

Bahan Ajar Pemrograman Web

Disusun Oleh:

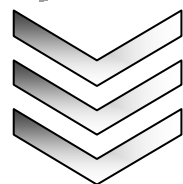
Aris Puji Widodo, M.T.

Adi Wibowo, S.Si

Mahasiswa Ilmu Komputer Angkatan 2004

Tim Modul Himakom Undip 2007-2008

WWW-HTML-CSS-JS-DOM-DHTML-XML



Program Studi Ilmu Komputer
Jurusan Matematika Fakultas MIPA
Universitas Diponegoro
2007

Revisi 3.0

6 Oktober 2007

DAFTAR ISI

Cover.....	i
Daftar isi.....	ii
Bab 1 Internet Overview	
1.1. What is Internet.....	1
1.2. Internet Service.....	1
1.3. Website & Web Application.....	1
1.4. Web Building Blocks.....	2
1.5. How Does Web Work.....	2
1.6. Web Client.....	3
1.7. Web Server.....	3
1.8. URL (Uniform Resource Locator).....	3
1.9. HyperText Transfer Protocol	4
1.10. HyperText Markup Language	4
1.11. Cascading Style Sheet.....	5
1.12. Web programming.....	5
Bab 2 How Internet Works	
2.1. Halaman WWW.....	6
2.2. HTML Code	6
2.3. WWW	6
2.4. URL.....	7
2.5. HTTP Session	7
2.6. HTTP Protocol.....	7
2.7. Connections	8
2.8. Addresses.....	8
2.9. TCP	8
2.10. Reliable TCP Operation	8
2.11. UDP.....	9
2.12. Internet Protocol.....	9
2.13. Ethernet.....	9
2.14. Routing	10
2.15. DNS	11
2.16. NameSpace Sructure	11
2.17. Network Model.....	12
2.18. Conclusion	12
Bab 3 HyperText Markup Language	
3.1. Pendahuluan	13
3.2. Struktur Dasar.....	13
3.3. Spesifikasi Simbol dan Elemen Tag	14
3.4. Background Colors & Images	14
3.5. Bold, Italic, Underlining, dan Nesting tags	15
3.6. Monospaced Font, Font Size, dan Font Face	15
3.7. Font Color, more about Nesting Tags.....	15
3.8. Section Headings, Link Colors	16
3.9. Paragraph Tag dan Paragraph Aligning, Centering	16
3.10. The "space code", dan Special Characters.....	16
3.11. Inserting Images, dan Relative Paths	17

3.12.	Link, Email Links, dan Image Links	18
3.13.	More on Images, Image Filesizes, Thumbnails, Anchors.....	18
3.14.	Ordered dan Unordered Lists	20
3.15.	Definition Lists.....	20
3.16.	Horizontal Rules.....	20
3.17.	Preformatting	22
3.18.	Tables.....	22
3.19.	Forms	26
3.20.	Frames	27
Bab 4	Cascading Style Sheet	
4.1.	Internal CSS	29
4.2.	Eksternal CSS.....	30
4.3.	Inline CSS	30
4.4.	CSS Classes	31
4.5.	CSS Background	32
4.6.	CSS Font	35
4.7.	CSS Text	38
4.8.	CSS Padding	41
4.9.	CSS Margin.....	42
4.10.	CSS Border	43
4.11.	CSS List	46
4.12.	CSS Link	48
4.13.	CSS Mouse Cursor.....	51
Bab 5	JavaScript	
5.1.	Sintaks	53
5.2.	Head Body	53
5.3.	File Eksternal.....	53
5.4.	Operator	54
5.5.	Variabel	55
5.6.	Functions.....	56
5.7.	Events	56
5.8.	Alerts.....	56
5.9.	Confirm.....	57
5.10.	Prompt.....	57
5.11.	Print.....	58
5.12.	Redirect	58
5.13.	Popup	59
5.14.	Date.....	60
5.15.	Array	62
5.16.	Image.....	62
5.17.	Case Study	64
Bab 6	Document Object Model	
6.1.	DOM dan Cara Modifikasinya dengan Javascript.....	68
6.2.	Membuat DOM Widget secara Dinamis.....	69
6.3.	Menggeser DOM Widget.....	71

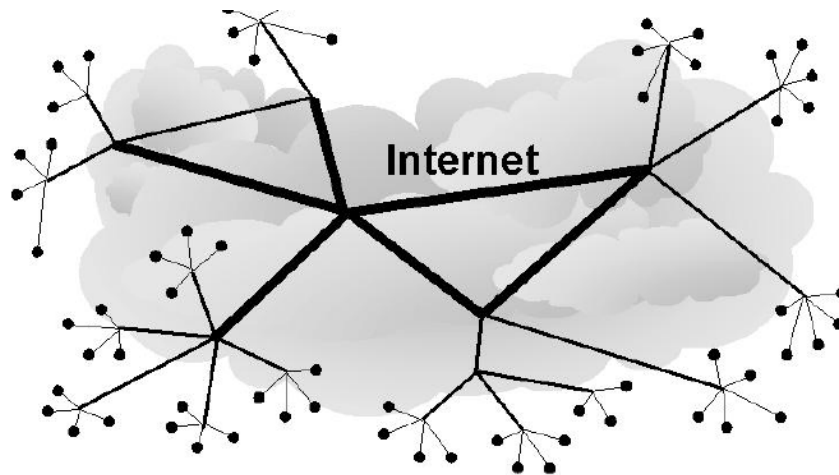
Bab 7	Dynamic HyperText Markup Language	
7.1.	Pengantar	74
7.2.	DHTML sebagai Perpaduan Teknologi Web.....	75
Bab 8	eXtensible Markup Language	
8.1.	Pengertian	77
8.2.	Tentang XML	77
8.3.	Penggunaan XML.....	78
8.4.	Sintaks XML	79
Lampiran HTML		83

BAB 1

Internet Overview

1.1. What is Internet?

- a large computer network (all over the world)
- network of networks
- based on Transmission Control Protocol/Internet Protocol (TCP/IP)



Gambar 1.1. Internet Network

1.2. Internet Services

- World Wide Web (WWW) : Hyper Text Transfer Protocol (HTTP)
- e-mail : Post Office Protocol (POP), Simple Mail Transfer Protocol (SMTP), Internet Message Access Protocol (IMAP)
- chat : Internet Relay Chat (IRC)
- file transfer : File Transfer Protocol (FTP)
- remote access : telnet, Shell Secure (SSH)
- Lightweight Directory Access Protocol (LDAP), Network management Application Protocol (NMAP), SOAP, Voice Over Internet Protocol (VoIP), etc.

1.3. Web Site & Web Application

- Web (WWW)
 - a collection of documents and files on the internet that are linked together, enabling users to find, retrieve, and use the information
- Web site / homepage
 - static pages
 - information based (users gather information)
- Web application
 - dynamic pages
 - task based (users do something)
- Types of web site/web application
 - personal/corporate web site
 - e-commerce, e-learning, e-government
 - news/information/community portal
 - corporate portal, etc.

1.4. Web Building Blocks

- Web client (browser)
- Web server
- URL : Uniform Resource Locator
- HTTP : HyperText Transfer Protocol
- HTML : HyperText Markup Language
- CSS : Cascading Style Sheet

and for web programming, there're more :

- CGI, server side program execution (Perl, C)
- server side scripting (PHP, ASP)
- client side scripting (JavaScript, JScript, VBScript)

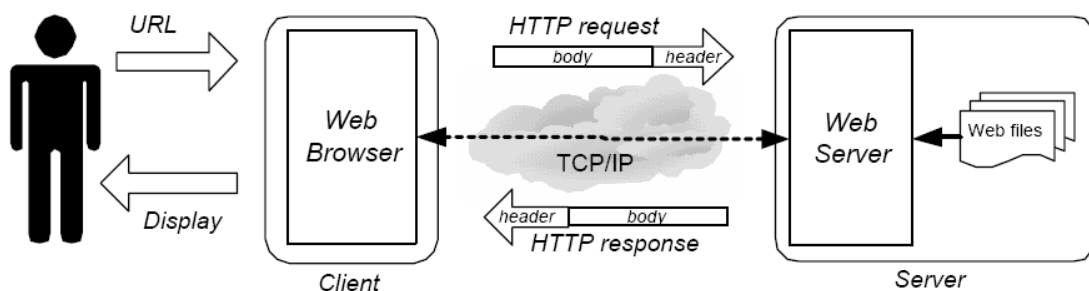
1.5. How does Web Work?

What happened :

- user types URL into browser
http://www.google.com
 - browser translates URL to an IP address and connects using TCP to server on port 80 (HTTP service)
connects to a computer with IP address 216.239.39.99 port 80
 - upon connection, browser sends a GET request (HTTP request)
GET / HTTP/1.1
 - server responds with HTTP response (with headers and the file requested in the body). The mime type is also included which tells browser what type of file it is (HTML, text, image, etc).
- ```
Content-type: text/html
<html>
<head><title> Google </title></head>
<body>Welcome to google </body>
</html>
```
- inline files (like images) don't come with the main file, so browser has to re-submit a new request to get every other file listed in the page.  
GET /logo.gif HTTP/1.0
  - browser renders received files to be displayed to user
  - with HTTP, once the file is done downloading, the connection broken, and the server forgets we exist.

HTTP is **stateless** :

No continuing interaction between a user and a server.  
Each request is discrete.



