berupa tombol)
- Submit (input berupa tombol submit/proses)
- Reset (input tombol, mengembalikan nilai ke sebelumnya)
- Checkboxes (input pilihan kotak)
- Options Buttons (inputan pilihan tombol)
- Passwords (inputan text dengan text tersandi)
- Text Area (inputan text dengan lebih banyak baris)
- List Boxes (inputan dari deretan nilai yang ditentukan)

Pembuatan inputan dibedakan menjadi dua bagian yakni :

1. Bagian form

Biasanya berupa file HTML (bisa juga ASP atau lainnya), yang bertugas untuk menampilkan inputan yang bisa diisi oleh pemakai

2. Bagian Aksi

Bagian program berupa aksi yang diharapkan setelah form terisi, biasanya merupakan kode program ASP untuk pemrosesan data

Form merupakan suatu area yang berisi element-element untuk isian dari pengguna, seperti text field, text area, drop down menu, radio button, check box, dan sebagainya. Sebuah form didefinisikan dengan tag `<form>` dan diakhir dengan `</form>`.

Elemen dari form terdiri dari:

1. Atribut method, yang berisi pilihan untuk menentukan cara pengiriman data ke server, terdapat dua pilihan yaitu metode get atau post
2. Atribut Action, yang menentukan aksi ke sebuah URL yang dilakukan ketika pengguna menekan tombol submit
3. Input, area dimana pengguna melakukan isian ataupun pilihan

2.10.1 Membuat Form

Isian inputan dibuat biasanya dibuat dalam kode HTML dengan ditandai di awal blok form `<form>` dan diakhiri blok form `</form>`. Format penulisan form adalah :

```
<Form [name:nama_formnya] [action=file_aksinya] [method=metode_dipakai]
[target=namaframe/file] >
```

```
.....
<Form >
```

Keterangan

Tanda [...] : merupakan opsi, boleh ada boleh tidak
Name : nama form, sangat berguna jika dipakai lebih dari satu form dalam satu file HTML
Action : dipakai untuk mendefinisikan file aksi yang akan dipanggil / eksekusi jika tombol submit ditekan/klik oleh pemakai
Method : metode yang dipakai untuk tukar-menukar data antar file form dengan file aksinya. Kalau method ini tidak dituliskan maka otomatis browser, menganggap metodenya adalah get

a. Metode GET

Metode untuk form ini dipakai untuk mengambil data dan dibawa data tersebut ke dalam alamat URL dengan nama=datanya. Dipakai jika datanya tidak rahasia, sebab data yang dibawa dapat dilihat di alamat lokasi misalnya : www.yahoo.com/search?id=2322&list=go. Tanda ? berarti tanda metode get (mengambil) dan id adalah nama field sedangkan 2322 adalah datanya. Tanda & berarti field selanjutnya, list adalah nama field dan go adalah datanya, keterbatasan metode ini adalah panjang datanya kira-kira 2K atau 2047 karakter

b. Metode POST

Dengan meletakkan data sebagai aliran data yang dikirim dari file form ke file aksinya, dengan tidak terlihat oleh pemakai dan jumlah datanya tidak terbatas. Kelemahannya adalah beberapa *firewall* dapat memotong aliran data ini sehingga data bisa hilang. Sehingga metode ini sering dipakai untuk data yang bersifat rahasia, yang berarti sampaikan data yang ada tanpa diperlihatkan ke lokasinya. Target dipakai untuk menentukan hasil lokasi window dengan nama tertentu saat proses submit dilakukan.

2.10.2 Jenis Input pada Form

Pada form tersedia berbagai pilihan input, yang tergantung dari penggunaannya. Kategori input

- **Input Tombol**
 - tombol aksi (submit)
 - tombol kembalikan nilai (reset)
 - tombol khusus (button)
- **Input Data string**
 - Input text tunggal (tipe text)
 - Input text multiline (tipe text area)
 - Input text sembunyi (tipe text hidden)
 - Input text sandi (tipe text password)
- **Input Data Pilihan**
 - Input pilihan kotak pilih (tipe checkbox)
 - Input pilihan nilai tunggal (tipe radiobutton)
 - Input pilihan berupa daftar (tipe pulldown box)
 - Input pilihan berupa multilist (tipe scroll box)
- **Input Data Khusus**
 - Input file
 - Input gambar (image)

2.10.3 Input tipe Tombol

Input tipe ini tampil di browser berupa tombol dengan format baku antara lain :

```
<input tipe=[submit,button,reset] [name=nama tombol] [value=judul tombol]
[height=x] [width=y]>
```

2.10.4 Tombol Tipe submit

Didalam form hampir selalu dijumpai tombol dengan tipe submit, yang berarti jika tombol tersebut ditekan/klik, maka proses selanjutnya akan mengerjakan file yang telah didefinisikan pada tag form action. Format penulisan submit adalah :

```
<input type=submit [value=text_yg_ditampilkan] [name=nama tombol] [width=lebarnya]
[height=tingginya]>
```

Tombol tipe submit dapat digantikan dengan tombol tipe button, namun fungsi akan sangat berbeda, sebab tombol submit jika ditekan akan mengeksekusi file aksi, sedangkan pada tombol tipe button tidak melakukan apapun, kecuali didefinisikan dalam bahasa javascript untuk mengeksekusi fungsi bahasa javascript. Tombol submit dapat lebih dari satu dan untuk membedakan satu dengan lainnya bisa diberi nama yang berbeda.

2.10.5 Input tombol tipe Reset

Selain tombol dengan tipe submit, hampir selalu juga disediakan tombol reset yang berarti mengembalikan nilai sebelum state sekarang, jika nilai state sebelumnya kosong, maka akan ditampilkan nilai kosong. Tipe tombol reset bukan berarti untuk menghapus inputan, tetapi karena biasanya state sebelumnya adalah kosong, maka biasanya dipakai sebagai penghapusan data yang diisikan. Format penulisan reset adalah :

```
<input type=reset [value=text_yg_ditampilkan] [name=nama tombol] [width=lebarnya]
[height=tingginya]>
```

2.10.6 Input tombol tipe button

Khusus untuk tipe ini dipakai untuk menjalankan kejadian (*event handler*) dari browser dan biasanya berupa fungsi javascript. Sangat berguna jika diperlukan banyak input tombol atau membuat menu pulldown dengan memakai browser. Hal ini akan dibahas lebih detil pada bab JavaScript. Format penulisan reset adalah :

```
<input type=button [name=nama tombol] [value=text_yg_ditampilkan] [width=lebarnya]
[height=tingginya] eventnya>
```

2.10.7 Input tipe Data String

Input tipe ini tampil di browser berupa kotak isian yang menampung data bertipe string/kata/text dengan format baku antara lain :

```
<input tipe=[text,hidden,password,text area] [name=nama fieldnya/data] [value=isi
data awal] [size=ukurannya]>
```

2.10.8 Input tipe Text

Tipe input ini paling banyak dipakai untuk menerima masukan dari pemakai berupa text (string) Format penulisan input tipe text adalah :

```
<input type=text [value=text_yg_ditampilkan] [name=nama_fieldnya]
[size=ukurannya]>
```

2.10.9 Input tipe Text Area

Dipakai untuk inputan kotak yang lebih banyak baris, tergantung lebar yang di kehendaki. Input yang dimasukkan berupa karakter. Format penulisan input tipe text area adalah :

```
<textarea [name=nama text areanya] [cols=panjang kolom] [rows= jumlah barisnya] >
isi dari text areanya</textarea>
```

Tag cols menunjukkan lebar dari text area yang diinginkan dan tag rows menunjukkan jumlah baris yang diinginkan

2.10.10 Input tipe sembunyi (hidden)

Hampir sama dengan input tipe text, perbedaannya adalah kotak input tidak muncul di browser. Biasanya dipakai untuk input antar form sehingga pertukaran data antar satu form/file dengan form/file lain tidak perlu ditampilkan. Format penulisan input tipe hidden adalah :

```
<input tipe=hidden [name=nama fieldnya/data] [value=isi data awal] >
```

2.10.11 Input tipe sandi (password)

Hampir sama dengan input tipe text, perbedaannya adalah kotak input ketika diisi oleh pemakai akan disandikan dengan tanda asterik '*'. Biasanya dipakai untuk memasukkan password dan dicek dalam database apakah passwordnya cocok atau tidak. Format penulisan input tipe hidden adalah :

```
<input tipe=password [name=nama fieldnya/data] [value=isi data awal] >
```

2.10.12 Input-input tipe Pilihan

Input tipe ini tampil di browser berupa kotak pilihan yang bisa diklik atau scroll dengan isian yang menampung data bertipe string/kata/text dengan format baku antara lain :

```
<input tipe=[radio,check] [name=nama fieldnya/data] [value=isi data awal] >
```

2.10.13 Input tipe Radio

Menampilkan sejumlah pilihan bagi pemakai dan hanya bisa dipilih satu saja dari beberapa pilihan yang ditampilkan. Format penulisan input tipe radio adalah :

```
<input type=radio [value=text_yg_ditampilkan] [name=nama_fieldnya] [cheked]>
```

Menampilkan sejumlah pilihan, maka input tipe radio ditulis berulang dengan name yang sama, sedangkan untuk menentukan nilai pilihan radio pertama, dipakai tag checked, pada pilihan radio yang dikehendaki. Tipe radio dapat juga dipakai untuk pilihan berdasarkan group (kelompok) dengan masing-masing pengelompokan adalah namanya disamakan

2.10.14 Input tipe check box

Dipakai untuk menampilkan sejumlah pilihan yang disertai kotak penanda pilihan (check box). Pemakai dapat memilih lebih dari pilihan yang disediakan dengan cara menandai kotak-kotak yang disediakan. Bisa juga dipakai sebagai group (kelompok) pertanyaan dengan pemisah nama dari masing-masing group yang dibedakan, sedangkan dalam satu group namanya sama. Format penulisan input tipe radio adalah :

```
<input type=checkbox [value=text_yg_ditampilkan] [name=nama_fieldnya] [cheked]>
```

2.10.15 Input tipe pulldown box

Dipakai untuk menampilkan sejumlah pilihan berupa kotak dengan window yang bisa dibuka untuk melihat pilihan (*pulldown box*). Pemakai dapat memilih satu pilihan yang disediakan dengan klik pilihan yang disediakan. Bisa juga dipakai sebagai group (kelompok) pertanyaan dengan pemisah nama dari masing-masing group yang dibedakan, sedangkan dalam satu group namanya sama. Tag **Selected**, dapat dipakai untuk menentukan pilihan awal yang muncul. Format penulisan input tipe pulldown box adalah :

```
<select name = namanya>
<option [selected]>Pilihan 1
....
<option [selected]>Pilihan n
</select>
```

Pemakaian pulldown box, biasanya dipakai input dengan pilihan yang lebih dari satu dengan inputan yang tidak dinamis dan hanya satu saja yang bisa dipilih. Contohnya adalah pemilihan nama hari, nama bulan, agama dll.

2.10.16 Input tipe scroll box

Dipakai untuk menampilkan sejumlah pilihan berupa kotak dengan window yang bisa dibuka untuk melihat pilihan (scroll box). Pemakai dapat memilih lebih dari satu pilihan yang disediakan dengan klik pilihan yang disediakan dan menekan tombol CTRL untuk memilih lebih dari satu. Bisa juga dipakai sebagai group (kelompok) pertanyaan dengan pemisah nama dari masing-masing group yang dibedakan, sedangkan dalam satu group namanya sama. Tag Selected, dapat dipakai untuk menentukan pilihan awal yang muncul. Format penulisan input tipe scroll box adalah :

```
<select name = namanya multiple>
<option [selected]>Pilihan 1
....
<option [selected]>Pilihan n
</select>
```

Pemakaian scroll box, biasanya dipakai input yang lebih dari satu dengan inputan yang tidak dinamis. Contohnya adalah pemilihan hobby, pekerjaan, kesukaan dll. Pemakaian scrollbar misalnya adalah pemilihan hobby.

2.10.17 Input tipe Khusus

Input tipe khusus ini jarang dipakai kecuali betul-betul mahir penggunaannya, seperti misalnya loading, upload, animasi gambar dan sebagainya.

1. Input tipe file

Input ini dipakai untuk memilih file yang ada di drive lokal dan biasanya berupa file yang bisa di-upload (untuk download tinggal memanfaatkan fasilitas alamat referensi dari HTML yang akan dibahas pada bab lain). Format penulisan input file adalah :

```
<input type=file [value=text_yg_ditampilkan] [name=nama_fieldnya] >
```

Contoh input tipe file adalah untuk mengetahui pilihan file dari pemakai yang ingin didaftarkan ke server.

2. Input tipe image

Input tipe gambar adalah menghasilkan koordinat x dan y dari gambar yang dipilih, tergantung posisi dari kursor mouse saat menekan gambar tersebut. Contoh adalah deteksi region yang sering dipakai pada model GIS (*geographic information system*). Dibuat suatu peta lokasi misalnya Jawa Timur, kemudian ditentukan posisi koordinat tiap kota, maka bisa di proses user memilih kota mana dan dari pemilihan tersebut bisa diinformasikan misalnya angka-angka statistik. Format penulisan input image adalah :

```
<input type=image src= namafile_petanya [name=nama_fieldnya] >
```

Bab 3

XHTML

HTML (Hypertext Markup Language) adalah bahasa dari World Wide Web yang dipergunakan untuk menyusun dan membentuk dokumen agar dapat ditampilkan pada program browser. XML (Extensible Markup Language) adalah bahasa yang digunakan untuk menyimpan dan mentransfer data yang dirancang khusus untuk penyampaian informasi melalui World Wide Web, persis seperti HTML. Sebuah XML didesain untuk mengolah dan mengatur data. Namun bukan untuk menggantikan HTML, karena keduanya mempunyai tujuan yang berbeda. XML untuk menggambarkan data dan memfokuskan data apa itu, sedangkan HTML untuk menampilkan data dan bagaimana tampilannya. Dengan kata lain, HTML untuk menampilkan informasi, sedang XML untuk menggambarkan sebuah informasi.