**Gambar 1.2.** HTTP Request and Response

### **1.6. Web Client (browser)**

- web browser
  - is a software
  - runs on user's computer
  - a tool to navigate the web
  - display web documents
- technically,
  - it constructs HTTP requests required to retrieve a web document (addressed with URL)
  - renders the responses into user's view
- most popular web browser software todate:
  - MS Internet Explorer, runs on Windows platform
  - Netscape Navigator/Communicator, runs on multi platform
  - Opera, runs on multi platform
  - Konqueror, runs on Linux platform
  - Mozilla , runs on multi platform
  - lynx (text based), runs on Linux platform

### **1.7. Web server**

- web server
  - is a software
  - runs on a server
  - makes web documents stored on the server become available to internet users
- technically,
  - it handles HTTP requests
  - get the requested document
  - sent it back as HTTP responses
- most popular web server software todate:
  - Apache, runs on multi platform (best on Linux)
  - MS Internet Information Server (IIS), runs on Windows platform
  - Tomcat (for java), runs on multi platform

### **1.8. URL (Uniform Resource Locator)**

- URL is the addressing system used in internet
- standard URL format is described in RFC 1738 (<http://www.ietf.org/rfc/rfc1738.txt>)
- addresses begin with the type of service/protocol
- examples of URL address:
  - <http://students.if.itb.ac.id/~if21008>
  - <mailto:if21008@student.if.itb.ac.id>
  - <ftp://ftp.informatika.org/>
- URL address used in web:
- addresses begin with http://, followed by the domain name and directory information
- examples of web URL address:
  - <http://www.itb.ac.id/campus-life/index.html>
  - <http://www.google.com/search?hl=en&q=URL+RFC>
  - <http://www.indymedia.org:8081/>

## 1.9. HyperText Transfer Protocol (HTTP)

- HTTP is communication protocol used in web
- standard HTTP specification (HTTP 1.1) is described in RFC 2616 (<http://www.ietf.org/rfc/rfc2616.txt>)
- example of "conversation":

## Browser

```
GET / HTTP/1.1
Accept: image/gif, image/jpeg, application/x-shockwave-flash, */*
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
```

## Server

```
Content-type: text/html
<html>
<head><title> Google </title></head>
<body>Welcome to google </body>
</html>
```

## Browser

```
GET /logo.gif HTTP/1.1
Accept: image/gif, image/jpeg, application/x-shockwave-flash, */*
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
```

## Server

Content-type: **image/gif**

GIF89a€6÷aD&n^+\_!VRŠ:¾è \_gÔ5ÈßŽ7J\_ %o ¥Ø£" !Y ¤\_K\_  
"d'qDbF9~""âκμ-È  
\_ö} &•%= æâz! ÛÑ- uébecòux'~ M%%"š\* ÉæOE,Ö~i\_ ÉI\$^„.....

## 1.10. HyperText Markup Language (HTML)

- HTML is a standard format for publishing web documents
- standard HTML specification (HTML 4.01) can be found at <http://www.w3.org/TR/html4/>
- example of HTML document:

```
<html>
<head>
<title>My first HTML document</title>
</head>
<body>
<p>Hello world!
Welcome to my first HTML page.
</p>
</body>
</html>
```

- display result :

Hello world!  
Welcome to my **first** HTML page.



### 1.11. CSS (Cascading Style Sheet)

- CSS is a simple mechanism for adding style (e.g. fonts, colors, spacing) to web documents
- standard CSS specification (CSS 2) can be found at <http://www.w3.org/TR/REC-CSS2/>
- example of HTML document using CSS:

```
<html>
<head>
<title>My first HTML document</title>
</head>
<style type='text/css'>
 p { font-family: Arial; text-decoration: underline; }
 p:first-letter { font-size: 20pt; }
 b { font-size: 20pt; }
</style>
<body>
<p>Hello world!
Welcome to my first HTML page.
</p>
</body>
</html>
```

- display result :

```
Hello world!
Welcome to my first HTML page.
```

### 1.12. Web programming

#### CGI, server side program execution (Perl, C)

Web server executes a program file and take its output as HTTP response

#### Server side scripting (PHP, ASP)

Web server parse and executes script embedded inside the web document

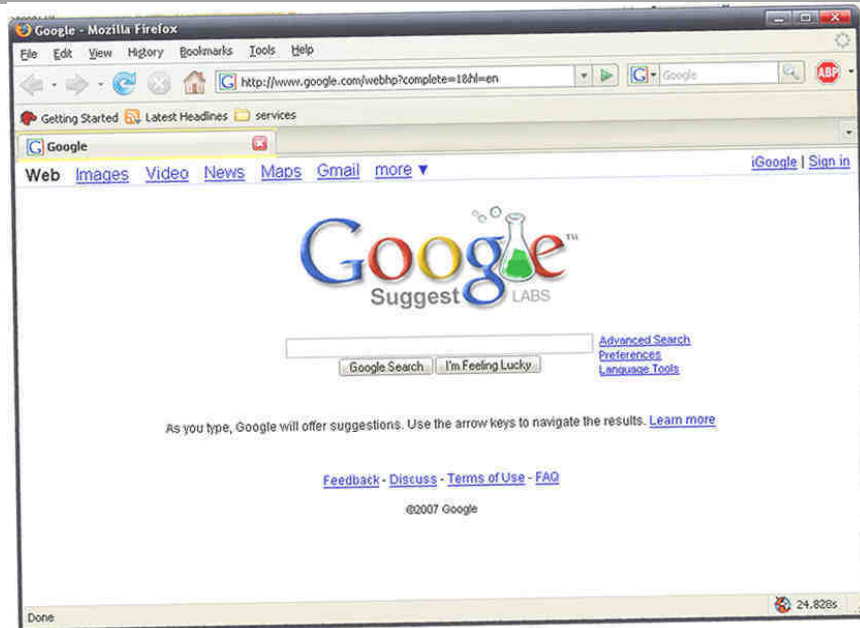
#### Client side scripting (JavaScript, JScript, VBScript)

Web browser parse and executes script embedded inside the received web document, and may update the display without re-submit a request to web server

## BAB 2

### How Internet Works

#### 2.1. Halaman World Wide Web (WWW)



**Gambar 2.1.** Halaman WWW

#### 2.2. HyperText Markup Language (HTML) Code

```
<HTML>
<HEAD>
<TITLE>Nixu International</TITLE>
</HEAD>
<BODY background="/gfx/back1.gif">
<CENTER>

<H1><I>Nixu Oy</I></H1>

Contact us
Jobs
What we do

Namesurfer

</CENTER>
</BODY> </HTML>
```

#### 2.3. WWW

- Combines text, pictures and hyperlinks to a hypertext document
  - a hyperlink points to another document
- Can also contain moving pictures, active programs etc.
- Hypertext document is written in HTML language
  - HyperText Markup Language
  - based on SGML (Standard Generalized Markup Language)
  - describes the structure of a document
- WWW browser decides how to display the document based on the structure description in HTML

## 2.4. URL

- Universal Resource Locator (URL)
- URL is an address to the current location of information
  - protocol://host[:server port]/path/file.html**
  - protocol://host[:server port]/path/**
- For example
  - <http://www.hut.fi/>
  - <http://www.nixu.fi/~kiravuo/etiketti/index.html>
  - <news:sfnet.harrastus.retkeily>
  - <ftp://ftp.funet.fi/pub/netinfo/rfc/>
- WWW browser uses the URL address to retrieve a document over the network

## 2.5. Hyper Text Transfer Protocol (HTTP) Session

- Connection to the HTTP server program in www.nixu.fi
  - equivalent to: telnet www.nixu.fi 80

```
GET /~kiravuo/demo.html HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/4.06 [en] (win95; I)
Accept: image/gif, image/x-xbitmap, image/jpeg,
image/pjpeg, image/png, */*
Accept-Charset: iso-8859-1,*,utf-8
...

HTTP/1.1 200 OK
Server: Apache/1.2.6
Last-Modified: Tue, 24 Nov 1998 06:35:48 GMT
Content-Length: 466
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive

Content-Type: text/html
...
<HTML>
<HEAD>
<TITLE>Nixu International</TITLE>
...

GET /gfx/back1.gif HTTP/1.0
Referer: http://www.nixu.fi/~kiravuo/demo.html
...

HTTP/1.1 200 OK
Content-Length: 902
Content-Type: image/gif
..GIF89a...
...
```

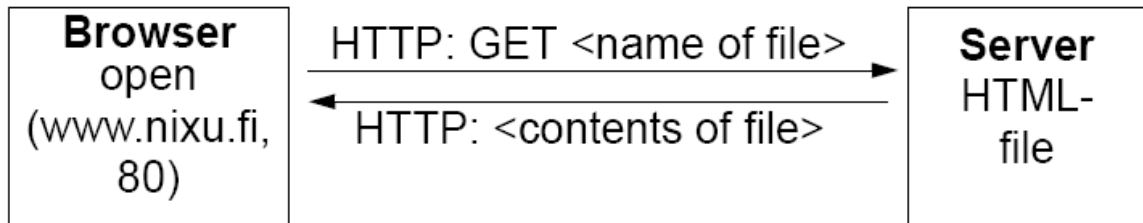
**Gambar 2.2. HTTP Session**

## 2.6. HTTP protocol

- HTTP is a simple protocol for file transfer over the network
- Client (browser) initiates the session by connecting to the server
  - pull protocol, server does not transmit on its own
- Client issues the GET-command to retrieve a file from the server
- Additional information is also passed
  - client type
  - acceptable data formats
- Server sends information about the file and the contents of the file

## 2.7. Connections

- HTML is a data format
- HTTP protocol transfers files over a reliable connection
  - no error checks or checksums in HTTP
- On the Internet we use TCP/IP for the reliable connection
  - TCP/IP provides a reliable byte-stream as a service to HTTP
- Browser uses HTTP to get the required HTML file specified by the URL



**Gambar 2.3.** WWW Connection

## 2.8. Addresses

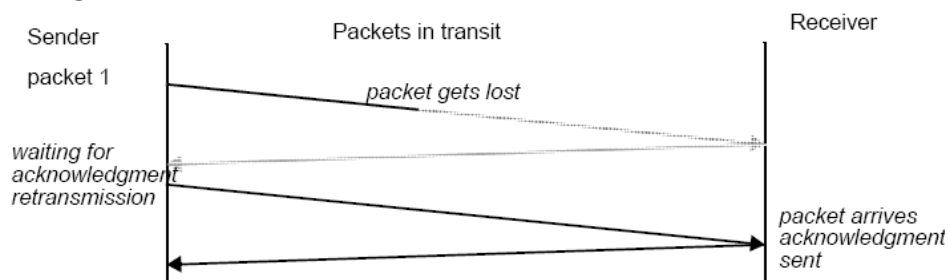
- Each host has a name
  - eg. www.hut.fi
- The network uses numeric addresses
  - eg. 130.233.224.28
- Translation is provided by the Domain Name Service (DNS)
- A host has also numbered ports, which are connected to server programs
  - port addresses are standardized, eg port 80 for HTTP, port 25 for Simple Mail Transfer Protocol (SMTP) email

## 2.9. TCP

- TCP = Transmission Control Protocol
- Defined in RFC-793
- Connection-oriented, reliable, **byte-stream** service
- Application data is broken into **segments**, which are sent as IP datagram.
- Features:
  - checksums, timeouts and flow control
  - segment reassembly in correct order, discarding duplicate packets