XHTML adalah sebuah kombinasi dari HTML dan XML. XHTML ato EXtensible HyperText Markup Language merupakan bentuk sempurna (dalam arti, lebih ketat dan bersih dibanding HTML). Karena XHTML merupakan versi HTML yang menggunakan standar XML jadi penulisan HTML-nya tidak bisa sembarangan dan harus menggunakan aturan XML.

3.1 Versi XHTML

Standard XHTML dibuat oleh W3C yang independen terhadap vendor-vendor browser. Selain itu ada juga organisasi yang menyebut dirinya dengan webstandards.org yang memperjuangkan masalah standard dokumen HTML dan sejenisnya. Tujuan dibentuknya XHTML adalah untuk menggantikan kedudukan HTML, kehadiran XHTML bukan hanya semata-mata menggantikan kedudukan HTML, tetapi diharapkan adanya bahasa standar dalam mendesain web. XHTML akan menetapkan aturan-aturan baru dalam pemformatan dokumen yang dulu masih dimaklumi oleh HTML. Jadi, dokumen XHTML tidak bisa ditampilkan dengan baik apabila aturan-aturan dalam XHTML dilanggar.

Beberapa versi XHTML antara lain

1. XHTML 1.0

Organisasi The World Wide Web Consortium (W3C) membuat xhtml berdasarkan html 4, bertujuan agar keuntungan xml dapat digunakan di dunia internet.

2. XHTML 1.1 – Module-based XHTML

Pengembangan xhtml 1.0 dengan penambahan ruby element (ruby, rbc, rtc, rb, rt and rp). Penghapusan atribut lang (xml:lang)

3. XHTML 1.2

Terdapat pengembangan support web Application dan peningkatan Symantic web

4. XHTML 2.0

Mengurangi dependency terhadap versi xhtml 1.0 maupun HTML 4. Contoh:

- html form diganti xform
- html frame diganti xframe
- DOM event diganti XML event

5. HTML5 – Vocabulary and API for HTML5 and XHTML5

Sebuah Web Hypertext Application Technology Working Group mengembangkan secara independent html 5. Muncul dalam versi regular text/html serialization dan XML serialization. Compatible dengan versi-versi sebelumnya

3.2 Kategori XHTML

Beberapa kategori atau macam-macam XHTML dilihat dari deklarasi XHTML-nya, antara lain

- Strict/strict doctype declaration**
Penggunaan doctype ini, kita mengontrol seluruh layout dan format sebuah halaman web melalui CSS, tidak menggunakan tag *font* dan *tabel* (deprecated). Presentasi harus benar-benar dipisahkan.
- Transitional/transitional doctype declaration**
Kita dapat menggunakan sebagian besar tag-tag HTML seperti background color, font, dan tabel untuk layouting halaman web.
- Frames/frameset declaration**
Mendukung penggunaan frame pada halaman web.

3.3 Aturan XHTML

Penulisan XHTML harus mengikuti aturan yang telah ditetapkan. Aturan Tag HTML terdapat 4 aturan dasar yang harus diperhatikan dengan XHTML:

1. Tag XHTML harus disarangkan dengan benar.
2. Tag XHTML harus selalu ditutup dengan benar.
3. Tag XHTML harus selalu ditulis dalam huruf kecil.
4. Dokumen XHTML harus selalu memiliki tag dasar berjumlah satu saja.

1. Tag XHTML Harus Disarangkan Dengan Benar

Sering kali desainer atau programmer web mendesain tag yang tidak disarangkan dengan benar. Di XHTML, ini harus dihindari, misalnya: `<div><p>Teks tertentu</div></p>`. Pada XHTML, penulisan tersebut salah karena ketika tag `<p>` belum disarangkan sudah ada penutup tag `<div>` sehingga tidak benar. Penulisan yang benar harusnya adalah: `<div><p>Teks tertentu</p></div>`

2. Tag HTML Harus Ditutup Dengan Benar

Elemen-elemen yang ada isinya, harus ditutup, berikut ini contoh yang salah, yaitu dalam penulisan list di unordered list maupun ordered list yang tidak diberi tag penutup.

```
<ul>
  <li>Item1</li>
  <li>Item2
  <li>Item3</li>
</ul>
```

Penulisan yang benar dari kode sebelumnya adalah:

```
<ul>
  <li>Item1</li>
  <li>Item2</li>
  <li>Item3</li>
</ul>
```

Sementara tag yang kosong harus ditutup dengan tanda `</>` misalnya `
` dan `<hr>`. Jadi penulisan berikut adalah salah: `
 `. Penulisan yang benar dari tulisan diatas adalah `
 `

3. Tag XHTML Harus Ditulis Huruf Kecil

Pada XHTML juga mensyaratkan semua teks yang digunakan di tag ditulis menggunakan huruf kecil. Disamping karena perintah XHTML, sebenarnya menggunakan huruf kecil lebih mudah dan juga lebih enak dibaca dibandingkan dengan huruf besar. Berikut ini contoh yang salah :

```
<BODY>
    <P>teks tertentu</P>
</BODY>
```

Penulisan yang benar dari tulisan diatas adalah

```
<body>
    <p> teks tertentu </p>
</body>
```

4. Dokumen XHTML Harus Memiliki Elemen Root Satu

Tag XHTML semuanya harus diletakkan dalam satu tag root yaitu tag `<HTML>`. Semua tag yang lain haruslah sub- elemen dari tag `<HTML>` tersebut. Contohnya seperti berikut:

```
<html>
    <head> ... </head>
    <body> ... </body>
</html>
```

5. Aturan-aturan Tambahan XHTML

Selain aturan pada subab 3.3, terdapat beberapa aturan tambahan XHTML, yaitu:

- Nama atribut juga harus huruf kecil.
- Nilai atribut harus ditutup oleh tanda petik dua.
- Tidak boleh melakukan minimasi atribut.
- Atribut id menggantikan atribut name.

1. Nama Atribut Juga Harus Huruf Kecil

Nama atribut dari sebuah tag/ elemen juga harus ditulis oleh huruf kecil seperti halnya nama tag tersebut. Berikut ini contoh yang salah: `` Penulisan yang benar dari tulisan diatas, seharusnya semua nama atribut kecil, seperti berikut: ``

2. Nilai Atribut Harus Ditutup Oleh Tanda Petik Dua

Aturan XHTML mensyaratkan semua nilai atribut, baik berupa angka, teks atau kombinasi untuk ditutup oleh tanda petik dua.

Berikut ini contoh yang salah: ` `

Penulisan yang benar dari tulisan diatas adalah

```
<a href="halaman.html">

```

3. Tidak Boleh Melakukan Minimasi Atribut

Atribut harus diset memiliki nilai dan tidak boleh menyebutkan nilainya saja, contoh yang salah misalnya:

```
<input checked>
<input readonly>
<input disabled>
<option selected>
<frame noresize>
```

Penulisan yang benar dari tulisan diatas adalah

```
<input checked="checked" />
<input readonly="readonly" />
<input disabled="disabled" />
<option selected="selected" />
<frame noresize="noresize" />
```

4. Atribut id Mengantikan Atribut name

Adanya atribut id digunakan untuk menggantikan atribut *name* yang biasa digunakan di kode HTML konvensional. Jadi berikut ini merupakan contoh yang salah menurut XHTML walaupun bisa juga dimengerti oleh browser:

```
<p class="putih" name="kecil"> Penulisan yang benar dari tulisan diatas adalah <p
class="putih" ID="kecil">
```

5. Elemen Yang Wajib di XHTML

Sebuah dokumen XHTML harus memiliki beberapa elemen wajib, yaitu deklarasi doctype, tag HTML, tag head dan tag body. Contoh kode yang wajib di dokumen XHTML adalah:

```
<!DOCTYPE html PUBLIC
"-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head><title></title></head><body></body>
</html>
```

3.4 Penggunaan XHTML

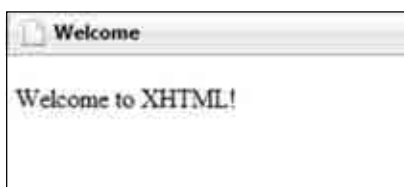
Dibandingkan dengan memakai HTML, pemakaian XHTML menjadikan sebuah halaman web akan benar-benar dirender (dibaca) oleh browser dengan benar. Pembagian area pada kode XHTML menyebabkan Source **XHTML** akan mudah dibaca karena ada pemisahan antara presentasi dan data. Perbedaan HTML dan XHTML terletak pada kode penulisan tag html, jika html biasa sebuah halaman web didahului dengan <html> dan di akhiri dengan </html>, sedangkan pada xhtml, kode <html> diganti menjadi <html xmlns = "http://www.w3.org/1999/xhtml">.

Penggunaan XHTML antara lain

1. Pembuatan halaman XHTML, diperlihatkan pada kode dibawah ini:

```
<?xml version = "1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<head> <title>XHTML</title></head>
<body><p>Welcome to XHTML!</p></body>
</html>
```

Hasil dari kode XHTML diatas ditunjukkan pada gambar 3.1 (tidak ada perbedaan dengan kode HTML biasa).



Gambar 3.1 Hasil XHTML

2. Pembuatan elemen header

Terdapat enam elemen header (mirip dengan HTML), penggunaannya ditunjukkan pada kode dibawah ini:

```
<?xml version = "1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title>Elemen Header </title></head>
<body>
<h1>Level 1 Header</h1>
<h2>Level 2 header</h2>
<h3>Level 3 header</h3>
<h4>Level 4 header</h4>
<h5>Level 5 header</h5>
<h6>Level 6 header</h6>
</body>
</html>
```

Hasil dari kode XHTML diatas ditunjukkan pada gambar 3.2 (tidak ada perbedaan dengan kode HTML biasa).



Gambar 3.2 Elemen Header

3. Pembuatan pertautan halaman, sama dengan HTML, pertautan memakai tag hyperlink, contoh kode dibawah ini:

```
<?xml version = "1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title>Pertautan - Links</title> </head>
  <body>
    <h1>Situs terpopuler</h1>
    <p><strong>Klik untuk mengunjungi situs</strong></p>
    <p><a href = "http://detik.com">detik</a></p>
    <p><a href = "http://kompas.com">Kompas</a></p>
    <p><a href = "http://jpnn.com">Jawapos</a></p>
  </body>
</html>
```

Hasil dari kode XHTML diatas ditunjukkan pada gambar 3.3 (tidak ada perbedaan dengan kode HTML biasa).



Gambar 3.3 Pertautan

4. Pembuatan tabel, beberapa elemen tambahan yang dikenal di dalam XHTML. Tabel didalam XHTML dianggap memiliki judul, header, body dan footer, sehingga elemen dapat didefinisikan untuk judul memakai <caption>, header memakai <thead>, body memakai <tbody> dan footer memakai <tfoot>, berikut ini contoh memakai tabel sederhana:

```
<?xml version = "1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title>Tabel XHTML Sederhana</title></head>
  <body>
    <tabel border = "1" width = "15%">
      <caption>Daftar Harga Barang</caption>
      <thead><tr><th>Barang</th><th>Harga</th></tr></thead>
      <tbody>
        <tr><td>Buku</td><td>2500</td></tr>
        <tr><td>Pensil</td><td>1500</td></tr>
        <tr><td>Mouse</td><td>5500</td></tr>
      </tbody>
      <tfoot> <tr><th>Total</th><th>59000</th></tr></tfoot>
    </tabel>
  </body>
</html>
```

Hasil dari kode XHTML diatas ditunjukkan pada gambar 3.4 (terdapat tambahan elemen dari HTML biasa).



Barang	Harga
Buku	2500
Pensil	1500
Mouse	5500
Total	59000

Gambar 3.4 Tabel Sederhana

Selain tambahan elemen penulisan, pada pengelompokan sel dari tabel juga terdapat tambahan tag, yaitu `colgroup` (pengelompokan dan penformatan kolom), elemen `col` dengan atribut: `align` (perataan), `span` (pemformatan jumlah kolom atau baris tertentu), `valign` (perataan secara vertical).

Bab 4

Cascading Style Sheets (CSS)

Sebuah CSS (*Cascading Style Sheets*) merupakan konsep untuk menentukan gaya penulisan pada penampilan halaman web seperti pengaturan, spasi, margin, dll. Penulisan kode CSS dapat diletakkan terpisah dari struktur dokumen, sehingga dengan adanya pemisahan struktur dari konten maka pengelolaan halaman web akan menjadi lebih baik. Selain itu perubahan pada CSS dapat dilakukan lebih mudah.

Saat ini, banyak Tag HTML dasar yang telah usang, dikarenakan fungsinya telah banyak digantikan oleh CSS. Sebuah Cascading Style Sheets (CSS) adalah cara yang mudah untuk memberikan tampilan halaman web tanpa harus menulis berulang-ulang kode tampilan tersebut. Terdapat tiga macam penulisan CSS yang dapat dipakai, tiga macam itu adalah

1. Pemakaian Secara InLine
2. Pemakaian Tag Style
3. Pemakaian Eksternal Style Sheet

Kode CSS membuat halaman lebih mudah diatur dengan pemisahan kode tampilan sebagai CSS dengan tag HTML sebagai isi dari halaman web. Perbedaan pemakaian CSS dengan tanpa CSS adalah jika tanpa memakai style sheets maka browser sepenuhnya mengontrol tampilan halaman Web. Namun jika, memakai style sheets, perancang (*designer*) dapat menentukan tampilan dan nuansa halaman Web secara bebas.

4.1 Pemakaian Secara Inline

Pemakaian secara Inline adalah sebuah cara instan, dimana CSS langsung dilekatkan pada tag HTML yang lain, Misalnya `<p style=> </p>` . Sebuah CSS di definisikan pada sebuah tag HTML, pada contoh ini adalah sebuah `<p>` seperti dibawah ini:

```
<p style="color:Green; font-family: arial; font-weight: bold">This paragraph uses the font 'arial' in red and bold</p>
```

Hasil yang tampak pada browser berupa text yang tertata dengan warna hijau dan jenis font yang tebal, seperti gambar 4.1



Gambar 4.1 Halaman CSS dengan Tag p

Pemakaian diatas dapat digabungkan dengan Tag HTML yang lain, sehingga pengaturan penampilan ini dapat memberikan hasil berbagai macam tipe pada satu alinea. Contoh berikut ini adalah sebuah tag <p> yang digabungkan dengan tag

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head> <title></title></head>
<body>
<p>This paragraph is divided up using the 'span' tag.
<span style="background-color:Orange; color: Maroon">Background Color.</span>
<span style="font-family: courier; font-size: medium">Medium Courier.</span></p>
<p style = "font-size: 20pt">Here is some more text</p>
<p style = "font-size: 20pt; color: #0000FF">Even more text</p>
</body></html>
```

Sebuah tag <p> biasa tanpa pengaturan, kemudian diberikan beberapa pengaturan yang berbeda dalam satu kalimat lewat , dimana pengaturan pertama adalah text dengan warna latar belakang hitam dan pengaturan kedua dengan jenis font courier berukuran medium. Hasil dari dua pengaturan dapat dilihat pada gambar 4.2:



Gambar 4.2 Halaman CSS dengan Tag span

Semua tag HTML dapat ditambahkan kode CSS dengan memanfaatkan tag atau tag <style> kecuali tag berikut ini: <BASE>, <BASEFONT>, <HEAD>, <HTML>, <META>, <PARAM>, <SCRIPT>, <STYLE>, and <TITLE>. Penulisan Inline style berarti masing-masing gaya elemen dideklarasikan menggunakan atribut STYLE. Setiap CSS properti diikuti oleh sebuah titik-dua dan nilai dari atribut. Beberapa properti dipisahkan oleh titik koma. Contoh penulisan: <p style = "font-size: 20 pt; color: #0000FF">. Penulisan secara inline styles berarti akan meng-override/menggantikan style lainnya.

4.2 Pemakaian Tag Style

Penulisan dengan memakai Tag Style, CSS didefinisikan cukup sekali pada sebuah halaman, kemudian halaman itu dapat disisipkan kedalam sebuah halaman web. Bisa juga definisi CSS dilakukan di awal halaman, kemudian di pakai di dalam halaman. Sebuah style

sheet pada bagian header, diawali dengan `<style type = "text/css">`. Semua style yang ditempatkan pada bagian ini akan diterapkan pada semua bagian dokumen. Sebagai catatan tipe yang lain pada atribut TYPE dapat berupa tipe MIME. Sebuah tipe MIME adalah sebuah standar untuk menentukan format konten. Tipe-tipe MIME lainnya mencakup text/html, image/gif and text/javascript . Tiap aturan pada body dimulai dan diakhiri dengan kurung kurawal ({ dan }). Pendeklarasian kelas ditulis terlebih dahulu dan diaplikasikan hanya pada elemen dimana kelas tersebut dicantumkan. Tiap property diikuti oleh titik dua dan kemudian nilai dari property tersebut. Lebih dari satu property atau Multiple properties dapat dipisahkan oleh titik koma. Contoh berikut ini adalah sebuah HTML yang memakai CSS pada tag `<h1>.../<h1>` dengan jenis tebal dan warna hijau. Suatu tag `<p>.../<p>` dengan ukuran huruf = 2, face arial, dan warna ungu, sebuah kelas CUSTOM dipakai untuk membuat jenis huruf besar dengan warna orange.

```
<head> <title></title>
  <style type = "text/css">
    H1{ color: #008000; font-weight: bold }
    P { font-family: Arial; color: #800080; font-size: 14px }
    .CUSTOM { font-size: large; color: #FF8000 }
  </style>
</head>
```

Kode CSS diatas memakai Tag style dengan type adalah CSS, definisi tersebut diletakkan diantara tag `<head> ... </head>`, kemudian label yang mendefinisikan sebagai kelas yaitu H1, P dan Custom. Ketiga kelas yang telah didefinisikan, digunakan untuk pengaturan gaya tampilan pada baris berikutnya. Contoh selengkapnya kode CSS dan HTML dibawah ini:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head> <title></title>
  <style type = "text/css">
    h1{ color: #008000; font-weight: bold }
    p { font-family: Arial; color: #800080; font-size: 14px }
    .CUSTOM { font-size: large; color: #FF8000 }
  </style>
</head>
<body>
<h1>Heading</h1><p>In this simple example, we use CSS to define some tag
attributes for one page</p>
<h1>Heading</h1>
<span class="CUSTOM">This is my custom style.</span>
</body></html>
```

Hasil dari pengaturan diatas dapat dilihat pada gambar 4.3:



Gambar 4.3 Halaman CSS dengan Tag style

4.3 Pemakaian Eksternal Style Sheet

Pengaturan CSS dapat dilakukan terpisah dengan file HTML, sehingga sebuah CSS dapat dipakai berulang-ulang pada banyak file HTML. Setelah sebuah file CSS dibuat, maka file HTML yang ingin memanfaatkan pengaturan dengan file CSS tersebut, dihubungkan dengan fasilitas LINK. Fasilitas LINK adalah sebuah Eksternal Style Sheets yang menghubungkan halaman terpisah dengan file CSS. Dengan fasilitas LINK, semua halaman web dapat menggunakan style sheet yang sama. Keuntungan yang didapatkan dengan memanfaatkan fasilitas LINK ini adalah dengan hanya memodifikasi satu file CSS untuk mengubah gaya pada halaman web, maka semua halaman yang terhubung dengan CSS ikut pula berubah.

Elemen LINK menentukan hubungan (REL atribut) antara dokumen ini dan dokumen lain dengan format LINK adalah :

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

Elemen LINK hanya dapat ditempatkan pada header. Contoh berikut ini adalah memakai fasilitas LINK dengan File CSS bernama mystyle.css

```
body { background-color: #CCFFCC; font-family: Arial,
      Delvetrica, sans-serif; color: #330066; padding: 50px, 70px}
a:link { color: #CC9900 }
a:visited { color: #660000 }
a:hover { color: #FFCC00 }
a:active { color: #FF0000 }
h1 { color: #996633; background-color: #FFFFCC }
```

Pada file HTML yang dibuat untuk semua pengaturan CSS, pada tag <head>...</head> dituliskan sebuah alamat LINK yaitu

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title></title>
  <link rel = "stylesheet" type = "text/css" href = "mystyle.css"/>
</head>
<body>
<h1>Heading</h1>
<p>silahkan kunjungi <a href="http://www.yahoo.com" > yahoo </a></p>
</body></html>
```

Hasil dari pengaturan diatas dapat dilihat pada gambar 4.4:



Gambar 4.4 Halaman CSS dengan eksternal Tag style

4.4 Pemakaian Style Sheet

Pemakaian CSS secara umum banyak diterapkan pada saat ini. Biasanya pengaturan CSS adalah pada jenis font antara lain :

1. **Generic font families**, pemakaian font famili ini dipakai untuk menentukan jenis font selain font yang ditentukan sebelumnya.
2. **Properti font-size** Selain masalah jenis font, biasanya ukuran font juga diberikan nilai tertentu sebagai ukuran relative: xx-small, x-small, small, smaller, medium, large, larger, x-large and xx-large

Beberapa pengaturan dari attribute CSS yang dapat dipakai diperlihatkan pada tabel 4.1:

Tabel 4.1 Daftar Pengaturan Atribut CSS

margin-top	Pengaturan top margin dari sebuah obyek. Semua nilai margin dapat dimasukkan sebagai nilai numeric atau nilai persentase.
margin-right	Pengaturan right margin dari sebuah obyek
margin-bottom	Pengaturan bottom margin dari sebuah obyek
margin-left	Pengaturan left margin dari sebuah obyek
margin	Pengaturan margin top, right, bottom dan left.
padding-top	Pengaturan top padding dari sebuah obyek. Semua nilai padding dapat dimasukkan sebagai nilai numeric atau nilai persentase.
padding-right	Pengaturan right padding dari sebuah obyek
padding-bottom	Pengaturan bottom padding dari sebuah obyek
padding-left	Pengaturan left padding dari sebuah obyek
padding	Pengaturan nilai padding sebagai top, right, bottom and left.
color	Pengaturan warna (biasanya warna text) dengan memakai kode warna HTML RGAMBAR (triplet codes).
background-color	Pengaturan warna latar belakang dengan memakai kode warna HTML RGAMBAR (triplet codes).
background-image	Pengaturan gambar latar belakang.
border-style	Pengaturan border style dari pilihan yaitu none, dotted, dashed, solid, double, groove, ridge, inset, dan outset. Borders dapat diset secara individually dengan memakai border-bottom-style, border-left-style, border-right-style, border-top-style.
border-color	Pengaturan warna border dengan memakai kode HTML RGAMBAR (triplet codes).
border-width	Mendefinisikan tipe border-width dari pilihan yang ada yaitu thin, medium, thick. Nilai ini dapat sebagai nilai panjang. Secara individual, tipe border dapat di pilih dengan memakai border-bottom-width, border-top-width, border-left-width, and border-right-width.
border	Pengaturan secara sama pada width, color, style pada pilihan semua border dari sebuah obyek.
font-family	Pengaturan font face. Nilai ini dapat lebih dari satu, sehingga komputer dapat menyesuaikan jenis font yang tersedia.
font-size	Pengaturan Font size. Isi iilai dapat berupa absolute, relative, atau percentage.
font-weight	Nilai dari font: normal, bold, bolder, atau lighter.
font-variant	Nilai dari font: small-caps or normal

Didalam pengisian nilai-nilai margin, size dan sebagainya terdapat beberapa isitilah yang biasanya disingkat, seperti px, em, ex dan sebagainya. Hal itu merupakan kekuatan dari CSS yang sederhana untuk dituliskan sebagai ganti dari tulisan sebenarnya:

- px: pixel adalah pengukuran relatif panjang. Bervariasi dalam ukuran yang didasarkan pada resolusi layar
 - em: ukuran font
 - ex: "x-height" dari font, biasanya diatur dengan tinggi huruf kecil
- Pengaturan nilai dapat berupa nilai absolute, relative dan persentase. Nilai absolute atau mutlak adalah pengukuran panjang-dalam: inci-cm: cm, mm: millimeter, pt: poin (1 pt = 1 / 72), pc: picas (1 pc = 12 pt).

4.5 Pemakaian Elemen Style Sheet

Elemen yang ada pada sebuah halaman web dapat diatur lebih fleksibel dengan memanfaatkan CSS. Pengaturan elemen tersebut dapat berupa posisi elemen, latar belakang elemen dan sebagainya.

4.5.1 Posisi elemen

Posisi elemen pada sebuah halaman diatur pada property CSS yang biasanya bernilai sebagai absolute positioning. Posisi elemen ini mengatur ulang dari posisi yang telah ada dengan cara menetapkan posisi elemen yang baru. Posisi elemen menurut jarak dari atas, kiri, kanan atau bawah margin dari elemen induk. Selain pengaturan posisi elemen, penampakan sebuah obyek dapat diatur dengan mekanisme atribut z-index. Suatu atribut z-index adalah pengaturan lapisan elemen yang mengatur penampakan secara tumpang tindih. Elemen dengan z-index yang lebih tinggi akan ditampilkan di depan elemen dengan z-index yang lebih rendah. Contoh program untuk mengatur posisi absolute dan z-index ditunjukkan pada program dibawah ini:

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

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title></title>
</head>
<body>

<h1 style= "position: absolute; top: 50px; left: 50px;z-index: 3">Positioned
Text</h1>

</body>
</html>
```

Hasil dari pengaturan dengan posisi absolute dan atribut z-index, tampak sebuah urutan penampakan obyek yang tumpang tindih, dimana tulisan text dengan atribut z-index tertinggi menjadi obyek yang tampak diatas obyek yang lain, seperti diperlihatkan pada gambar 4.5.

Posisi relative sebuah posisi pada browser yang dijabarkan sebagai nilai offset pada sebuah elemen terhadap halaman. Nilai offset elemen ditentukan sebagai atas, bawah, kiri atau kanan nilai-nilai.

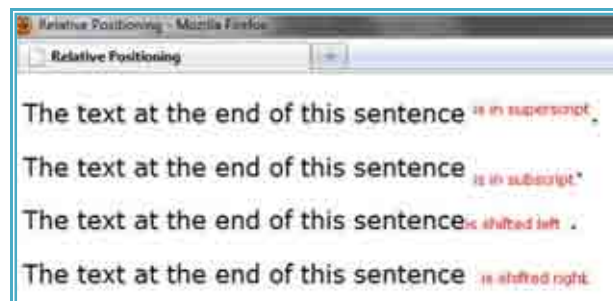


Gambar 4.5 Pengaturan absolute dan z-index

Berikut ini sebuah contoh program yang menggambarkan tentang posisi relative suatu obyek:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>Relative Positioning</title>
<style type = "text/css">
p { font-size: 1.3em;font-family: verdana, arial, sans-serif}
span { color: red; font-size: .6em;height: 1em }
.super { position: relative; top: -1ex }
.sub { position: relative; bottom: -1ex }
.shiftleft { position: relative; left: -1ex }
.shiftright { position: relative;right: -1ex }
</style>
</head>
<body>
<p>The text at the end of this sentence
<span class = "super">is in superscript</span>.</p>
<p>The text at the end of this sentence
<span class = "sub">is in subscript</span>.</p>
<p>The text at the end of this sentence
<span class = "shiftleft">is shifted left</span>.</p>
<p>The text at the end of this sentence
<span class = "shiftright">is shifted right</span>.</p>
</body>
</html>
```

Hasil dari pengaturan dengan posisi relative suatu obyek digambarkan pada gambar 4.6:



Gambar 4.6 Hasil pengaturan posis relative

4.5.2 Pengaturan latar belakang

Latar belakang dari halaman web dapat disisipkan dengan sebuah image. Hal ini dapat dilakukan dengan mengatur property dari `background-image`. Pengaturan lain pada `background` terletak pada posisi latar belakang. Pengaturan ini dapat ditentukan lewat property `background-position`. Nilai setting ini menentukan posisi gambar pada halaman sebagai nilai atas, bawah, tengah, kiri atau kanan. Contoh: `background-position: 50% 30px;`

Posisi gambar vertikal terpusat (50% dari jarak di layar) dan 30 pixel dari atas. Pilihan lain pada property `background` adalah pengaturan perulangan dengan beberapa pilihan yaitu `background-repeat` property controls tiling sebagai `no-repeat`, `repeat`, `x-repeat`, `y-repeat`.

Pilihan berikutnya pada properti `background-attachment` dengan pilihan: `fixed` atau `scrolling`.

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

<html xmlns="http://www.w3.org/1999/xhtml">
<head> <title>Relative Positioning</title>
<style type = "text/css">
body {background-image:url(circle.gif);
      background-position:right bottom;
      background-repeat:no-repeat;
      background-attachment:fixed
    }
</style>
</head><body>
  Halaman web dengan latar belakang gambar
</body></html>
```

Hasil dari pengaturan halaman web dengan latar belakang sebuah gambar digambarkan pada gambar 4.7:



Gambar 4.7 Hasil pengaturan dengan latar belakang gambar

4.5.3 Dimensi Elemen

Elemen sebagai nilai paling kecil pada sebuah sumber halaman web. Elemen ini dapat diatur dengan tag `<div>` dan tag ``

1. Elemen SPAN, pengelompokan elemen secara generic. Kegunaan utama adalah untuk menerapkan gaya atau ID atribut untuk blok teks.
2. Elemen DIV, serupa dengan SPAN, tapi lebih khusus pada pengaturan secara *block - level element*. Ditampilkan pada baris sendiri dengan margin atas dan di bawah.