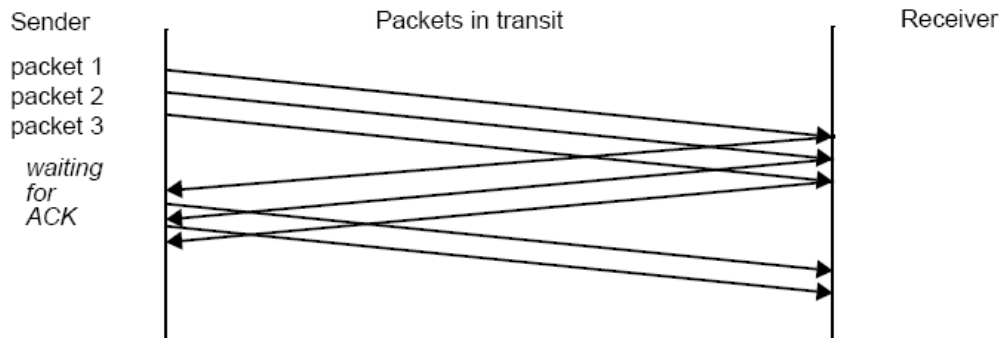
## 2.10. Reliable TCP operation

- Receiver sends acknowledgment for each segment.
- If a packet gets lost, timeout will ensure it's retransmitted



**Gambar 2.4.** TCP Operation

- Normally a sliding window technique



**Gambar 2.5.** Sliding Window Technique

### 2.11. UDP

- UDP = User Datagram Protocol
- Defined in RFC-768
- Unreliable datagram-oriented transportation layer protocol
  - offers little extra functionality besides port numbers
  - light-weight, easy to implement
- Applications using UDP: DNS, Simple Network Time Protocol (SNTP), Simple Network Management Protocol (SNMP)

### 2.12. Internet Protocol (IP)

- IP = The Internet Protocol
- Defined in RFC-791
- IP sends simple **datagrams** over network.
- It provides **unreliable** and **connectionless** delivery service.
  - unreliable = no guarantees, best effort only
  - connectionless = each packet is routed separately

### 2.13. Ethernet

- Defined in IEEE 802.3
- Local area network
  - limited to short range connections (100 m - 1 km)
  - different cabling options (coaxial cabling, twisted pair etc.)

#### 2.13.1. Data client -> server

```
ETHER: Destination = 8:0:20:74:f1:2c, Sun
ETHER: Source = 0:0:3b:80:e:93,
ETHER: Ethertype = 0800 (IP)
IP: Protocol = 6 (TCP)
IP: Source address = 194.197.118.74, chili.nixu.fi
IP: Dest. address = 194.197.118.20, jalopeno.nixu.fi
TCP: Source port = 35620
TCP: Destination port = 80 (HTTP)
TCP: Sequence number = 760000273
TCP: Acknowledgement number = 2370000258
HTTP: "GET /~kiravuo/demo.html HTTP/1.0
 Connection: Keep-Alive
 User-Agent: Mozilla/4.06..."
```

**2.13.2. Ack server -> client**

```

ETHER: Destination = 0:0:3b:80:e:93,
ETHER: Source = 8:0:20:74:f1:2c, Sun
ETHER: Ethertype = 0800 (IP)
IP: Flags = 0x4 (do not fragment)
IP: Protocol = 6 (TCP)
IP: Source address = 194.197.118.20, jalopeno.nixu.fi
IP: Dest. address = 194.197.118.74, chili.nixu.fi
TCP: Source port = 80 (HTTP)
TCP: Destination port = 35620
TCP: Sequence number = 2370000258
TCP: Acknowledgement number = 760000379
TCP: Flags = 0x10 (ACK)
HTTP: ""

```

**2.13.3. Data server -> client**

```

ETHER: Destination = 0:0:3b:80:e:93,
ETHER: Source = 8:0:20:74:f1:2c, Sun
ETHER: Ethertype = 0800 (IP)
IP: Protocol = 6 (TCP)
IP: Source address = 194.197.118.20, jalopeno.nixu.fi
IP: Dest. address = 194.197.118.74, chili.nixu.fi
TCP: Source port = 80 (HTTP)
TCP: Destination port = 35620
TCP: Sequence number = 2370000299
TCP: Acknowledgement number = 760000379
TCP: Data offset = 20 bytes
TCP: Flags = 0x10 (ACK)
HTTP: "HTTP/1.1 200 OK
 Server: Apache/1.2.6
 Last-Modified: Tue, 2..."

```

**2.13.4. Ack client -> server**

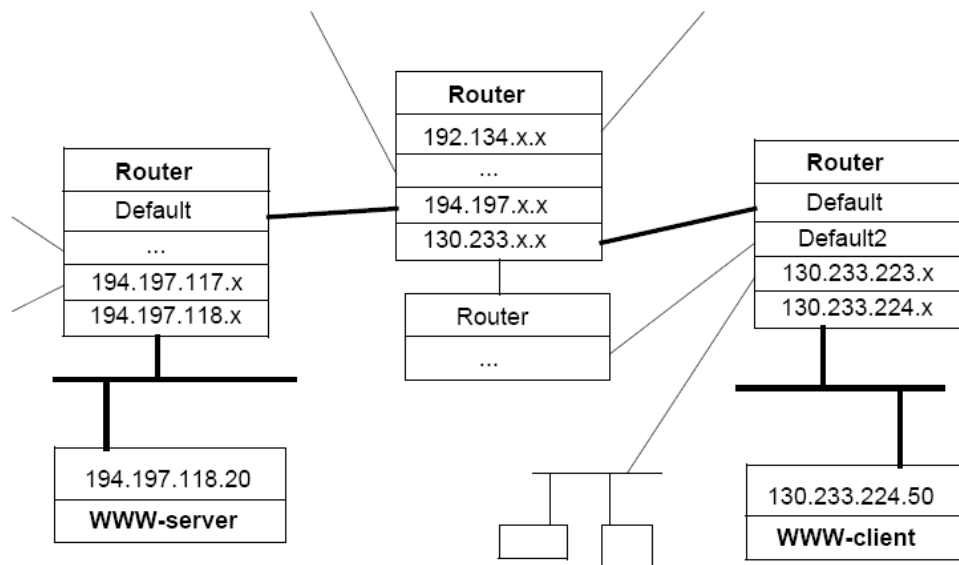
```

ETHER: Destination = 8:0:20:74:f1:2c, Sun
ETHER: Source = 0:0:3b:80:e:93,
ETHER: Ethertype = 0800 (IP)
IP: Protocol = 6 (TCP)
IP: Source address = 194.197.118.74, chili.nixu.fi
IP: Dest. address = 194.197.118.20, jalopeno.nixu.fi
TCP: Source port = 35620
TCP: Destination port = 80 (HTTP)
TCP: Sequence number = 760000379
TCP: Acknowledgement number = 2370000627
TCP: Flags = 0x10 (ACK)
HTTP: ""

```

**2.14. Routing**

- Local Area Network (LAN)s are connected by routers
- Each router has a table that tells which network address can be reached by which connection
- Backbone routers update their information automatically and can route around breaks in the network
- Routing example

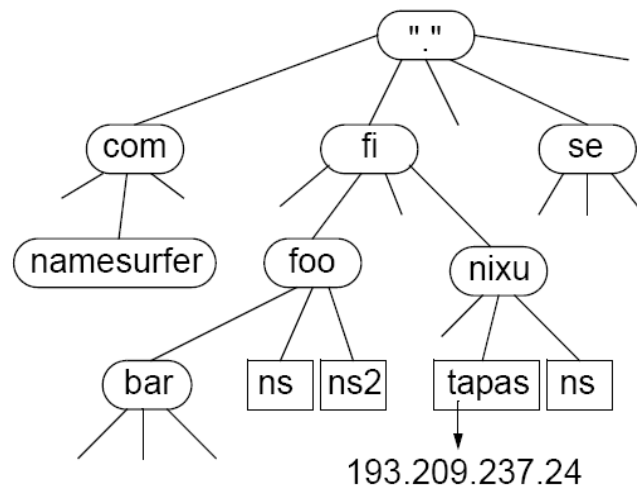


**Gambar 2.6.** Routing

### 2.15. Domain Name System

- Distributed database with local caching
- Distributed administration (adding or deleting data)
- Services provided:
  - Hostname to IP address
  - IP address to hostname
  - Mail exchange records

### 2.16. Name Space Structure

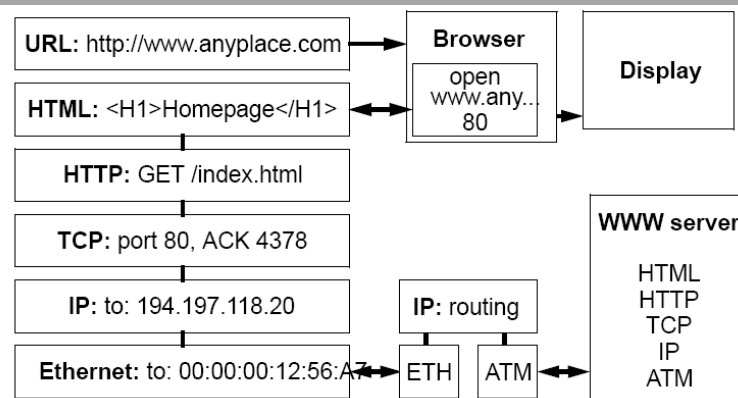


**Gambar 2.7.** Names Space Structure

## 2.17. Network models

TCP/IP	OSI	UColorado	UKUUG
		Political	Confusion
		Financial	Financial
Application	Application	Application	Application
	Presentation	Presentation	
	Session	Session	
TCP, UDP	Transport	Transport	Transport
IP	Network	Network	IP
Data link	Data link	Data link	Network
Physical	Physical	Physical	

## 2.18. Conclusion



**Gambar 2.8.** Conclusion



## BAB 3

### *Hyper Text Markup Language (HTML)*

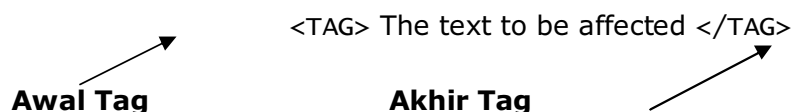
#### 3.1. Pendahuluan

HTML merupakan format standar yang digunakan untuk membuat dokumen web page. HTML merupakan bahasa bertanda, menggunakan serangkaian teks tertentu (Tag) untuk memberi tanda teks yang memiliki interpretasi khusus, dan file HTML berbentuk file teks (plain text file) bukan binary file.