Dimensi elemen memiliki beberapa property yang dapat diatur

- Pengaturan lebar elemen dengan memakai tag width, contoh `<DIV STYLE = "width: 20%">`
- Pengaturan tinggi elemen dengan memakai property height
- Pengaturan perataan suatu teks, diatur melalui properti text-align. Pilihan yang ada antara lain rata kiri: left, rata kanan: right dan tengah:center
- Pengaturan scroll pada teks yang tidak cukup pada kotak yang telah disediakan. Pengaturan ini dapat memakai properti overflow, yang ketika diberikan sebuah nilai, maka pada kotak akan muncul scroll tambahan pada scrollbars jika teks yang ada melebihi dari batas-batas yang ada.

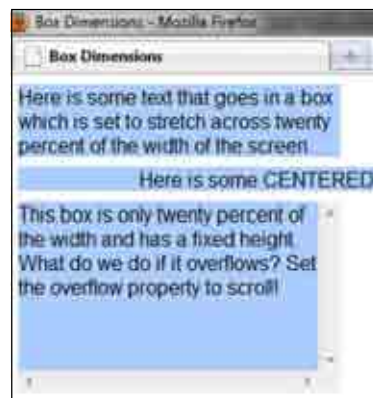
Contoh pemakaian dimensi elemen ditunjukkan pada program dibawah ini:

```
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <title>Box Dimensions</title>
    <style type = "text/css">
      div { background-color: #aaccff;
            margin-bottom: .5em;
            font-family: arial, helvetica, sans-serif }
    </style>
  </head>
  <body>
    <div style = "width: 20%">Here is some
    text that goes in a box which is
    set to stretch across twenty percent
    of the width of the screen.</div>

    <div style = "width: 80%; text-align: center">
    Here is some CENTERED text that goes in a box
    which is set to stretch across eighty percent of
    the width of the screen.</div>

    <div style = "width: 20%; height: 150px; overflow: scroll">
    This box is only twenty percent of
    the width and has a fixed height.
    What do we do if it overflows? Set the
    overflow property to scroll!</div>
  </body>
</html>
```

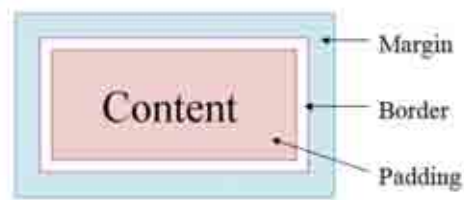
Hasil dari pengaturan halaman web dengan dimensi elemen yang mengatur lebar dan tinggi suatu area digambarkan pada gambar 4.8:



Gambar 4.8 Hasil pengaturan dengan latar belakang gambar

4.5.4 Aliran Text dan model Box

Text yang dituliskan pada halaman web biasanya dituliskan dari kiri ke kanan. Posisi text terhadap text lainnya dapat diatur sedemikian rupa sehingga posisi antar text menjadi mengambang (*float*) terhadap posisi text yang lain. Model text yang mengambang memungkinkan sebuah text dipindah pada sisi layar dan text lain yang mengelilingi text yang dipindah tersebut dapat diatur batas diantara text tersebut. Batas-batas text dapat juga ditentukan berdasarkan tiga komposisi batas yaitu margin, padding dan border. Pengaturan tiga komposisi batas pada sebuah obyek disebut sebagai model box, hal ini diperlihatkan pada gambar 4.9



Gambar 4.9 Model Box

Contoh pengaturan text mengambang diperlihatkan pada program dibawah ini:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Flowing Text Around Floating Elements</title>
<style type="text/css">
  div.heading
  {
    background-color: #bbddff;
    text-align: left;
    font-family: arial, helvetica, sans-serif;
    padding: .2em;
  }
  p
  {
    text-align: justify;
    font-family: verdana, geneva, sans-serif;
    margin: .5em;
  }
  div.floated
  {
    background-color: #eeeeee;
    font-size: 1.5em;
    font-family: arial, helvetica, sans-serif;
    padding: .2em;
    margin-left: .5em;
    margin-bottom: .5em;
    float: right;
    text-align: right;
    width: 50%;
  }
  div.section
  {
    border: 1px solid #bbddff;
  }
</style>
</head>
<body>
  <div class="heading">
    
```

```
</div>
<div class="section">
  <div class="floated">
    Corporate Training and Publishing</div>
  <p>
    The 1st International Conference on Industrial
    Engineering and Service Science (IESS2011) will
    be held in September, 2011 in Surakarta, Indonesia.
  </p>
</div>
<div class="section">
  <div class="floated">
    Leading-Edge Programming Textbooks</div>
  <p>
    Through its publishing
  </p>
</div>
</body>
</html>
```

Hasil dari pengaturan halaman web dengan text yang mengambang digambarkan pada gambar 4.10:



Gambar 4.10 Hasil pengaturan dengan Text Mengambang

Bab 5

JAVA SCRIPT

JavaScript merupakan sebuah bahasa scripting yang dikembangkan oleh Netscape. Dengan JavaScript kita dapat dengan mudah membuat halaman web yang interaktif. Javascript dapat dibuat dan berjalan dengan baik pada browser yang telah ada sampai saat ini.

5.1 Memakai JavaScript

Javascript, didisain sebagai bahasa pemrograman *lightweight* yang dapat dipakai untuk menambahkan halaman web menjadi lebih dinamis. Tidak sama dengan dengan bahasa pemrograman yang umum seperti c, C ++, Visual Basic, dan Java—pada javascript tidak memerlukan lingkungan khusus didalam menulis aplikasi Javascript. Editor teks biasa dapat dipakai untuk menciptakan halaman Web. Untuk memasukkan satu skrip ke dalam satu halaman Web yang telah ada ada sebelumnya, javascript dimasukkan dalam kode tag `<script> ... </script>`. Tag ini berarti browser akan melaksanakan kode javascript tersebut. Sebagai satu contoh, di sini adalah satu halaman Web sederhana, kode JavaScript dituliskan langsung pada halaman HTML.

```
<html>
<body> HTML normal
<script language="JavaScript">
    document.write("Hello JavaScript") </script>
    Tulisan HTML
</body>
</html>
```

Kode Javascript dimulai dengan `<script> </script>`, seperti contoh diatas javascript adalah dibawah ini:

```
<script language="JavaScript">
    document.write("Hello JavaScript ")
</script>
```

Pemakaian tag `<script>` berarti segala sesuatu yang berada di antara tag `<script>` dan `</script>` diterjemahkan sebagai kode JavaScript. Di situ kita bisa melihat penggunaan

`document.write()` - salah satu perintah yang penting dalam pemrograman dengan JavaScript. `document.write()` digunakan untuk menulis sesuatu pada dokumen (dalam hal ini adalah dokumen HTML). Jadi program JavaScript sederhana kita di atas berfungsi untuk menuliskan text "Hello JavaScript!" ke dokumen HTML.

5.1.1 Non-JavaScript browser

Beberapa browser jaman dulu memiliki versi yang tidak kompatibel dengan javascript, agar menghindari kesalahan dari pembacaan script, maka kode javascript ditambahkan kode penjejak `<!-- -->`. Kode tersebut biasa didalam HTML adalah sebuah kode untuk memberikan komentar. Pada contoh sebelumnya, untuk menunjang browser yang tidak kompatibel dengan javascript, kode diatas ditambahkan sehingga menjadi seperti dibawah ini:

```
<html><body>
HTML normal
<script language="JavaScript">
<!-- hide from old browsers
    document.write("Hello JavaScript")
    // --> </script>
Tulisan HTML
</body>
</html>
```

5.1.2 Interaksi Pengguna dengan JavaScript

JavaScript memiliki tiga perintah sederhana untuk berinteraksi dengan pengguna. Penulisan informasi ke pengguna melalui halaman web dapat dikerjakan dengan salah satu perintah dalam javascript yakni:

- Penulisan dokumen dengan perintah `document.write("informasi ");`
- Penampilan jendela informasi dengan perintah `alert("informasi");`
- Meminta isian dari pengguna dengan perintah `prompt("isian");`

Contoh program yang melakukan interaksi dengan pengguna diperlihatkan dibawah ini:

```
<?xml version = "1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title>Interaksi Dengan JavaScript</title>
<script type = "text/javascript">
<!--
var bil1, bil2, jumlah;
bil1 = window.prompt("Masukkan sebuah bilangan ", "0");
bil2 = window.prompt("Masukkan sebuah bilangan ", "0");
bil1 = parseInt(bil1);bil2 = parseInt(bil2);
jumlah = bil1 + bil2;
alert("Hasil dari " + bil1 + " + " + bil2 + " adalah " + jumlah);
document.writeln("<h1>Jumlah nya " + jumlah + "</h1>");
// -->
</script>
</head><body>
    <p>klick Refresh (atau Reload) untuk menjalankan lagi</p>
</body>
</html>
```


Hasil dari program diatas berupa dua inputan yang harus diisi oleh pengguna, kemudian dua inputan tersebut dijumlahkan dan hasilnya ditampilkan di halaman, seperti pada gambar 5.1



Gambar 5.1 Interaksi dengan Pengguna

5.2 Konsep Memori

Pemesanan memori yang biasanya dipakai untuk membuat variabel, dapat dilakukan untuk sembarang tipe variabel. Dengan kata kunci `var`, yang berarti deklarasi variabel, kemudian diikuti dengan nama variabel, maka nama tersebut menunjukkan sebuah korespondensi ke alamat memori komputer. Sebuah variabel dapat diberi nama dengan sembarang identitas, yaitu sederetan karakter, huruf, digit, underscores ('_') dan tanda dollar ('\$'). Penamaan variabel tidak dapat diawali dengan digit.

Contoh penamaan yang valid: `halo`, `$x`, `_nilai`, sedangkan penamaan yang tidak valid diawali dengan digit, misalnya `5kode` atau mengandung karakter `\` atau `#` seperti berikut `varia\bel` atau `data#1`. Sebuah variabel bersifat case-sensitive.

Biasanya penamaan variabel dibedakan untuk setiap kata, dimana awal dari kata adalah huruf besar yang dimaksudkan untuk membedakan antara kata-kata dalam variabel, contohnya `gajiPegawai`, `nilaiMaksimum` dan sebagainya. Ketika sebuah variabel diberikan nilai, maka nilai lama akan digantikan dengan nilai baru (*overwrite*).

5.3 Operasi Aritmatika Pada JavaScript

Seperti pada bahasa pemrograman lain, maka javascript menyediakan operasi yang dapat dilakukan terhadap nilai bilangan. Operasi ini disebut operasi aritmatika, secara umum operasi ini diperlihatkan pada tabel 5.1.

Tabel 5.1 Operasi Aritmatika

Operasi JavaScript	Operator
Penambahan	+
Pengurangan	-
Perkalian	*
Pembagian	/
Bagi Sisa (modulo)	%

Sifat dari operasi aritmatika dari kanan ke kiri dengan derajat urutan operasi seperti yang diperlihatkan pada tabel 5.2

Tabel 5.2 Derajat Operasi

Operator	Derajat
()	Parenthese, derajat yang akan dikerjakan terlebih dahulu, jika terdapat operator yang bersarang (nested), maka sarang yang paling dalam dikerjakan terlebih dahulu, jika operator sederajat, maka dikerjakan dari kiri ke kanan
/, * atau %	Jika terdapat operator yang lebih dari satu, maka dikerjakan dari kiri ke kanan
+ atau -	Jika terdapat operator yang lebih dari satu, maka dikerjakan dari kiri ke kanan

Operasi yang dilakukan tidak hanya terbatas pada bilangan saja, pada sebuah string, operator '+' dapat diartikan sebagai operasi concat, dimana sebuah string akan disambung dengan string yang lain. Umumnya operasi dilakukan dengan menyimpan sebuah nilai kedalam variabel yang disebut sebagai Assignment Operators. Secara umum bentuk operasi assignment operators : $c = c + 3$ sama dengan $c += 3$, dua ekspresi tersebut sama-sama berfungsi untuk menambahkan nilai 3 kepada variabel c. Bentuk $c += 3$ hanya memiliki sedikit perbedaan dalam kecepatan eksekusi dengan bentuk $c = c + 3$, namun kecepatan eksekusi akan jauh berbeda ketika melibatkan jumlah operasi yang besar. Semua operator aritmatika dapat diperlakukan sebagai Assignment Operators, contohnya adalah $+=$, $-=$, $/=$ dan $*=$. Dua operator yakni + dan - dapat dikembangkan menjadi operasi *decrement* dengan operator '++' dan *increment* dengan operator '--'.

5.4 Struktur Kontrol Pada JavaScript

Struktur kontrol dalam javascript dibagi menjadi bagian :

- Struktur Percabangan (*Branching structures*)
- Struktur Perulangan (*Looping controls*)

5.4.1 Struktur Percabangan

Terdapat tiga tipe percabangan dengan kondisi yang dimasukkan. Seleksi berdasarkan data yang dimasukkan dengan data pembandingan. Tiga tipe dari struktur seleksi antara lain:

1. Struktur Percabangan Tunggal

Struktur ini dituliskan sebagai bentuk statemen if yaitu *Single-selection structure*. Digunakan untuk menyeleksi suatu kondisi tunggal. Bila proses yang diseleksi terpenuhi atau bernilai benar, maka pernyataan yang ada di dalam blok if akan diproses dan dikerjakan. Bentuk umum dari if adalah