Sekarang ini aplikasi yang berkembang banyak cenderung menggunakan dokumen web page, dengan tujuan untuk memberikan kemudahan instalasi, dan akses yang tidak tergantung pada waktu serta tempat (baik secara intranet atau internet). Untuk mendukung kondisi ini dokumen web page di buat dengan menggunakan HTML yang memiliki ekstensi file html atau htm. HTML dapat diakses menggunakan browser, seperti *Internet Explorer, Netscape Navigator, Mozilla Firefox, Opera*, dan browser-browser lainnya. HTML dibuat dengan menggunakan editor text, seperti *Notepad, EditPlus, Notepad++*, dan editor lainnya.

Versi HTML yang dikembangkan adalah HTML 1.0, HTML 2.0, HTML 3.2, dan HTML 4.01 yang merupakan versi terakhir (pada saat modul ini di buat).

Dokumen HTML selalu memiliki pasangan mark, yang disebut sebagai TAG untuk memberikan efek yang diinginkan. Secara umum format TAG adalah sebagai berikut :

  
**Awal Tag**                      **Akhir Tag**

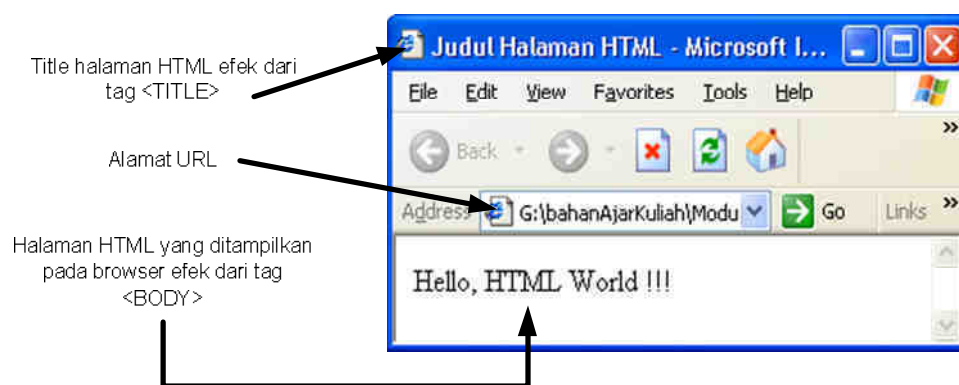
Tag HTML kebanyakan tidak bersifat case sensitive, hanya beberapa tag saja yang bersifat case sensitive.

#### 3.2. Struktur Dasar Halaman HTML

Setiap halaman HTML selalu dimulai dengan tag `<HTML>` dan diakhiri dengan tag `</HTML>`, digunakan oleh browser untuk melakukan pembacaan file yang terdiri dokumen HTML. Tag berikutnya adalah untuk membagi dokumen menjadi dua sesion head dan body. Sesion head adalah sebagai pengantar suatu halaman, yang didefinisikan dengan pasangan tag `<HEAD>` dan `</HEAD>` yang diletakan pada di bawah tag `<HTML>`. Sesion body adalah sebagai tempat untuk memasukkan text dan property lainnya yang akan tampil pada halaman, yang didefinisikan dengan pasangan tag `<BODY>` dan `</BODY>`.

Terdapat kebutuhan untuk memberikan judul suatu halaman yang akan tampil pada menu bar browser, dimana dengan menambahkan pasangan tag `<TITLE>` dan `</TITLE>` yang diletakan di bawah tag `<HEAD>`. Text dan property lainnya yang akan ditampilkan pada browser diletakan dibawah tag `<BODY>`. Untuk struktur dasar halaman HTML adalah sebagai berikut (tampilan pada browser diberikan pada gambar 3.1):

<b>Nama File : halaman.html</b>
<pre> &lt;!-- Tanda Komentor --&gt; &lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"&gt; &lt;!-- HTML versi 4.0 yang digenerate --&gt; &lt;!-- menggunakan editPlus 2.10 c --&gt;  &lt;HTML&gt; &lt;HEAD&gt; &lt;!-- Ditampilkan sebagai title pada menu bar browser--&gt; &lt;TITLE&gt; Judul Halaman HTML &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;!-- Ditampilkan pada browser --&gt; &lt;BODY&gt;     Hello, HTML world !!! &lt;/BODY&gt; &lt;/HTML&gt; </pre>



Gambar 3.1. Tampilan pada Browser

### 3.3. Spesifikasi Simbol dan Elemen Tag HTML

HTML yang digunakan adalah HTML versi 4.01 yang dikenali oleh browser yang ada pada sekarang ini baik produk Netscape maupun Microsoft. Pada bagian ini hanya merupakan referensi singkat bukan merupakan spesifikasi lengkap HTML, yang diberikan pada bagian **Lampiran Bare Bones HTML Guide**. Untuk detilnya dapat di lihat pada halaman *World Wide Web Consortium (W3C)* dengan alamat <http://www.w3.org/Markup>, untuk *step-by-step HTML Guide* pada <http://webach.com/web/wwwhelp.html>, <http://home.netscape.com>, dan <http://www.microsoft.com>

### 3.4. Background Colors dan Images

<b>Nama File : backColor.html</b>
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Background Colors &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;!-- Memberikan warna pada background --&gt; &lt;BODY BGCOLOR="#CCFFCC"&gt;     Hello, HTML world !!! &lt;/BODY&gt; &lt;/HTML&gt; </pre>
<b>Nama File : backImage.html</b>
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Background Colors &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;!-- Memberikan background menggunakan image --&gt; </pre>

<b>Nama File : backColor.html</b>
<pre> &lt;!-- dengan syarat file image harus ada (*.gif, *.bmp, *.jpg, etc) -- &gt; &lt;BODY BACKGROUND="swirlies.gif"&gt;     Hello, HTML world !!! &lt;/BODY&gt; &lt;/HTML&gt; </pre>
<b>Nama File : backFixed.html</b>
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Background Colors &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;!-- BGPROPERTIES=FIXED memberikan efek background dalam --&gt; &lt;!-- tetap dan yang berjalan isi pada &lt;BODY&gt; --&gt; &lt;BODY BACKGROUND="images\qpbut1.gif" BGPROPERTIES=FIXED&gt; &lt;!-- images\qpbut1.gif adalah file qpbut1.gif --&gt; &lt;!-- pada folder images --&gt;     Hello, HTML world !!! &lt;/BODY&gt; &lt;/HTML&gt; </pre>

### 3.5. Bold, Italic, Underlining, dan Nesting tags

<b>Nama File : formatChar.html</b>
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; bold, italic, underlining, nesting tags &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; &lt;!-- nesting tags yang tidak terjadi overlap (good) --&gt;     &lt;B&gt;&lt;U&gt;Hello&lt;/U&gt;,&lt;I&gt;HTML world &lt;/I&gt; !!!&lt;/B&gt; &lt;!-- Terjadi overlap &lt;B&gt; &lt;I&gt;...&lt;/B&gt;&lt;/I&gt; (bad)--&gt; &lt;/BODY&gt; &lt;/HTML&gt; </pre>

### 3.6. Monospaced Font, Font Size, dan Font Face

<b>Nama File : fontFormat.html</b>
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Font color, more about nesting tags &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; &lt;!-- set ukuran font sebesar dari 1 - 7 FONT SIZE--&gt; &lt;!-- sekaligus memberikan set jenis font FONT FACE --&gt;     Hello, HTML &lt;FONT SIZE=1 FACE="Comic Sans MS"&gt;world !!!&lt;/FONT&gt;     &lt;BR&gt;     Hello, HTML &lt;FONT SIZE=2 FACE="Arial"&gt;world !!!&lt;/FONT&gt; &lt;BR&gt;     Hello, HTML &lt;FONT SIZE=3 FACE="Arial Black"&gt;world !!!&lt;/FONT&gt;&lt;BR&gt;     Hello, HTML &lt;FONT SIZE=4 FACE="Courier New"&gt;world !!!&lt;/FONT&gt;&lt;BR&gt;     Hello, HTML &lt;FONT SIZE=5 FACE="Impact"&gt;world !!!&lt;/FONT&gt;&lt;BR&gt;     Hello, HTML &lt;FONT SIZE=6 FACE="Verdana"&gt;world !!!&lt;/FONT&gt;&lt;BR&gt;     Hello, HTML &lt;FONT SIZE=7 FACE="Georgia"&gt;world !!!&lt;/FONT&gt;&lt;BR&gt; &lt;/BODY&gt; &lt;/HTML&gt; </pre>

### 3.7. Font Color, more about Nesting Tags

<b>Nama File : fontColor.html</b>
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Font color, more about nesting tags &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; </pre>

**Nama File : fontColor.html**

```

<!-- set warna font FONT COLOR -->
<!-- sekaligus memberi format font secara neting Tag -->
Hello,

 HTML <U>World</U> !!!

</BODY>
</HTML>

```

**3.8. Section Headings, Link Colors****Nama File : headlinkCol.html**

```

<HTML>
<HEAD>
<TITLE> Section headings, link colors </TITLE>
</HEAD>
<!-- Mendefinisikan warna background, text, dan link -->
<BODY BGCOLOR="#000000" TEXT="#FFFFFF66" LINK="#00FF66" VLINK="#00BB33"
ALINK="#00CCFF">
<!-- Membuat halaman link-->
Halaman Link
<!-- Heading H1 sampai H6 -->
 <H1>Hello, HTML World !!!</H1>

<!-- Heading menggunakan atribut ALIGN -->
 <H2 ALIGN="left">Hello, HTML World !!!</H2>

 <H3 ALIGN="center">Hello, HTML World !!!</H3>

 <H4 ALIGN="right">Hello, HTML World !!!</H4>

 <H5>Hello, HTML World !!!</H5>

 <H6>Hello, HTML World !!!</H6>

<!-- Memberikan efek roller coaster -->
You can even make a rollercoaster!
</BODY>
</HTML>

```

**3.9. Paragraph Tag dan Paragraph Aligning, Centering****Nama File : paragraph.html**

```

<HTML>
<HEAD>
<TITLE> Paragraph tag & paragraph aligning, centering </TITLE>
</HEAD>
<!-- Mendefinisikan warna background, text, dan link -->
<BODY>
<!-- Membuat paragraph <P> </P> -->
 <P>Hello, HTML World !!!</P>
<!-- Membuat paragraph <P> tanpa </P> perhatikan hasilnya -->
 <P>Hello, HTML World !!!
<!-- Paragraph menggunakan atribut ALIGN -->
 <P ALIGN="left">Hello, HTML World !!!</P>
 <P ALIGN="center">Hello, HTML World !!!</P>
 <P ALIGN="right">Hello, HTML World !!!</P>
<!-- Memberikan efek center menggunakan tag <CENTRE> -->
 <CENTRE>
 <H5>Hello, HTML world !!!</H5>
 <CENTRE>
</BODY>
</HTML>

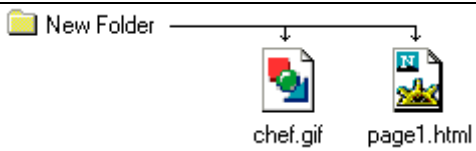
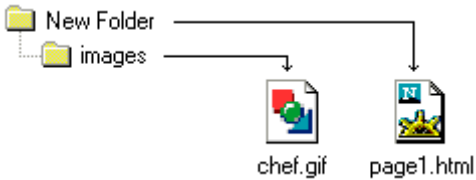
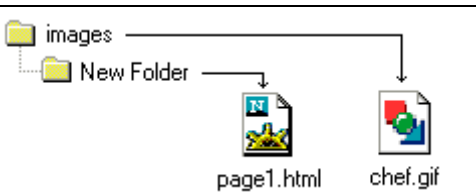
```

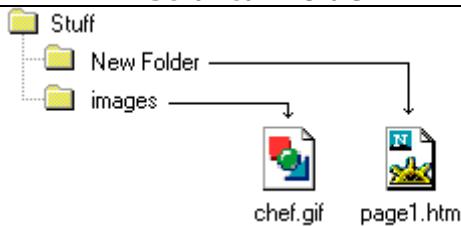
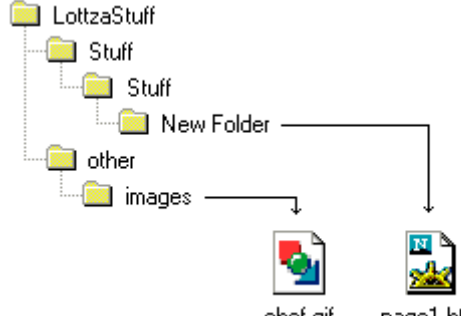
### 3.10. The "space code", dan Special Characters

Nama File : specialChar.html
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; The "space code", special characters &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; &lt;!-- Membuat spasi &amp;nbsp; --&gt; &lt;P&gt;Hello,&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;HTML world !!!&lt;/P&gt; &lt;!-- Membuat paragraph &lt;P&gt; tanpa &lt;/P&gt; perhatikan hasilnya --&gt; &lt;P&gt;Hello, HTML world !!! &lt;!-- Membuat quotation mark &amp;quot; --&gt; &lt;P ALIGN="left"&gt;&amp;quot;Hello, HTML World !!!&amp;quot;&lt;/P&gt; &lt;!-- Membuat &lt; less-than symbol &amp;lt; --&gt; &lt;!-- Membuat &gt; greater-than symbol &amp;gt; --&gt; &lt;!-- Membuat &amp; ampersand &amp;amp; --&gt; &lt;P ALIGN="center"&gt;&amp;lt;Hello&amp;gt;, HTML&amp;amp; world !!!&lt;/P&gt; &lt;/BODY&gt; &lt;/HTML&gt; </pre>

### 3.11. Inserting Images, dan Relative Paths

Nama File : imagePath.html
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Inserting images, relative paths &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; &lt;!-- Menampilkan images dari file chef.gif --&gt; &lt;!-- file chef.gif berada pada satu folder images --&gt; &lt;P ALIGN="center"&gt;&lt;!-- posisi center --&gt; &lt;IMG SRC="images\chef.gif" WIDTH=130 HEIGHT=101&gt; &lt;/BODY&gt; &lt;/HTML&gt; </pre>

Struktur Folder	Aturan Pemanggilan
	SRC="chef.gif", berarti file gambar berada satu folder dengan file html
	SRC="images/chef.gif", berarti file gambar berada pada folder anak dari file html berada. Dalam hal ini file gambar pada folder images
	SRC="../chef.gif", berarti file gambar berada di folder yang diatas folder letak file html

Struktur Folder	Aturan Pemanggilan
	SRC="../../images/chef.gif", berarti file gambar dan file html terpisah folder namun pada tingkat yang sama
	SRC="../../../other/images/chef.gif", sebaiknya penempatan semacam ini dihindari

### 3.12. Link, Email Links, dan Image Links

Nama File : link.html
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Link, Email Links, dan Image Links &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; &lt;!-- Membuat halaman Link --&gt; &lt;A HREF="http://www.netCool.net"&gt;&lt;H4&gt;Halaman Link&lt;/H4&gt;&lt;/A&gt;  &lt;!-- Link ke email sekaligus melakukan inisialisasi --&gt; &lt;!-- nilai subject email --&gt; &lt;P&gt;     Send me &lt;A HREF="mailto:masarisdong@yahoo.com?subject=Hello"&gt;         E-Mail     &lt;/A&gt; &lt;/P&gt;  &lt;!-- Membuat halaman Link menggunakan images link --&gt; &lt;P ALIGN="center"&gt;&lt;!-- posisi center --&gt; Go to &lt;A HREF="http://www.netCool.net"&gt;     &lt;IMG SRC="images\chef.gif" WIDTH=130 HEIGHT=101&gt; &lt;/A&gt;&lt;/P&gt; &lt;!-- Membuat Link ke format file --&gt; &lt;A HREF="cover.doc"&gt;Download my resume&lt;/A&gt;&lt;BR&gt; &lt;!-- Membuat Link dengan membuka windows baru --&gt; Go to &lt;A HREF="http://home.netscape.com/" TARGET="_blank"&gt;Netscape!&lt;/A&gt; &lt;/BODY&gt; &lt;/HTML&gt; </pre>

### 3.13. More on Images, Image Filesizes, Thumbnails, Anchors

Nama File : moreImage.html
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; more on images, image filesizes, thumbnails, anchors &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; </pre>

**Nama File : moreImage.html**

```
<!-- membuat halaman link dg images yg memiliki -->
<!-- little blue dashes sbg empty space -->
<!-- karena ganti baris akibat enter-->
<!-- dan diberikan tool tip text atribut ALT -->
<CENTER>

<IMG SRC="images\myhome.gif"
 WIDTH="66" HEIGHT="23"
 BORDER="0" ALT="Home">

<IMG SRC="images\myemail.gif"
 WIDTH="66" HEIGHT="23"
 BORDER="0" ALT="Email">

<IMG SRC="images\mylinks.gif"
 WIDTH="66" HEIGHT="23"
 BORDER="0" ALT="Links">

</CENTER>
</BODY>
</HTML>
```

**Nama File : moreImage1.html**

```
<HTML>
<HEAD>
<TITLE> more on images, image filesizes, thumbnails, anchors </TITLE>
</HEAD>
<BODY>
<!-- membuat halaman link dg images tanpa memiliki -->
<!-- little blue dashes sbg empty space -->
<!-- karena diletakan dlm satu baris tanpa enter-->
<!-- dan diberikan tool tip text atribut ALT -->
<CENTER>
<IMG
 SRC="images\myhome.gif"
 WIDTH="66" HEIGHT="23"
 BORDER="0" ALT="Home">
<IMG
 SRC="images\myemail.gif"
 WIDTH="66" HEIGHT="23"
 BORDER="0" ALT="Email">
<IMG
 SRC="images\mylinks.gif"
 WIDTH="66" HEIGHT="23"
 BORDER="0" ALT="Links">
</CENTER>
</BODY>
</HTML>
```

**Nama File : moreImage2.html**

```
<HTML>
<HEAD>
<TITLE> more on images, image filesizes, thumbnails, anchors </TITLE>
</HEAD>
<BODY>
<!-- membuat halaman link dg images tanpa memiliki -->
<!-- empty space karena <A> diletakan dlm -->
<!-- satu baris sebelumnya tanpa enter -->
<!-- dan diberikan tool tip text atribut ALT -->
```

**Nama File : moreImage1.html**

```

<CENTER>
<IMG
 SRC="images\myhome.gif"
 WIDTH="66" HEIGHT="23"
 BORDER="0" ALT="Home"><IMG
 SRC="images\myemail.gif"
 WIDTH="66" HEIGHT="23"
 BORDER="0" ALT="Email"><IMG
 SRC="images\mylinks.gif"
 WIDTH="66" HEIGHT="23"
 BORDER="0" ALT="Links">
</CENTER>
</BODY>
</HTML>

```

**3.14. Ordered dan Unordered Lists****Nama File : orderedUnList.html**

```

<HTML>
<HEAD>
<TITLE> ordered & unordered lists </TITLE>
</HEAD>
<BODY>
Materi Pemrograman Internet
<!-- Add a pair of unordered list tags -->

 <!-- Add a list item -->
 Introduction
 How Internet work?
 HTML

<!-- Add a pair of ordered list tags -->

 <!-- Add a list item -->
 JavaScript
 CSS
 PHP

</BODY>
</HTML>

```

**3.15. Definition Lists****Nama File : defineList.html**

```

<HTML>
<HEAD>
<TITLE> Definition lists </TITLE>
</HEAD>
<BODY>
<DL> <!-- Definition List -->
<!-- Definition Title -->
<DT>10th Amendment
<!-- Definition Item -->
<DD>The powers not delegated to the United States by the
 Constitution, nor prohibited by it to the States, are
 reserved to the States respectively, or to the people.
</DL>