```
if (kondisi) {pernyataan;}
```

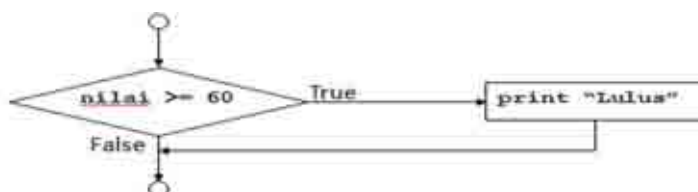
Kondisi terdiri dari [masukan] [operator] [datapembandingan]

- Masukan merupakan variabel, keadaan sistem
- Operator antara lain $>$, $<$, $>=$, $<=$, $<>$
- Datapembandingan berupa data konstan, data sistem yang sesuai dengan tipe data dari masukan

Contoh program adalah pada Pseudocode: *Jika nilai mahasiswa lebih besar dari atau sama dengan 60 Print "lulus"*. JavaScript statement:

```
if( grade >= 60 ) document.writeln( "Lulus" );
```

Bentuk Flowchart:



2. Struktur Percabangan Majemuk

Struktur ini dituliskan sebagai bentuk pasangan statemen if dan else yaitu if/else (*Double-selection structure*). Dalam if....else minimal terdapat dua pernyataan. Bila kondisi yang diperiksa bernilai benar, maka pernyataan pertama yang akan dilaksanakan dan jika kondisi yang diperiksa bernilai salah maka pernyataan yang kedua yang akan dilaksanakan.

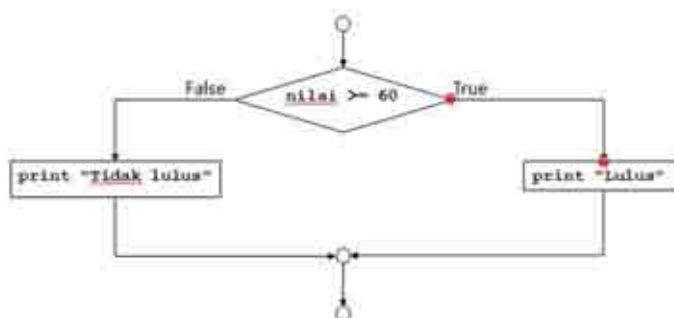
Bentuk umum if...else adalah :

```
if(kondisi)
    {pernyataan1;}
else
    {pernyataan2;}
```

Contohnya adalah pada Pseudocode: *Jika nilai mahasiswa lebih besar dari atau sama dengan 60 Print "Lulus" else Print "Tidak lulus"*. JavaScript statement:

```
if ( grade >= 60 )
    document.writeln( "Lulus" );
else
    document.writeln( "Tidak lulus" );
```

Bentuk Flowchart:



Percabangan bentuk ini biasanya dipakai dalam bentuk bersarang (nested), contoh berikut ini adalah sebuah persoalan yang memakai bentuk percangan dengan bersarang:

Pseudocode:

```

If nilai mahasiswa lebih besar atau sama dengan 90
  Print "A"
else If Nilai mahasiswa lebih besar atau sama dengan 80
  Print "B"
else If nilai mahasiswa lebih besar atau sama dengan 70
  Print "C"
else If nilai mahasiswa lebih besar atau sama dengan 60
  Print "D"
else Print "F"
  
```

JavaScript statement:

```
if ( studentGrade >= 90 )
    document.writeln( "A" );
else if ( studentGrade >= 80 )
    document.writeln( "B" );
else if ( studentGrade >= 70 )
    document.writeln( "C" );
else if ( studentGrade >= 60 )
    document.writeln( "D" );
else
    document.writeln( "F" );
```

Selain bentuk percabangan yang telah dibahas diatas, maka percabangan pada satu statemen dengan memeriksa kondisi dari nilai variabel dapat dilakukan sebagai bentuk *Conditional Operator*. Bentuk umum:

```
condition ? expression1 : expression2;
```

Contoh

```
document.writeln(
    nilaiMahasiswa >= 60 ? "Lulus" : "Tidak lulus" );
```

Hasilnya sama seperti **if/else** pada contoh sebelumnya.

3. Struktur Percabangan Seleksi

Struktur ini dituliskan sebagai bentuk statemen switch (*Multiple-selection structure*) yakni : switch....case....default digunakan untuk penyeleksian nilai dengan kemungkinan yang terjadi cukup banyak. Fungsi ini akan melaksanakan salah satu dari beberapa pernyataan 'case' tergantung nilai yang tertera di dalam switch. Selanjutnya proses diteruskan hingga ditemukan pernyataan 'break'. Jika tidak ada nilai pada case yang sesuai dengan nilai di dalam switch, maka proses akan diteruskan kepada pernyataan yang ada di 'default'.

Pernyataan switch-case dibuat untuk menangani keputusan yang meliputi beberapa pilihan. Tetapi, switch disini masih memiliki batasan yaitu:

- Variabel seleksi harus bertipe Integer, atau yang cocok dengan Integer seperti character, boolean, byte, short dan long.
- Batasan variabel yang bisa diperiksa antara 0 sampai 255.

Bentuk umum pernyataan switch-case adalah:

```
switch(variabel)
{
    case nilai1: pernyataan1; break;
    case nilai2: pernyataan2; break;
    case nilaiN: pernyataanN; break;
    default: pernyataan_default;
}
```

Nilai variabel akan dibandingkan dengan setiap nilai pada tag case. Apabila sesuai, maka seluruh statemen dibelakang case akan dieksekusi. Penulisan break menyebabkan statemen setelah break tidak dikerjakan. Jika tanpa pernyataan break diakhir case, maka eksekusi akan dilanjutkan ke case selanjutnya.

Contoh program adalah :

```
<?xml version = "1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title>Interaksi Dengan JavaScript</title>
<script type="text/javascript">
var d = new Date();var theDay = d.getDay();
switch (theDay) {
case 5:document.write("Jumat");break;
case 6:document.write("Sabtu");break;
case 0:document.write("Senin");break;
default: document.write("Hari Kerja!");
}</script>
</head>
<body> <p>klick Refresh (atau Reload) untuk menjalankan lagi</p></body>
</html>
```

5.4.2 Struktur Perulangan

Perulangan adalah statemen yang diulang dalam sebuah program, terdapat empat tipe struktur perulangan yakni while, do/while, for dan for/in. Secara umum perulangan ini terbagi menjadi dua tipe struktur perulangan yakni : For ... Loop dan Do while. Pemakaian perulangan ini tergantung dari jumlah perulangan apakah tertentu (telah diketahui jumlahnya) atau belum. Jika telah diketahui, maka memakai perulangan For ... Loop, sedangkan jika perulangan relatif terhadap sesuatu maka memakai Do while.

1. Statemen For ... loop

Sintaks penulisan adalah :

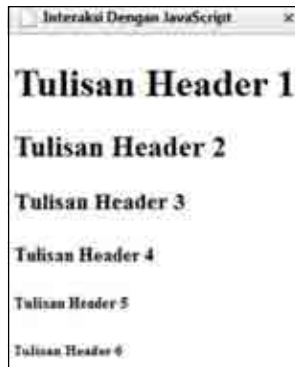
```
For (variabel=startvalue;
    variabel<=endvalue;variabel=variabel+increment)
{ kode-kode yang akan dieksekusi; }
```

Contoh:

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

<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title>Interaksi Dengan JavaScript</title></head>
<body>
<script type="text/javascript">
    for (var i = 1; i <= 6; i++) {
        document.write("<h" + i + "> Tulisan Header " + i + "</h>");
    }
</script>
</body>
</html>
```

Hasil dari program diatas berupa tulisan dengan bentuk header dari 1 sampai 6 yang ditampilkan di halaman, seperti pada gambar 5.2



Gambar 5.2 Tulisan Header

Khusus untuk menampilkan data-data yang ada di koleksi obyek memakai statemen `for ... in`, yang berarti tiap bagian dari koleksi, ditelusuri dan diakses satu persatu.

```
for (variabel in obyek)
  { kode yang akan dieksekusi; }
```

Contoh program

```
var mahasiswa={nrp:"001",nama:"tukul",umur:25};

for (x in mahasiswa)
  { document.write(person[x] + " "); }
```

2. Statemen Do ... While

Perulangan dengan perintah `do ... while` berarti kerjakan sampai syarat tertentu. Terdapat modifikasi `do ... while`, tergantung dari kapan syarat tersebut di periksa.

a. Statement While (pemeriksaan kondisi di awal)

Menggunakan statemen `while` dengan sebuah variabel counter yang di-incremen atau di-decremen, tiap kali looping sebagai pengecekan. Perulangan akan berhenti ketika counter mencapai nilai tertentu dan membuat kondisi menjadi false. Digunakan ketikai jumlah looping yang akan dilakukan sudah diketahui.

Sintaks penulisan adalah :

```
while (variabel<=endvalue)
  { kode yang akan dieksekusi; }
```

b. Statement Do While (pemeriksaan kondisi di akhir)

Pada perulangan memakai tersebut tidak diketahui secara pasti jumlah perulangan yang ada dan pemeriksaan dilakukan di akhir statemen. Biasanya dipakai untuk meminta inputan pengguna yang kemudian akan diperiksa nilai yang dimasukkan.

Sintaks penulisan adalah :

```
do
  { kode yang akan dieksekusi; }
while (variabel<=endvalue)
```

5.5 Event Pada JavaScript

Sebuah event dan event handler merupakan hal yang sangat penting dalam pemrograman JavaScript. Sebuah event adalah sesuatu yang terjadi karena aksi pemakai. Contohnya jika user men-click tombol mouse terjadilah event Click. Jika *mousepointer* bergerak melewati sebuah link terjadilah event MouseOver. Ada banyak event yang terdapat dalam JavaScript yang selengkapnya bisa di lihat pada *JavaScript Reference*.

Program JavaScript dapat dipergunakan untuk menangani reaksi jika terjadi suatu event tertentu. Ini bisa dilakukan dengan bantuan event-handlers. Sebagai contoh kita memiliki sebuah tombol yang akan mengeluarkan sebuah *popup window* jika di-tekan. Ini berarti bahwa sebuah *window popup* harus muncul sebagai reaksi dari event Click. Kode event-handler yang harus kita gunakan adalah `onClick`, yang memberitahukan apa yang harus dikerjakan oleh komputer jika event ini terjadi. Contoh di bawah ini menunjukkan bagaimana mudahnya menggunakan event-handler `onClick`:

```
<form>
<input type="button" value="Click me" onClick="alert('Hello')">
</form>
```

Sebuah form dan sebuah tombol dalam bahasa HTML standard, dan aksi pada tombol tersebut akan diarahkan ke sebuah fungsi javascript yaitu `alert`. Bagian `onClick="alert('Hello')"` yang berada di dalam tag `<input>` mendefinisikan event yang akan terjadi jika tombol ditekan. Ketika terjadi event Click (user menekan tombol **Click Me**, komputer akan mengeksekusi `alert('Hello')`, yang merupakan kode JavaScript (perhatikan: tanpa menggunakan tag `<script>` pada kasus ini). Fungsi `alert()` untuk menampilkan sebuah *window popup*. Di dalam tanda kurung yang berarti parameter dari fungsi `alert`, adalah sebuah string yang akan muncul pada window yang dimaksud. Jadi script di atas menampilkan window dengan tulisan *'Hello'* saat user men-click tombol.

Pemakaian double quotes (") menunjukkan blok string, ketika sebuah string menjadi bagian dari blok string tersebut, maka untuk membedakan blok string dengan isi string, dipakai single quotes ('). Pada contoh ini : `onClick="alert('Hello')"` - menjelaskan blok string dan sisi string (double dan single quote) . Ketika penulisan dengan semua memakai double quote seperti : `onClick="alert("Hello")"` maka komputer akan bingung mana yang menjadi bagian event-handler `onClick` dan mana yang bukan. Urutan penggunaannya tidak menjadi masalah. Jadi bisa juga kita tuliskan `onClick='alert("Hello")'`.

5.6 Fungsi Pada JavaScript

Fungsi dalam javascript merupakan cara untuk menyatukan beberapa perintah. Contoh berikut ini memperlihatkan sebuah perintah yang dikumpulkan menjadi satu fungsi

```
<html>
<script language="JavaScript"><!-- hide
document.write("Hello <br>");
document.write("JavaScript<br>");
document.write("Hello <br>");
document.write("JavaScript<br>");
document.write("Hello <br>");
document.write("JavaScript<br>");
// --></script>
</html>
```

Kode diatas terdapat suatu perintah yang ditulis berulang-ulang-ulang. Hal ini dapat disingkat menjadi sebuah fungsi sebagai berikut ini

```
<html>
<script language="JavaScript"><!-- hide
function myFunction() {
document.write("Hello <br>");
document.write("JavaScript<br>");}
myFunction();myFunction();myFunction();
// --></script>
</html>
```

Pada script di atas kita definisikan sebuah function yang dilakukan melalui baris-baris:

```
function myFunction()
{
    document.write("Hello <br>");
    document.write("JavaScript<br>");
}
```

Perintah-perintah di dalam tanda {} merupakan milik function `myFunction()`. Ini berarti ada dua perintah `document.write()` yang dijadikan satu dan dapat dieksekusi melalui pemanggilan function yang dimaksud. Pada contoh tersebut `myfunction()` ditulis sebanyak tiga kali dan berarti bahwa function akan dieksekusi sebanyak tiga kali. Sebuah fungsi dapat dikombinasikan dengan event-handler seperti pada contoh berikut ini:

```
<html>
<head>
<script language="JavaScript"><!-- hide
function calculation() {
    var x = 12;
    var y = 5;
    var result = x + y;
    alert(result);}
// -->
</script></head>
<body>
<form>
<input type="button" value="Calculate" onClick="calculation()">
</form>
</body>
</html>
```

Tombol `Calculate` akan memanggil fungsi `calculation()` jika di-click. Kita lihat bahwa function melakukan perhitungan tertentu sehingga kita perlu menggunakan variabel `x`, `y` dan `result`. Kita mendefinisikan variabel dengan keyword `var`. Variabels dapat digunakan untuk menyimpan harga-harga yang berbeda- seperti angka, text, strings dan lainnya.