```



Nama File : defineList.html
</BODY> </HTML>

### 3.16. Horizontal Rules

Nama File : horizontalRule.html
<pre>&lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Horizontal rules &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; Horizontal Rules &lt;HR&gt;&lt;!-- Horizontal Rules --&gt;  &lt;!-- Horizontal Rules dg few options available --&gt; &lt;BR&gt;Horizontal Rules dg few options available &lt;HR WIDTH=20%&gt; &lt;HR WIDTH=50%&gt; &lt;HR WIDTH=100%&gt; &lt;HR WIDTH=20&gt; &lt;HR WIDTH=50&gt; &lt;HR WIDTH=100&gt;  &lt;!-- Horizontal Rules dg atribut ALIGN --&gt; &lt;BR&gt;Horizontal Rules dg atribut ALIGN &lt;HR WIDTH=60% ALIGN="left"&gt; &lt;HR WIDTH=60% ALIGN="right"&gt; &lt;HR WIDTH=60% ALIGN="center"&gt;  &lt;!-- Horizontal Rules dg control the thickness of the line --&gt; &lt;BR&gt;Horizontal Rules dg control the thickness of the line &lt;HR WIDTH=60% SIZE=1&gt; &lt;HR WIDTH=60% SIZE=3&gt; &lt;HR WIDTH=60% SIZE=8&gt; &lt;HR WIDTH=60% SIZE=15&gt;  &lt;!-- Horizontal Rules dg make it a solid line --&gt; &lt;BR&gt;Horizontal Rules dg make it a solid line &lt;HR WIDTH=60% SIZE=1 NOSHADE&gt; &lt;HR WIDTH=60% SIZE=3 NOSHADE&gt; &lt;HR WIDTH=60% SIZE=8 NOSHADE&gt; &lt;HR WIDTH=60% SIZE=15 NOSHADE&gt; &lt;/BODY&gt; &lt;/HTML&gt;</pre>

### 3.17. Preformatting

```

Nama File : preFormat.html

<HTML>
<HEAD>
<TITLE> Preformatting </TITLE>
</HEAD>
<BODY>
<!-- Menampilkan pada browser sesuai dengan format -->
<!-- pada editor plain text -->
<PRE>
 ///\ _
 (@ @)
+---oOO---()-----+
| ARIS PUJI WIDODO |
| For |
| President |
+---oOO---+
 | | | |
 || ||
 oOO Ooo

</PRE>
</BODY>
</HTML>

```

### 3.18. Tables

Pada dasarnya terdapat 3 tag yang digunakan untuk membuat table pada dokumen web, yaitu :

1. <TABLE>

Merupakan tag utama yang digunakan untuk mendefinisikan tabel dengan beberapa atribut seperti size, border width, dan sebagainya.

2. <TR>

TableRow mendefinisikan baris tabel secara horisontal yang kemudian dibagi-bagi secara vertikal membentuk cell oleh <TD>

3. <TD>

TableData menyatakan cell tunggal.

<--This--	---is---	----a---	---Table--	---Row-->
			cell	
	cell			cell

**Gambar 3.2.** Ilustrasi Table pada Dokumen Web

```

Nama File : table.html

<HTML>
<HEAD>
<TITLE> Membuat Table </TITLE>
</HEAD>
<BODY>
<TABLE> <!-- Membuat table nilai BORDER=0 (default)-->
 <TR> <!-- Setiap table minimal memiliki 1 baris <TR> -->
 <!-- Setiap baris minimal memiliki 1 cell <TD> -->
 <TD>My First Cell</TD>
 </TR>
</TABLE>
</BODY>
</HTML>

```

**Nama File : table1.html**

```
<HTML>
<HEAD>
<TITLE> Membuat Table </TITLE>
</HEAD>
<BODY>
<!-- Tabel pertama -->
<!-- Membuat table dg atribut BORDER=3 dan WIDTH=100%-->
<!-- WIDTH=100% : satu layar penuh lebarnya -->
Table Pertama
<TABLE BORDER=3 WIDTH=100%>
 <!-- baris pertama -->
 <TR>
 <!-- baris pertama Cell pertama -->
 <!-- <TD> dg atribut ALIGN dan VALIGN-->
 <TD ALIGN="center" VALIGN="middle">
 First Row First Cell
 </TD>
 <!-- baris pertama Cell kedua -->
 <TD ALIGN="left" VALIGN="top">
 First Row Second Cell
 </TD>
 </TR>
 <!-- baris kedua -->
 <TR>
 <!-- baris kedua Cell pertama -->
 <TD ALIGN="right" VALIGN="bottom">
 Second Row First Cell
 </TD>
 <!-- baris kedua Cell kedua -->
 <TD ALIGN="center" VALIGN="top">
 Second Row Second Cell
 </TD>
 </TR>
</TABLE>

<!-- Tabel kedua -->
Table Kedua
<TABLE BORDER=2 WIDTH=25% HEIGHT=50%>
<TR>
 <TD ALIGN="center" VALIGN="middle">
 <!-- Cell di isi dengan images -->
 <IMG SRC="images\titlepic08.gif" WIDTH=50% HEIGHT=50%
 ALT="titlePic08">
 </TD>
</TR>
</TABLE>

<!-- Tabel ketiga -->
Table Ketiga
<TABLE BORDER=1 WIDTH=300 HEIGHT=75 >
<TR>
 <!-- <TD> dg atribut WIDTH -->
 <TD WIDTH=60%>Aris</TD>
 <TD WIDTH=20%>Shinta</TD>
 <TD WIDTH=20%>Rick</TD>
</TR>
<TR>
 <TD>Renata</TD>
 <TD>Curly</TD>
 <!-- : 1 kali spasi -->
 <TD> </TD>
```

**Nama File : table1.html**

```
</TR>
</TABLE>
</BODY>
</HTML>
```

**Nama File : table2.html**

```
<HTML>
<HEAD>
<TITLE> Membuat Table </TITLE>
</HEAD>
<BODY>
<!-- CELLPADDING is the amount of space between -->
<!-- the border of the cell and the contents of the cell.-->
<!-- nilai CELLSPACING dan CELLPADDING silahkan anda -->
<!-- lakukan perubahan dan amati perubahannya -->
<TABLE BORDER=3 CELLSPACING=12 CELLPADDING=12 >
<TR>
 <!-- <TD> dg atribut WIDTH -->
 <TD WIDTH=60%>Aris</TD>
 <TD WIDTH=20%>Shinta</TD>
 <TD WIDTH=20%>Rick</TD>
</TR>
<TR>
 <TD>Renata</TD>
 <TD>Curly</TD>
 <TD>Mira</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

**Nama File : table3.html**

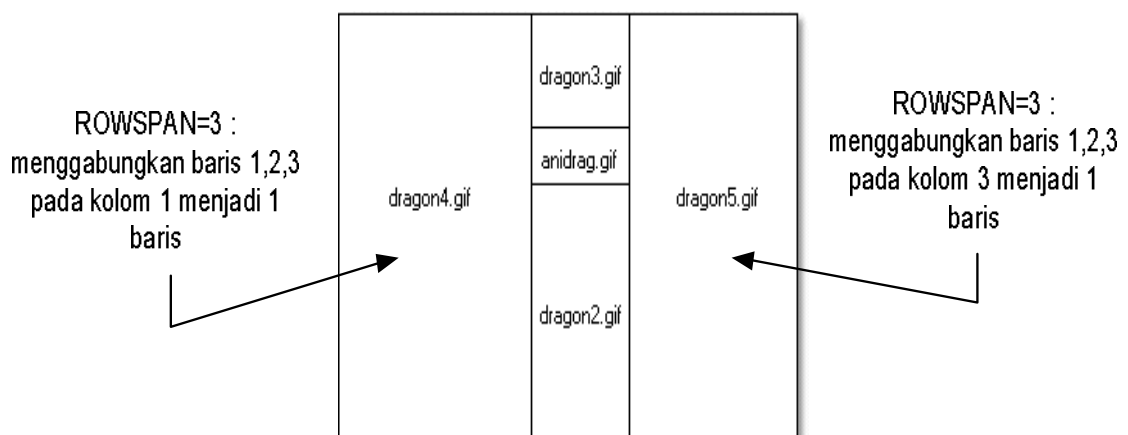
```
<HTML>
<HEAD>
<TITLE> Membuat Table </TITLE>
</HEAD>
<BODY>
<!-- memberi warna background Table,Row & Cell -->
<TABLE BGCOLOR="#FF6633" CELLSPACING=0 CELLPADDING=5 >
<TR BGCOLOR="#009900">
 <!-- <TD> dg atribut WIDTH -->
 <TD BGCOLOR="#9999FF" WIDTH=60%>Aris</TD>
 <TD WIDTH=20%>Shinta</TD>
 <TD WIDTH=20%>Rick</TD>
</TR>
<TR>
 <TD>Renata</TD>
 <TD>Curly</TD>
 <TD BGCOLOR="#9999FF">Mira</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

**Nama File : table4.html**

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

Nama File : table4.html	
<pre> &lt;TABLE BORDER=0 WIDTH=100%&gt; &lt;TR&gt;   &lt;TD WIDTH=40% ALIGN="right"&gt;     &lt;IMG SRC="images\titlepic04.gif" WIDTH=150 HEIGHT=138&gt;   &lt;/TD&gt;   &lt;TD WIDTH=60%&gt;     Pemrograman Internet     &lt;UL&gt;       &lt;LI&gt;HTML       &lt;LI&gt;JavaScript       &lt;LI&gt;PHP       &lt;LI&gt;JSP     &lt;/UL&gt;   &lt;/TD&gt; &lt;/TR&gt; &lt;/TABLE&gt; &lt;/BODY&gt; &lt;/HTML&gt; </pre>	

Menampilkan image dengan layout seperti pada gambar 3.3. dengan tujuan untuk mempercepat load image pada browser, hal ini disebabkan karena image yang besar dipotong-potong menjadi bagian kecil-kecil dengan tidak mengurangi informasi aslinya.