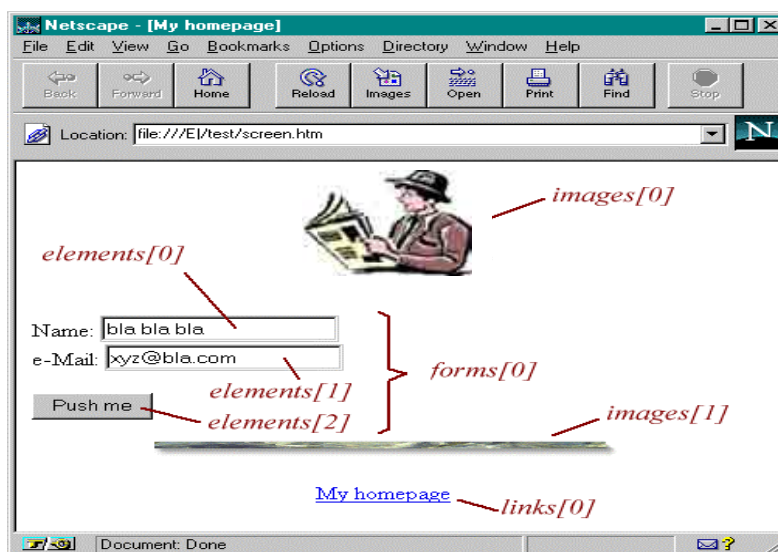
Baris `var result= x + y;` memberitahu browser untuk membuat variabel `result` dan menyimpan harga `x + y` (`5 + 12`) dalam variabel `result`. Setelah operasi ini isi variabel `result` adalah `17`. Perintah `alert(result)` artinya sama dengan `alert(17)`, yaitu *popup window* akan muncul dengan isi angka `17`.

5.7 Hirarki Pada JavaScript

Dalam sebuah halaman web terdapat bermacam-macam elemen seperti gambar (*image*), link, form, tombol, input text, dan sebagainya. JavaScript menyusun semua elemen halaman web dalam satu susunan hirarki. Setiap elemen dilihat sebagai obyek. Setiap obyek dapat memiliki beberapa properti (yang menentukan sifat dan tampilannya) dan method (yang menentukan apa yang bisa dikerjakan oleh obyek yang bersangkutan). Dengan bantuan JavaScript, programmer dapat dengan mudah memanipulasi obyek - obyek tersebut. Sebagai contoh kode berikut ini merupakan halaman HTML sederhana berikut ini.

```
<html>
<head></head>
<body bgcolor=#ffffff>
<center>
</center><p>
<form name="myForm">
Name: <input type="text" name="name" value=""><br>
e-Mail:<input type="text" name="email" value=""><br><br>
<input type="button" value="Push me" name="myButton"   onClick="alert('Yo')">
</form>
<p>
<center>
<p><a href="http://www.yaho.com/">My homepage</a> </center>
</body></html>
```

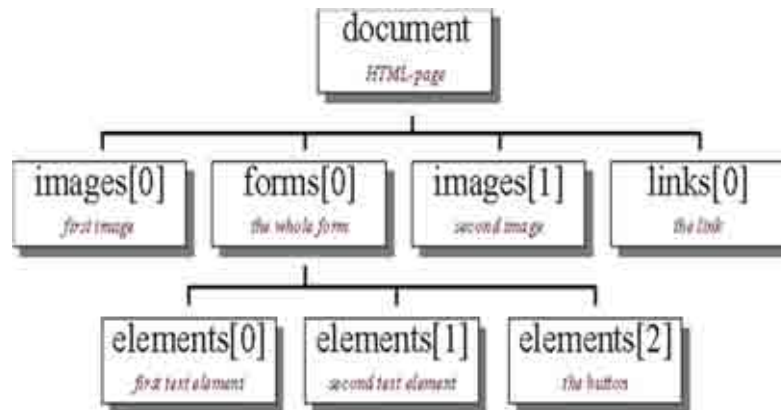
dengan tampilan seperti gambar 5.3



Gambar 5.3 Element Halaman Web

Terlihat dua image, satu link dan sebuah form dengan dua text field dan sebuah tombol. Dalam pandangan JavaScript window browser adalah sebuah obyek window. Sebuah obyek window berisi elemen-elemen seperti statusbar dll. Di dalam window, kode-kode HTML diterjemahkan dan dituliskan sebagai dokumen HTML, hingga munculah halaman web. Halaman ini merupakan sebuah obyek dokumen. Artinya obyek dokumen mewakili dokumen HTML yang di-load pada saat itu. Sebuah obyek dokumen merupakan obyek yang sangat penting dalam JavaScript. Terdapat sifat (*property*) dari obyek dokumen contohnya adalah warna latar belakang pada halaman.

Agar dapat memperoleh informasi tentang suatu obyek tertentu dan memanipulasinya, terlebih dahulu harus diketahui bagaimana mengakses obyek tersebut. Secara hierarki nama obyek dapat diketahui, misalnya untuk mengakses image yang pertama, terlebih dahulu harus melihat pada hirarki dimulai dari atas. Gambar 5.4 menunjukkan ilustrasi hirarki yang dibuat oleh contoh halaman HTML di atas:



Gambar 5.4 Hierarchy Element Halaman Web

Secara hirarki obyek pertama disebut dokumen. Image pertama dalam halaman itu diwakili oleh `images[0]`. Ini artinya cara mengaksesnya melalui JavaScript dengan nama `document.images[0]`. Demikian juga jika misalnya diinginkan untuk mengetahui apa yang dimasukkan user pada elemen pertama dari form, terlebih dahulu harus tahu bagaimana cara mengakses obyek yang dimaksud. Dimulai dari atas hirarki, kemudian ikuti terus jalurnya sampai menemukan obyek bernama `elements[0]` – satukan semua nama obyek yang terlewat, sehingga kumpulan obyek tersebut secara hirarki menemukan nama text field pertama yaitu: `document.forms[0].elements[0]`. Dari sebuah elemen, semua elemen memiliki sifat (property) dan fungsi (metode), misalnya untuk dapat membaca apa yang dimasukkan user dapat memakai statemen berikut ini: `name = document.forms[0].elements[0].value;`

Isinya akan disimpan dalam variabel `name` sehingga sekarang program dapat bekerja dengan variabel ini. Misalnya variabel tersebut, dipakai membuat window popup dengan `alert("Hi " + name)`. Jika input adalah 'Lina' perintah `alert("Hi " + name)` akan mengeluarkan window popup dengan text 'Hi Lina'. Jika sebuah halaman Web, memiliki halaman HTML yang besar, akan sangat merepotkan mengakses obyek obyek yang ada dengan angka-angka contohnya:

```
document.forms[3].elements[17]
document.forms[2].elements[18]
```

Agar menghindari kesulitan pada penomoran elemen, pemakaian nama yang unik untuk setiap obyek dapat diterapkan, seperti dibawah ini:

```
<form name="myForm">
Name: <input type="text" name="name" value=""><br>
...
```

Dengan cara seperti ini, akses `forms[0]` dengan `myForm`, dan `elements[0]` dengan `name` Jadi selain penulisan `name= document.forms[0].elements[0].value;` penulisan dengan nama dapat pula diterapkan: `name= document.myForm.name.value;`

Sifat (property) dari obyek JavaScript tidak hanya dibatasi pada operasi pembacaan saja, melainkan dapat juga mengisi harga baru untuk property tertentu. Contohnya penulisan string baru ke text field sebagai berikut.

```
<form name="myForm">
  <input type="text" name="input" value="bla bla bla">
  <input type="button" value="Write"
  onClick="document.myForm.input.value= 'Ya ya ya!'; ">
```

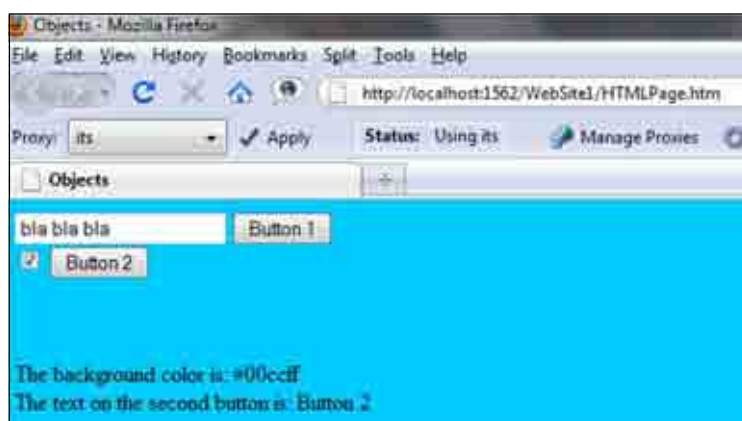
Contoh diatas, secara lengkap kode di bawah ini.

```
<html><head>
<title>Obyeks</title>
<script type="text/javascript">
function first() {
  alert("The value of the textelement is: " +
document.myForm.myText.value);}

function second() {
  var myString = "The checkbox is ";
  if (document.myForm.myCheckbox.checked) myString += "checked"
  else myString += "not checked";
  alert(myString);}
</script>
</head>
<body bgcolor="#00ccff">
<form name="myForm" action="HTMLPage.htm">
<input type="text" name="myText" value="bla bla bla"/>
<input type="button" name="button1" value="Button 1" onclick="first()"/>
<br/>
<input type="checkbox" name="myCheckbox" checked/>
<input type="button" name="button2" value="Button 2" onclick="second()"/>
</form><br/><br/>

<script type="text/javascript">
  document.write("The background color is: ");
  document.write(document.bgColor + "<br>");
  document.write("The text on the second button is: ");
  document.write(document.myForm.button2.value);
</script>
</body></html>
```

Hasil dari kode program diatas, diperlihatkan pada gambar 5.5



Gambar 5.5 Hasil program Java script

Ketika button 2 diklik, maka muncul jendela window pada fungsi alert dengan membawa nilai checkbox, sedangkan jika button 1 diklik, jendela akan muncul dengan nilai yang sama dengan yang diketikkan pada textbox.

5.8 Location-obyek Pada JavaScript

Selain jendela, obyek dokumen, pada JavaScript terdapat obyek penting yang lain yaitu: lokasi obyek. Sebuah obyek ini mewakili alamat dari dokumen HTML yang di-load. Jadi ketika halaman web di-load, misalnya halaman `http://www.yahoo.com/page.html` maka obyek `location.href` berisi sama dengan alamat ini. Bagian penting dengan obyek ini adalah pengisian alamat baru dapat diterapkan. Sebagai contoh, tombol berikut ini jika di click akan me-load halaman baru kedalam window:

```
<form>
<input type=button value="yahoo" onclick="location.href='http://www.yahoo.com'; ">
</form>
```

5.9 Frame dan JavaScript

Sebuah jendela pada browser dapat dibagi menjadi beberapa frame. Sebuah frame adalah daerah segiempat yang berada di dalam jendela browser. Setiap frame menampilkan isi dokumennya masing-masing (umumnya dokumen HTML). Dua frame dapat dibuat dalam sebuah halaman web, frame pertama me-load suatu halaman dan frame kedua me-load halaman lain. Untuk membuat frame perlu dua tag: `<frameset>` dan `<frame>`. Halaman HTML untuk membuat dua frame bisa seperti di bawah ini:

```
<html>
<frameset rows="50%,50%">
  <frame src="http://www.yahoo.com" name="frame1">
  <frame src="http://www.google.com" name="frame2">
</frameset>
</html>
```

Kode di atas akan menghasilkan dua frame. Perhatikan bahwa pemakaian property `rows` pada tag `<frameset>`. Artinya kedua frame akan tersusun dalam susunan atas-bawah. Frame atas me-load halaman HTML `http://www.yahoo.com` dan frame bawah menampilkan halaman `http://www.google.com`. Hasilnya akan terlihat pada gambar 5.6:



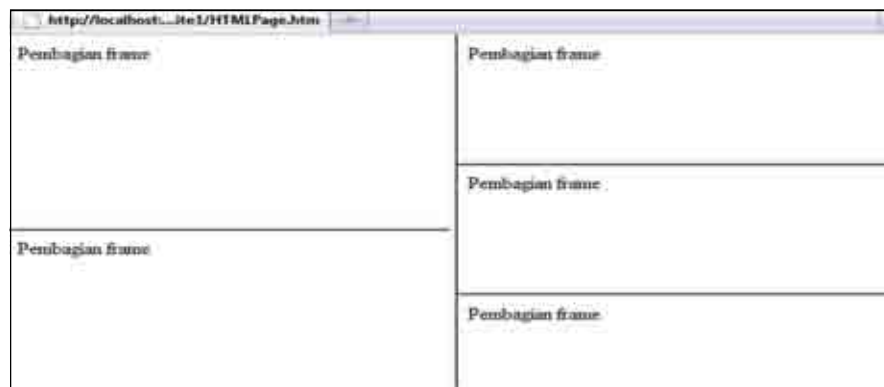
Gambar 5.6 Pembagian Frame Halaman Web

Jika hendak menyusun frame pada susunan kolom penulisan `cols` sebagai pengganti `rows` pada tag `<frameset>`. Bagian "50%,50%" menentukan seberapa besar ukuran kedua frame. Penulisan "50%," jika tidak mau menghitung berapa besar frame kedua (dengan ukuran total keduanya 100%). Penulisan juga dapat tentukan ukuran dalam satuan pixel dengan

menghilangkan tanda %. Setiap frame harus memiliki nama yang unik melalui property name pada tag <frame>. Nama ini akan memudahkan akses ke setiap frame melalui JavaScript. Pembuatan tag <frameset> yang berlapis. Berikut ini contohnya:

```
<frameset cols="50%,50%">
  <frameset rows="50%,50%">
    <frame src="cell.htm">
    <frame src="cell.htm">
  </frameset>
  <frameset rows="33%,33%,33%">
    <frame src="cell.htm">
    <frame src="cell.htm">
    <frame src="cell.htm">
  </frameset>
</frameset>
```

Hasilnya akan seperti terlihat pada gambar 5.7:

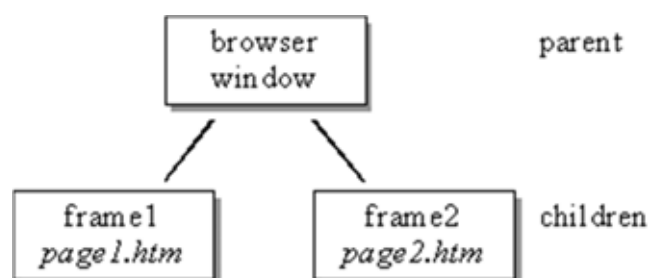


Gambar 5.7 Pembagian FrameSet Halaman Web

Penentuan ukuran border melalui property border pada tag <frameset>. Jika border=0 berarti tidak menggunakan border di antara frame

5.10 Frame dan JavaScript

JavaScript dapat 'melihat' frame pada jendela browser dengan memakai susunan semua elemen pada halaman web dalam susunan hirarki. Hal ini berlaku sama dengan frame. Gambar 5.8 memperlihatkan hirarki contoh pertama di atas:



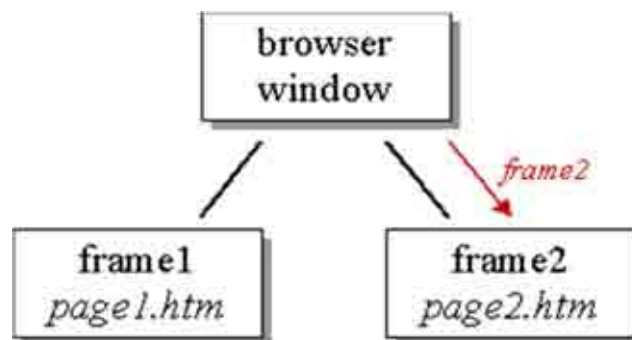
Gambar 5.8 Dua Frame

Puncak hirarki adalah *window browser*. Sebuah window ini terbagi menjadi dua frame. Window browser disebut parent pada hirarki ini dan kedua frame disebut children. Identifikasi frame dengan nama yang unik yaitu `frame1` dan `frame2`. Dengan bantuan nama ini, javascript dapat saling mempertukarkan informasi diantaranya.

Sebuah script yang dapat menyelesaikan permasalahan berikut: user men-click link pada frame pertama tapi halaman yang muncul akan di-load di frame kedua dan bukan pada frame pertama. Contohnya nanti bisa digunakan untuk menubar (atau `navigationbar`) dimana satu frame selalu tampil sama dan menampilkan beberapa link untuk navigasi sepanjang homepage. Dalam mempertukarkan informasi antar frame kita lihat tiga kasus berikut ini:

- parent window/frame mengakses child frame
- child frame mengakses parent window/frame
- child frame mengakses child frame lain

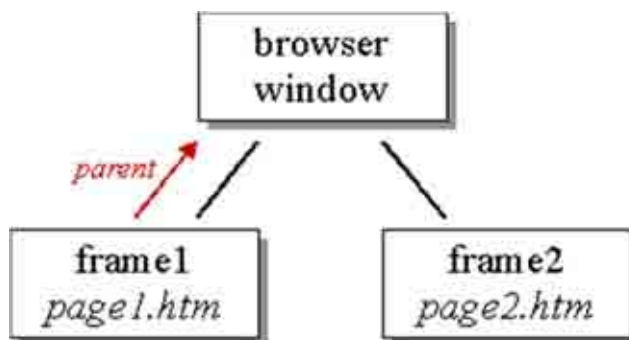
Dari sudut pandang window kedua frame disebut `frame1` dan `frame2`. Kita dapat melihat pada gambar di atas bahwa ada hubungan langsung dari parent window ke setiap frame. Jadi jika kita memiliki script pada parent window – artinya pada halaman yang membuat frame – dan kita ingin mengakses salah satu frame, kita dapat menggunakan nama frame yang dimaksud. Sebagai contoh: `frame2.document.write("Pesan ini dari parent window.");`



Gambar 5.9 Hubungan Brwswe ke Frame

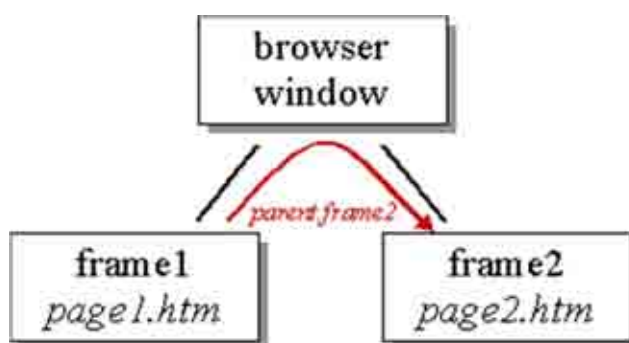
Kadang perlu juga mengakses parent window dari suatu frame. Ini diperlukan misalnya untuk menghapus frame. Menghapus frame sebenarnya cuma me-load halaman baru sebagai pengganti halaman yang menghasilkan frame. Pada kasus ini halaman yang dimaksud adalah yang berada di parent window.

Agar dapat mengakses parent window (atau parent frame) dari child frame dengan parent. Untuk me-load sebuah dokumen baru kita harus menentukan URL baru ke `location.href` pada parent window sehingga kita bisa me-load halaman baru pada parent window dengan perintah: `parent.location.href= "http://...";`



Gambar 5.10 Hubungan Frame ke Browser

Seringkali kita ingin mengakses suatu child frame dari child frame yang lain. Pada gambar di atas tidak terlihat adanya hubungan langsung antara kedua frames. Artinya kita tidak bisa langsung memanggil frame2 dari frame frame1 karena frame ini tidak mengetahui keberadaan frame kedua. Tetapi dari sudut pandang parent window, frame kedua disebut frame2 dan parent window disebut parent dilihat dari frame pertama. Jadi kita harus menulis kode berikut agar bisa mengakses document-obyek pada frame kedua dari frame pertama: `parent.frame2.document.write("Hi, ini panggilan dari frame1.");`



Gambar 5.11 Hubungan antar Frame

5.11 Predefined obyek JavaScript

Kode JavaScript telah menyediakan *predefined obyek*s yaitu obyek-obyek yang telah terdefinisi dan siap dipakai seperti *Date-obyek*, *Array-obyek* atau *Math-obyek*. Selengkapnya lihat pada referensi JavaScript.

5.11.1 Date-obyek

Obyek *Date-obyek* menyediakan fasilitas penentuan tanggal dan waktu. Misalnya dengan gampang kita bisa menghitung berapa hari lagi hari lebaran sejak hari ini, atau juga bisa menampilkan jam pada halaman HTML. Mari kita mulai dengan menampilkan waktu. Kita harus membuat *Date-obyek* yang baru dengan menggunakan operator *new*. Lihat pada baris kode berikut: `today= new Date()`

Kode ini akan membuat sebuah *Date-obyek* baru bernama *today*. Jika kita tidak menentukan hari dan waktu tertentu sebagai argumennya berarti tanggal dan waktu sekarang yang akan digunakan. Artinya setelah mengeksekusi `today= new Date()`, *today* akan mewakili

tanggal dan waktu sekarang. Sebuah Date-obyek menyediakan method yang bisa kita gunakan melalui obyek today. Method-method ini contohnya adalah `getHours()`, `setHours()`, `getMinutes()`, `setMinutes()`, `getMonth()`, `setMonth()` dan seterusnya (lihat pada dokumentasi Netscapes JavaScript). Perhatikan bahwa Date-obyek hanya mewakili tanggal dan waktu tertentu dan bukan seperti jam yang berubah setiap detik secara otomatis. Untuk memperoleh tanggal dan waktu yang lain selain tanggal sekarang, kita bisa menggunakan kode berikut ini: `today= new Date(1997, 0, 1, 17, 35, 23)`. Ini akan membuat Date-obyek baru yang berisi tanggal 1 January 1997 pada jam 17:35 dan 23 detik. Jadi kita bisa menentukan tanggal dan hari dengan format berikut: `Date(year, month, day, hours, minutes, seconds)`

Perhatikan bahwa kita menggunakan angka 0 untuk January. Angka 1 untuk February dan seterusnya. Sekarang kita akan membuat script yang mengeluarkan tanggal dan waktu sekarang dengan hasil seperti ini:

```
Time: 10:8  
Date: 1/24/2010
```

Baris kodenya adalah sebagai berikut:

```
<script language="JavaScript">  
<!--hide  
now = new Date();  
document.write("Time: " + now.getHours() + ":" +  
    now.getMinutes()+"<br>");  
document.write("Date: " + (now.getMonth() + 1) + "/" +  
    now.getDate() + "/" + (1900 + now.getYear()));  
// --></script>
```

Di sini gunakan method-method seperti `getHours()` untuk menampilkan waktu dan tanggal yang terdapat pada Date-obyek `now`. Perhatikan bahwa kita menambahkan 1900 pada tahun, karena method `getYear()` mengeluarkan angka tahun sejak 1900. Artinya untuk tahun 1997 hasilnya akan berupa angka 97 dan jika tahun 2010 hasilnya akan 110 – bukan 10! Jika kita menambahkan 1900 kita tidak akan mendapatkan permasalahan untuk tahun 2000-an.

Perhatikan juga penambahan satu pada angka yang diperoleh dari `getMonth()` karena alasan yang sama seperti sebelumnya. Script ini tidak mengecek apakah angka menit kurang dari 10 artinya kita mungkin akan memperoleh waktu dengan tampilan seperti ini: 14:3 yang sebenarnya adalah 14:03. Kita selesaikan masalah ini pada script berikutnya. Sekarang kita akan membuat script yang menampilkan sebuah jam yang benar-benar bekerja: Kodenya adalah sebagai berikut:

```
<html> <head>  
<script Language="JavaScript"><!-- hide  
var timeStr, dateStr;  
  
function clock() {  
    now= new Date();  
    hours= now.getHours();  
    minutes= now.getMinutes();  
    seconds= now.getSeconds();  
    timeStr= "" + hours;  
    timeStr+= ((minutes < 10) ? ":0" : ":") + minutes;  
    timeStr+= ((seconds < 10) ? ":0" : ":") + seconds;  
    document.clock.time.value = timeStr;  
  
    date= now.getDate();  
    month= now.getMonth()+1;  
    year= now.getYear();
```



```

    dateStr= "" + month;
    dateStr+= ((date < 10) ? "/0" : "/" ) + date;
    dateStr+= "/" + year;
    document.clock.date.value = dateStr;

    Timer= setTimeout("clock()",1000);
}

// --></script>
</head>
<body onLoad="clock()">
<form name="clock">
  Time: <input type="text" name="time" size="8" value=""><br>
  Date: <input type="text" name="date" size="8" value="">
</form>
</body>
</html>

```

Pada program diatas method `setTimeout()` dipakai untuk men-set waktu dan tanggal setiap detiknya, jadi harus membuat *Date-obyek* baru setiap detik. Perhatikan bahwa kita memanggil *function clock()* pada event-handler `onLoad` di tag `<body>`. Pada bagian body halaman HTML kita memiliki dua text-element. Function `clock()` menulis waktu dan tanggal pada kedua text-element ini dengan format yang benar. Kita menggunakan dua string `timeStr` dan `dateStr` untuk keperluan ini. Permasalahan menit yang kurang dari 10 kita selesaikan pada script di atas melalui baris kode berikut: `timeStr+= ((minutes < 10) ? ":0" : ":") + minutes;`

Di sini angka menit ditambahkan dengan string `timeStr`. Jika menit kurang dari 10 kita harus menambah angka 0. Kode di atas sebenarnya sama maksudnya dengan :

```

if (minutes < 10) timeStr+= ":0" + minutes
else timeStr+= ":" + minutes;

```

5.11.2 Array-obyek

Sebuah array merupakan suatu hal yang sangat penting diketahui. Misalkan diinginkan untuk menyimpan 100 nama yang berbeda. Bagaimana melakukannya dengan JavaScript? Hal ini dapat dilakukan dengan mendefinisikan 100 variabel dan mengisinya dengan sederetan nama-nama. Cara ini cukup rumit dan tidak efisien. Array bisa dilihat sebagai banyak variabel yang disatukan. Cara mengaksesnya melalui satu nama dan penomoran. Misalnya array kita bernama `names`. Maka cara mengakses nama pertama melalui `names[0]`, nama kedua dengan `names[1]` dan seterusnya. Setelah JavaScript 1.1 (Netscape Navigator 3.0) hal ini dapat menggunakan Array-obyek. Perintah untuk membuat array baru dengan perintah `myArray = new Array()`. Kemudian bisa mengisi harganya dengan:

```

myArray[0]= 17;
myArray[1]= "Stefan";
myArray[2]= "Koch";

```

Sebuah array pada JavaScript sangat flexible. Penentuan ukuran tidak harus ditentukan terlebih dahulu karena ukurannya akan ditentukan secara dinamis. Jika kita tulis `myArray[99]= "xyz"` ukuran array akan menjadi 100 elemen (array JavaScript hanya bisa bertambah bukan berkurang, jadi buat array sekecil mungkin.). Tidak jadi masalah apakah menyimpan angka, string atau obyek lain dalam array. Contoh dibawah ini adalah keluaran dari yang diinginkan adalah seperti:

```
first element  
second element  
third element
```