**Gambar 3.3.** Layout Menampilkan Image

Nama File : tableImage.html	
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Image Menggunakan Tabel &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; &lt;!-- Menampilkan image pada table --&gt; &lt;TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0&gt; &lt;CAPTION ALIGN="top"&gt;&lt;FONT SIZE=6&gt;&lt;B&gt;Netscape vs. Microsoft?&lt;/B&gt;&lt;/FONT&gt;&lt;/CAPTION&gt; &lt;TR&gt; &lt;!-- &lt;/TD&gt; jangan diberi efek enter,jika ditekan --&gt; &lt;!-- enter akibatnya tampilan image tidak bagus --&gt; &lt;!-- ROWSPAN=3 : menggabungkan 3 baris menjadi 1 baris --&gt; &lt;!-- COLSPAN=3 : menggabungkan 3 cell menjadi 1 cell --&gt;   &lt;TD ROWSPAN=3&gt;     &lt;IMG SRC="images/dragon4.gif" WIDTH=250 HEIGHT=406&gt;&lt;/TD&gt;   &lt;TD&gt;     &lt;IMG SRC="images/dragon3.gif" WIDTH=122 HEIGHT=109&gt;&lt;/TD&gt;   &lt;TD ROWSPAN=3&gt; </pre>	

Nama File : tableImage.html	
	<IMG SRC="images/dragon5.gif" WIDTH=219 HEIGHT=406></TD>
</TR>	
<TR>	
	<TD>
	<IMG SRC="images/anidrag.gif" WIDTH=122 HEIGHT=50></TD>
</TR>	
<TR>	
	<TD>
	<IMG SRC="images/dragon2.gif" WIDTH=122 HEIGHT=247></TD>
</TR>	
</TABLE>	
</BODY>	
</HTML>	

### 3.19. Forms

Konstruksi dasar form HTML adalah :

<FORM> -- pembukaan form --

<INPUT> -- bermacam-macam tipe inputan --

</FORM> -- penutupan form --

Nama File : form.html	
<HTML>	
<HEAD>	
<TITLE> Membuat Komponen-komponen FORM </TITLE>	
</HEAD>	
<BODY>	
<!-- saat ditekan submit buttons akan membuka halaman form.html-->	
<FORM ACTION="form.html">	
<B>Input Text Box</B> 	
<!-- input text box dengan type adl text sekaligus inisialisasi-->	
<INPUT TYPE="text" NAME="ADDRESS" VALUE="Jl. Banjarsari 50" SIZE=20>	
<!-- MAXLENGTH=10 : maksimum karakter pd input text box-->	
<INPUT TYPE="text" NAME="ADDRESS" SIZE=15 MAXLENGTH=10>	
<!-- input text box dengan type adl password -->	
<INPUT TYPE="password" NAME="USERPASS" MAXLENGTH=10>	
<!-- input text box dengan type adl hidden -->	
<!-- tidak ditampilkan pada browser -->	
<INPUT TYPE="hidden" NAME="LOCATION" VALUE="USA Form">	
<B>Radio Buttons</B>	
<!-- Radio Buttons dengan memberi nilai default CHECKED -->	
 <INPUT TYPE="radio" NAME="bahasa" VALUE="C" CHECKED>Bahasa C	
 <INPUT TYPE="radio" NAME="bahasa" VALUE="CPP">Bahasa C++	
 <INPUT TYPE="radio" NAME="bahasa" VALUE="Java">Bahasa Java	
<B>Checkbox</B>	
<!-- Checkbox dgn memberi nilai default CHECKED lebih dari 1 -->	
 <INPUT TYPE="checkbox" NAME="C" VALUE="YES" CHECKED>Bahasa C	
 <INPUT TYPE="checkbox" NAME="CPP" VALUE="YES">Bahasa C++	
 <INPUT TYPE="checkbox" NAME="Java" VALUE="YES" CHECKED>Bahasa Java	
 <INPUT TYPE="checkbox" NAME="Ada" VALUE="YES">Bahasa Ada	
<B>Pull Down List</B> 	

Nama File : form.html
<pre> &lt;!-- SELECT dgn memberi nilai default SELECTED --&gt; &lt;!-- SELECT dapat diberi atribut SIZE, lakukan perubahan --&gt; &lt;!-- &lt;SELECT NAME="bahasa" SIZE=4&gt; amati perubahannya --&gt; &lt;SELECT NAME="bahasa"&gt; &lt;OPTION VALUE="C"&gt;Bahasa C &lt;OPTION VALUE="CPP"&gt;Bahasa C++ &lt;OPTION VALUE="Java" SELECTED&gt;Bahasa Java &lt;OPTION VALUE="Ada"&gt;Bahasa Ada &lt;/SELECT&gt; &lt;BR&gt;&lt;BR&gt;  &lt;B&gt;Text Area&lt;/B&gt;&lt;BR&gt; &lt;!-- Textarea, nilai WRAP=hard/soft/off--&gt; &lt;TEXTAREA NAME="COMMENTS" ROWS=3 COLS=30 WRAP="hard"&gt; &lt;/TEXTAREA&gt; &lt;BR&gt;&lt;BR&gt; &lt;!-- mengambil nama file dari drive --&gt; &lt;INPUT TYPE="FILE" NAME="myfile"&gt;&lt;BR&gt;&lt;BR&gt; &lt;!-- submit buttons, ACTION membuka form.html --&gt; &lt;INPUT TYPE="submit" VALUE="Send to"&gt; &lt;!-- reset buttons --&gt; &lt;INPUT TYPE="reset" VALUE="Clear"&gt; &lt;!-- submit buttons menggunakan image --&gt; &lt;!-- ACTION membuka form.html --&gt; &lt;INPUT TYPE="image" SRC="images\submit.gif" WIDTH=94 HEIGHT=26 BORDER=0 ALT="Submit"&gt; &lt;/FORM&gt; &lt;/BODY&gt; &lt;/HTML&gt; </pre>

### 3.20. Frames

Tag HTML yang digunakan untuk membuat frame adalah <FRAMESET> dan <FRAME>.

Nama File : frameCol.html
<pre> &lt;!-- membuat frame berdasarkan kolom --&gt; &lt;FRAMESET COLS="10%,20%,30%,15%,25%"&gt;   &lt;!-- lebar kolom 10 % --&gt;   &lt;FRAME SRC="link.html"&gt;   &lt;!-- lebar kolom 20 % --&gt;   &lt;FRAME SRC="table.html"&gt;   &lt;!-- lebar kolom 30 % --&gt;   &lt;FRAME SRC="table1.html"&gt;   &lt;!-- lebar kolom 15 % --&gt;   &lt;FRAME SRC="table2.html"&gt;   &lt;!-- lebar kolom 25 % --&gt;   &lt;FRAME SRC="table3.html"&gt; &lt;/FRAMESET&gt; </pre>
Nama File : frameRow.html
<pre> &lt;!-- membuat frame berdasarkan baris --&gt; &lt;FRAMESET ROWS="10%,20%,30%,15%,25%"&gt;   &lt;!-- lebar baris 10 % --&gt;   &lt;FRAME SRC="link.html"&gt;   &lt;!-- lebar baris 20 % --&gt;   &lt;FRAME SRC="table.html"&gt;   &lt;!-- lebar baris 30 % --&gt;   &lt;FRAME SRC="table1.html"&gt;   &lt;!-- lebar baris 15 % --&gt;   &lt;FRAME SRC="table2.html"&gt;   &lt;!-- lebar baris 25 % --&gt;   &lt;FRAME SRC="table3.html"&gt; &lt;/FRAMESET&gt; </pre>

**Nama File : frameGap.html**

```
<!-- membuat frame berdasarkan baris dan kolom -->
<FRAMESET COLS="50,*,2*">
 <FRAME SRC="table3.html">
 <FRAME SRC="table.html">
 <FRAMESET ROWS="50%,50%">
 <FRAME SRC="table1.html">
 <FRAME SRC="table2.html">
 </FRAMESET>
</FRAMESET>
```

**Nama File : frameGap1.html**

```
<FRAMESET ROWS="60,*">
 <FRAME SRC="table.html" NAME="top">
 <FRAMESET COLS="160,*">
 <FRAME SRC="table1.html" NAME="left">
 <FRAME SRC="table2.html" NAME="main">
 </FRAMESET>
</FRAMESET>
```



## BAB 4

### Cascading Style Sheets (CSS)

CSS digunakan untuk mengorganisasikan dan mengontrol tampilan dokumen HTML secara efisien, sehingga dengan CSS dapat dilakukan sebagai berikut :

- Menambahkan style tampilan HTML.
- Melakukan perubahan ulang style web sites secara keseluruhan dengan hanya beberapa perubahan kode CSS.
- Menggunakan style pada halaman yang diinginkan.

Style sheets dapat secara penuh dipisahkan dari dokumen HTML. Pada saat, kita memiliki master CSS dan HTML, kita dapat memisahkan perancangan dan format (CSS) halaman web dari contents (HTML).

#### 4.1. Internal CSS

Untuk memformat halaman HTML dengan menggunakan CSS dapat dilakukan dengan cara internal, eksternal, dan inline. Pada bagian ini akan di bahas tentang pembuatan CSS secara internal. Untuk pembuatan CSS secara internal harus ditambahkan tag `<style>` yang diletakan diantara tag `<head>` dan `</head>`, seperti yang diberikan pada kode sumber 4.1.

**Nama File : cssLatih01.html**

```
<html>
<head>
<style type="text/css">
 <!--Pendefinisian Format Halaman dengan CSS secara internal-->
</style>
</head>
<body>
<p>Your pages content!</p>
</body>
</html>
```

**Kode Sumber 4.1.** Pendefinisian CSS Secara Internal

Kode CSS tidak dituliskan secara sama dengan kode HTML, hal ini disebabkan karena CSS bukan HTML, tetapi sebagai langkah untuk melakukan manipulasi HTML yang sudah ada. Untuk contoh sederhana diberikan pada kode sumber 4.2. dan 4.3.

**Nama File : cssLatih02.html**

```
<html>
<head>
<style type="text/css">
 p {color: white; }
 body {background-color: black; }
</style>
</head>
<body>
<p>Teks berwarna Putih dengan Background berwarna Hitam!</p>
</body>
</html>
```

**Kode Sumber 4.2.** Contoh Sederhana CSS Secara Internal

Pada kode sumber 4.2. dan 4.3. tag HTML yang dilakukan manipulasi adalah tag BODY dan P, dimana secara umum format CSS adalah :

**"HTML tag" { "CSS Property" : "Value" ; }**

**Nama File : cssLatih03.html**

```
<html>
<head>
<style>
 body { background-color: blue; }
 p { color: white; }
</style>
</head>
<body>
<h2>Internal CSS</h2>
<p>Halaman ini menggunakan internal CSS. Menggunakan style tag dapat
untuk melakukan modifikasi elemen-elemen HTML yang ditampilkan.</p>
</body>
</html>
```

**Kode Sumber 4.3.** Contoh Sederhana CSS Secara Internal**4.2. Eksternal CSS**

Pada saat menggunakan CSS disarankan lebih baik untuk memisahkan CSS dengan HTML. Penempatan CSS di dalam suatu file terpisah memungkinkan seorang perancang web dengan sepenuhnya untuk membedakan antara contents (HTML) dan design (CSS). Alasan menggunakan eksternal CSS: Untuk memelihara designs dan contents secara terpisah, Mudah menggunakan kode CSS untuk setiap halaman web dengan format yang sama, Mempermudah perubahan secara drastis terhadap halaman web dengan hanya melakukan beberapa perubahan pada file tunggal CSS. Eksternal CSS adalah file yang hanya berisi kode-kode CSS dan di simpan dengan ekstensi file **.css**. File CSS melakukan references ke HTML menggunakan `<link>` sesuai dengan `<style>` yang didefinisikan. Untuk contoh sederhana diberikan pada kode sumber 4.4. dan 4.5.