Baris kodenya adalah:

```
<script language="JavaScript">  
<!-- hide  
  
myArray= new Array();  
  
myArray[0]= "first element";  
myArray[1]= "second element";  
myArray[2]= "third element";  
  
for (var i= 0; i< 3; i++) {  
    document.write(myArray[i] + "<br>");  
}  
  
// --></script>
```

Pada kode di atas, pertama kali buat array dengan nama myArray. Lalu diisi dengan tiga harga yang berbeda. Setelah itu mulai sebuah loop. Perintah loop ini menjalankan perintah `document.write(myArray[i]+"
");` sebanyak tiga kali. Variabel `i` menghitung dari 0 sampai 2 dengan for-loop. Penggunaan `myArray[i]` di dalam for-loop. Karena `i` menghitung dari 0 sampai 2, maka diperoleh tiga kali pemanggilan `document.write()`. Kita bisa menulis loop di atas seperti ini:

```
document.write(myArray[0] + "<br>");  
document.write(myArray[1] + "<br>");  
document.write(myArray[2] + "<br>");
```

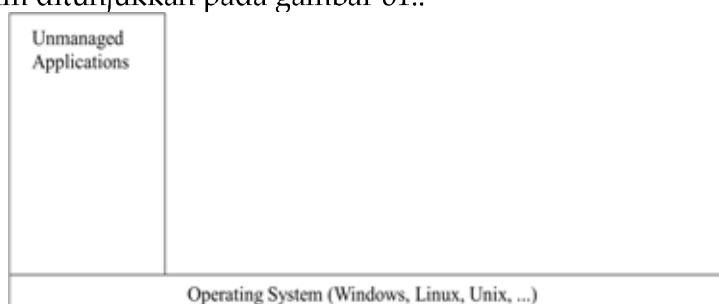
Bab 6

Pengenalan ASP.NET

ASP.NET yang telah berkembang dari versi sebelumnya, yaitu ASP, kemudian ASP.NET 1.x. Microsoft memperkenalkan ASP.NET (yang dipanggil sebagai ASP+), dengan di iringi sebuah Framework. Perkenalan ini diluncurkan pada bulan juli,2000.Pemrograman .NET adalah pengembangan baru pada kerangka kerja (framework) yang menyediakan sebuah interface terhadap aplikasi program melalui sebuah service (layanan) dan kumpulan API.

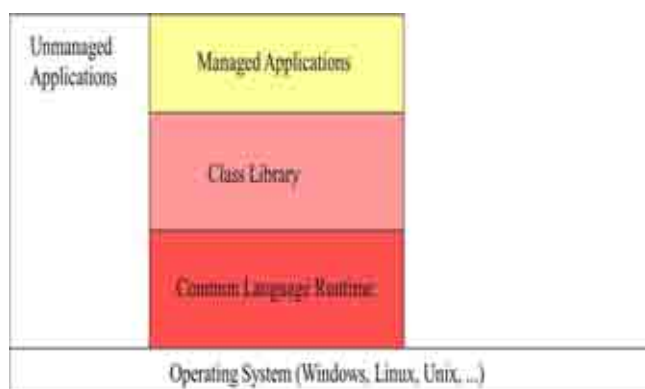
6.1 Sekilas tentang .NET

Pemrograman .NET adalah platform perangkat lunak yang dikembang untuk aplikasi desktop dan aplikasi web. Era .Net adalah sebuah pengaturan pemrograman yang baru dengan memandang program program sebelum sebagai program yang tidak ter-manage dengan baik, sebagai gambaran umum ditunjukkan pada gambar 6.1.:



Gambar 6.1 Arsitektur Aplikasi lama

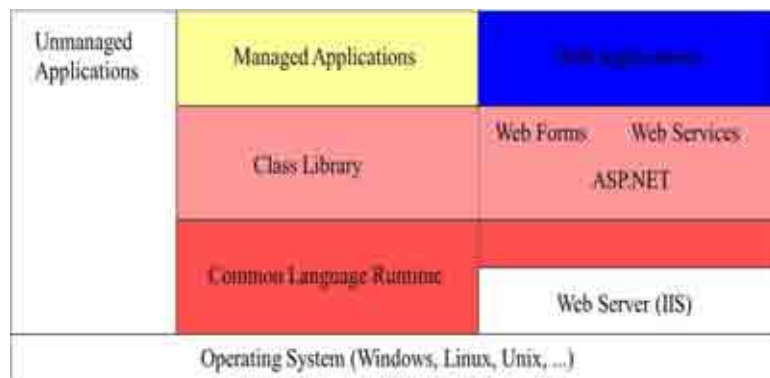
Cara pandang baru terhadap pemrograman dengan mengelompokkan kedalam CLR dan pustaka kelas. Hal ini tergambarakan seperti pada gambar 6.2:



Gambar 6.2 Arsitektur Aplikasi BerbasisPustaka

- **Common Language Runtime:** Sebuah mesin yang berjalan diatas sistem operasi yang bertujuan untuk mengatur aplikasi agar dapat dipakai secara umum terdiri dari interoperability, security, garbage collection, versioning, ...
- **Class Library:** Sebuah pustaka kelas yang dapat dipakai oleh aplikasi terdiri dari GUI, collections, threads, networking, reflection, XML, ...

Cara pandang pemrograman yang baru, mengikuti perkembangan teknologi dengan lahirnya teknologi berbasis web. Peranan ASP.NET terletak pada program berbasis web yang dapat digambarkan pada gambar 6.3:



Gambar 6.3 Arsitektur Aplikasi ASP.NET

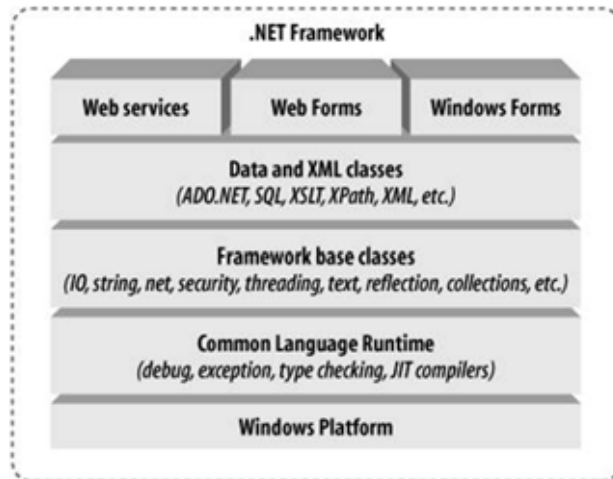
Program aplikasi ASP.NET terbagi menjadi dua jenis yaitu:

- **ASP.NET, Web Forms:** Program yang mengakses web pada web server, yang biasa disebut sebagai Web GUI (object-oriented, event-based, browser-independent)
- **Web Services:** Layanan berbasis web yang biasanya dipakai untuk pertukaran data untuk menunjang *distributed services* melalui RPC (SOAP, HTTP)

6.2 Framework .NET 4.0

Framework .NET merupakan sebuah jembatan dari aplikasi ke system operasi, dimana sebuah framework .NET berisi:

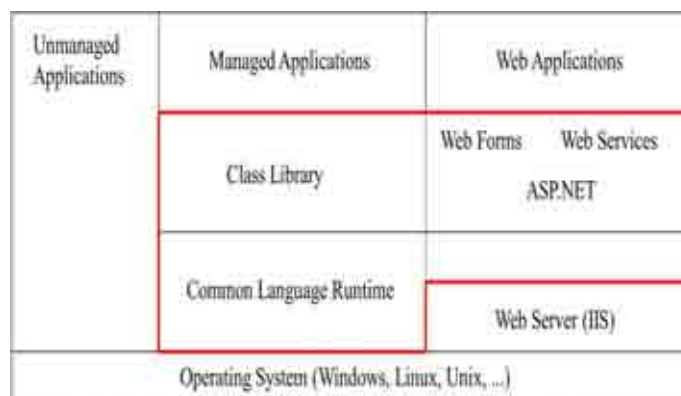
- Sebuah compiler untuk lima bahasa (C#, Visual Basic, Managed C++, J#, and the Jscript scripting language)
- Sejumlah kelas yang terhubung yang dikenal sebagai FCL (Framework Class Library), sebagai dukungan untuk akses kepada windows, aplikasi web, data akses, web service dan sebagainya
- Common Language Runtime (CRL), sebuah mesin object oriented yang bertugas untuk mentranslasi kode yang dihasilkan oleh compiler bahasa ke dalam bentuk kode native yang dipakai untuk mengeksekusi sebuah aplikasi



Gambar 6.4 Arsitektur framework .NET

Peranan dari sebuah CRL, mengeksekusi program pada web server. CRL mengaktifasi obyek dan melakukan pengecekan aktivasi terhadap keamanan, mengatur memori dan mengeksekusi program tersebut dan menhandel persoalan garbage yang ditimbulkan oleh program yang dijalankan. Pada gambar 6.4. diatas, sebuah arsitektur framework .NET yang berbasiskan pada kelas-kelas, di-ikuti dengan tambahan lapisan dari data dan kelas XML, beserta lapisan selanjutnya sebagai web service, web site dan windows form. Kesemua itu terkumpul ke dalam FCL, sampai saat ini terdapat 5.000 kelas, dimana FCL merupakan sebuah rapid development yang dipakai dalam aplikasi ASP.NET.

Sebuah framework .NET mendukung CLS (Common Language Spesification), yang mengijinkan pada pemilihan sebuah sintak penulisan yang paling nyaman. Pemilihan bahasa dapat dipakai tanpa ada keterbatasan dengan masing-masing bahasa dapat terhubung dengan bahasa lainnya. Garis besar dari framework dibandingkan dengan aplikasi aplikasi sebelumnya, dapat digambarkan pada gambar dibawah ini:



Gambar 6.5 Garis besar framework

Pengembangan framework didukung oleh tool, server dan service. Hal ini dapat dipilih diantaranya adalah:

- + **Tools** (Visual Studio .NET, ildasm, gacutil, ...)
- + **Servers** (SQL Server, BizTalk Server, ExchangeServer, ...)
- + **Services** (My Service, Passport Service, ...)

6.3 Tujuan ASP.NET

Pembuatan program dengan basis ASP.NET bertujuan untuk mensekukan model dari program desktop dengan program web. Hal ini adalah perkembangan dari program sebelumnya, dimana program masih terpisah antara desktop dan web seperti yang ditunjukkan pada gambar 6.6:

Desktop programming	Web programming
object-oriented	ASP (not object-oriented)
compiled (C/C++, Fortran, ...)	interpreted (VBScript, Javascript, PHP, ...)
extensive class libraries	specialized libraries

Gambar 6.6 Program desktop dan Web versi lama

Hadirnya ASP.NET sebagai kelanjutan dari ASP adalah menggabungkan kemampuan desktop yang dapat diterapkan pada program berbasis web, seperti yang ditunjukkan pada gambar 6.7

Desktop and Web programming
object-oriented (even ASP.NET)
compiled (C#, C++, VB.NET, Fortran, ...)
uniform class library

Gambar 6.7 Program desktop dan Web versi ASP.NET

Selain bertujuan mensekukan program, tujuan lain dari program ASP.NET adalah *Interoperability* diantara bahasa program yang berbeda-beda. Sebagai gambaran dari program yang sebelumnya adalah pengembangan program terdahulu biasanya:

- millions of lines of code in C++, Fortran, Visual Basic, ...
- very limited interoperability

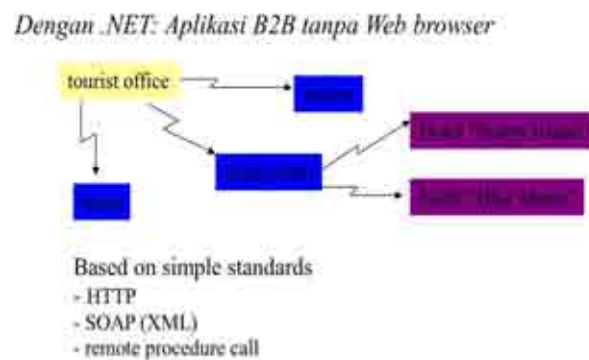
Program ASP.NET memiliki kompatibilitas secara biner diantara lebih dari 20 bahasa pemrograman, languages (C#, C++, VB.NET, Java, Eiffel, Fortran, Cobol, ML, Haskell, Pascal, Oberon, Perl, Python, ...) . Contoh kompatibilitas dari program ASP.NET ditunjukkan pada gambar 6.8:

<pre>Public Class A Public x As Integer Public Sub Foo() ... End Class</pre>	<pre>class B: A public string ... public void Bar() ...</pre>	<pre>class Client feature obj: B; ... create obj; obj.Bar; ... end</pre>
--	---	--

Gambar 6.8 Kompatibilitas ASP.NET

6.4 Web Services

Tujuan ASP.NET yang telah disebutkan pada subab 6.2, dikembangkan lebih jauh lagi, tidak hanya sekedar memenuhi kebutuhan web server. Kebutuhan lain dari perkembangan teknologi informasi adalah kemampuan B2B (*business to business*), dimana sebagai pendukungnya adalah web service. Berbeda dengan web server (menyajikan data ke pemakai) ataupun email, web service dipakai untuk menyediakan layanan transaksi yang terdistribusi yang digambarkan pada gambar 6.8. Sebuah kantor pariwisata terhubung dengan layanan pesawat terbang, jaringan hotel dan pihak perbankan. Keterhubungan ini didasarkan pada teknologi web service, masing-masing layanan menyediakan service yang dapat diakses.



Gambar 6.9 Aplikasi B2B

6.5 Halaman Awal

Pengembangan ASP.NET sampai saat ini telah memasuki pada versi Visual Studio versi 2010, sesaat halaman awal dari Visual Studio 2010 terbuka, maka pilihan pertama adalah isian pilihan lingkungan yang akan dipakai, seperti terlihat pada gambar 6.10:



Gambar 6.10 Pilihan Lingkungan Kerja Visual Studio 2010