**Nama File : latihCSS01.css**

```
body{ background-color: gray }
p { color: blue; }
h3{ color: white; }
```

**Kode Sumber 4.4.** Pendefinisian CSS dalam File Terpisah**Nama File : cssLatih04.html**

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="latihCSS01.css" />
</head>
<body>
<h3> Header berwarna Putih </h3>
<p> Paragraph ini dengan font berwarna Biru. warna background halaman
ini adalah Gray karena dilakukan perubahan dengan CSS! </p>
</body>
</html>
```

**Kode Sumber 4.5.** Contoh Sederhana CSS Secara Eksternal**4.3. Inline CSS**

Inline CSS adalah menempatkan kode-kode CSS di dalam HTML. Inline CSS memiliki prioritas tertinggi dibandingkan dengan internal CSS dan eksternal CSS, hal ini berarti bahwa style yang didefinisikan pada internal CSS dan eksternal CSS dapat dilakukan override oleh inline CSS. Inline CSS bertentangan dengan tujuan CSS yang sebenarnya, sehingga dalam penggunaannya harus hati-hati. Format penulisan inline CSS adalah :

```
<htmltag style="cssproperty1: value; cssproperty2: value;"> </htmltag>
```

Untuk contoh penggunaan inline CSS diberikan pada kode sumber 4.6., dan 4.7.

**Nama File : cssLatih05.html**

```
<html>
<head>
<body>
<p style="background: blue; color: white;">warna background and font
dengan inline CSS</p>
<!-- Tidak Dapat Bekerja-->
<p style="background: url("links_bot_fill.gif");">
This is broken : (background menggunakan image)</p>
<!-- Dapat Bekerja-->
<p style="background: url(links_bot_fill.gif);">
This is working : (background menggunakan image)</p>
</body>
</html>
```

**Kode Sumber 4.6.** Contoh Sederhana CSS Secara Inline

**Nama File : cssLatih06.html**

```
<html>
<head>
<style>
</style>
</head>
<body>
<p style="background: blue; color: white;">warna background and font
dengan inline CSS</p>
<!--file links_bot_fill.gif berada pada folder-->
<!--C:\apache\htdocs\latihan\-->
<!--Dapat bekerja dengan benar jika apache juga di RUN-->

<!--Tidak dapat bekerja-->
<p style="background:
url("http://localhost/latihan/links_bot_fill.gif");">
This is broken : http://localhost/latihan/links_bot_fill.gif</p>

<!--Dapat bekerja-->
<p style="background:
url(http://localhost/latihan/links_bot_fill.gif);">
This is working : http://localhost/latihan/links_bot_fill.gif</p>
</body>
</html>
```

**Kode Sumber 4.7.** Varian Contoh Sederhana CSS Secara Inline

#### 4.4. CSS Classes

CSS memungkinkan untuk melakukan manipulasi elemen HTML dengan banyak efek, misalnya pada satu sisi diinginkan font ukuran besar dengan warna putih, dan pada sisi lain diinginkan font ukuran kecil dengan warna hitam. Secara normal CSS tidak memperbolehkan melakukan format yang berbeda untuk satu tag HTML. Untuk memberikan efek format yang banyak terhadap satu tag HTML dapat dilakukan dengan menggunakan classes yang didefinisikan di dalam kode CSS, seperti yang diberikan pada kode sumber 4.8.

**Nama File : cssLatih07.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk <P> -->
<style>
 p.satu { background-color: gray; color: red; font-size: 20px;}
</style>
```

```

 p.dua { background-color: red; }
 p.tiga {
 background: purple;
 color: white;
 }
 p.empat{ color: blue; }
 p.lima{ font-size: 12px; }
 p.enam{ color: red; font-size: 20px; }
</style>
</head>
<body>
<h2>CSS Menggunakan Classes</h2>
<!--Penggunaan format tag HTML berdasarkan style-->
<p class="satu">Ini adalah paragraph p.satu</p>
<p class="dua">Ini adalah paragraph p.dua</p>
<p class="tiga">Ini adalah paragraph p.tiga</p>
<p class="empat">Ini adalah paragraph p.empat</p>
<p class="lima">Ini adalah paragraph p.lima</p>
<p class="enam">Ini adalah paragraph p.enam</p>
</body>
</html>

```

**Kode Sumber 4.8.** Contoh Sederhana CSS Menggunakan Classes

#### 4.5. CSS Background

Background pada suatu halaman web adalah merupakan bagian yang penting, dengan menggunakan CSS dapat dilakukan setting background menggunakan warna atau gambar sebagai elemen dari CSS. CSS dapat digunakan untuk menampilkan background gambar, menampilkan background secara berulang baik horizontal, vertical, atau bukan kedua-duanya, menampilkan background sebagai fixed position atau secara scroll. Untuk contoh sederhana CSS background dan variannya diberikan pada kode sumber 4.9., 4.10., 4.11., 4.12., 4.13., dan 4.14.

##### Nama File : cssLatih08.html

```

<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 h4 { background-color: yellow; }
 p { background-color: #1078E1; }
 ul { background-color: rgb(149, 206, 145); }
</style>
</head>
<body>
<h4>Ini adalah tag h4 menggunakan background Yellow</h4>
<!--Penggunaan format tag HTML berdasarkan style-->
<p>Ini adalah paragraph</p>

 Bullet Nomor Satu
 Bullet Nomor Dua

</body>
</html>

```

**Kode Sumber 4.9.** Contoh Sederhana CSS Untuk Background Warna

Pada kode sumber 4.9. untuk mendefinisikan format background-color dapat digunakan dengan tiga cara, yaitu : color name, nilai hexadecimal (dimulai dengan tanda (#) dan terdiri dari 6 nilai hexsadesima (0 - F) ), dan RGB (adalah Red Blue Green dengan range untuk setiap nilai adalah 0-255).

**Nama File : cssLatih09.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 p { background-image: url(links_bot_fill.gif); }
 h4{ background-image:
url(http://localhost/latihan/links_bot_fill.gif); }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h4>Ini adalah tag h4 menggunakan background Image</h4>
<p>Ini adalah paragraph menggunakan background Image</p>
</body>
</html>
```

**Kode Sumber 4.10.** Contoh Sederhana CSS Untuk Background Image**Nama File : cssLatih10.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 p { background-image: url(links_bot_fill.gif);
 background-repeat: repeat;
 }
 h4 { background-image: url(links_bot_fill.gif);
 background-repeat: repeat-y;
 }
 ol { background-image: url(links_bot_fill.gif);
 background-repeat: repeat-x; }
 ul { background-image: url(links_bot_fill.gif);
 background-repeat: no-repeat;
 }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<p>Ini adalah paragraph menggunakan background Image</p>
<h4>Ini adalah tag h4 menggunakan background Image dg repeat
vertikal(y)</h4>

 Ordered list
 Dengan background Repeat
 Secara Horisontal (x)

 UnOrdered list
 Dengan background Repeat
 neither direction

</body>
</html>
```

**Kode Sumber 4.11.** Contoh Sederhana CSS Untuk Background Image Berulang

Pada kode sumber 4.11. image background dapat dilakukan secara horizontal (x), vertical (y), antara kedua-duanya, atau tidak kedua-duanya.

**Nama File : cssLatih11.html**

```

<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 textarea.noScroll { background-image: url(links_bot_fill.gif);
 background-attachment: fixed;
 }
 textarea { background-image: url(links_bot_fill.gif);
 background-attachment: scroll;
 }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<textarea name="textarea"></textarea>
<textarea name="textarea1"></textarea>
</body>
</html>

```

**Kode Sumber 4.12.** Contoh Sederhana CSS Untuk Fixed Background Image

Pada kode sumber 4.12. fixed background image scroll tidak dapat bekerja pada kebanyakan browser, kecuali Internet Explorer 6.0 dapat ditampilkan secara benar.

**Nama File : cssLatih12.html**

```

<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 p {
 background-image: url(links_bot_fill.gif);
 background-position: 20px 10px;
 }
 h4 { background-image: url(links_bot_fill.gif);
 background-position: 30% 30%;
 }
 ol { background-image: url(links_bot_fill.gif);
 background-position: top center;
 }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<p>Ini adalah paragraph menggunakan background Image Position</p>
<h4>Ini adalah tag h4 menggunakan background Image Position</h4>

 Ordered list
 Dengan background Position
 Menggunakan Keywords

</body>
</html>

```

**Kode Sumber 4.13.** Contoh Sederhana CSS Untuk Background Position

Pada kode sumber 4.13. untuk menempatkan posisi image background dapat menggunakan pixel, (A)px lokasi pixel dari kiri screen dan (B)px lokasi pixel dari atas screen, dengan A dan B adalah bilangan integer. Atau dapat menggunakan persentase, A% lokasi dari kiri screen dan B% lokasi dari atas screen, dengan A dan B adalah bilangan integer. Dan dapat juga menggunakan positioning keywords : top, right, bottom, left, and center.

**Nama File : cssLatih13.html**

```
<html>
<head>
<style>
 body { background-color: black; }
 p { background-color: yellow; }
 h2 { background-color: red; }
 ul { background-image: url(links_bot_fill.gif);
 background-repeat: repeat-y;
 color: green;
 }
</style>
</head>
<body>
<h2>CSS Backgrounds</h2>
<p>Background halaman ini adalah black. warna background, paragraph
dan header dapat dilakukan perubahan sesuai keinginan.</p>

 List memiliki image sebelah kiri
 karena efek repeat-y
 Dan tidak repeat-x

</body>
</html>
```

**Kode Sumber 4.14.** Contoh Sederhana CSS Untuk Varian Background**4.6. CSS Font**

CSS dapat digunakan untuk melakukan pengaturan terhadap text yang ditampilkan pada browser, baik dari sisi warna, ukuran, jenis huruf, format huruf, dan lain sebagainya, seperti yang diberikan pada kode sumber 4.15., 4.16., 4.17., 4.18., 4.19., dan 4.20.

**Nama File : cssLatih14.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { background-image: url(bg-kot.gif);}
 h3 { color: red; }
 h4 { color: #9000A1;}
 h5 { color: rgb(0, 220, 98);}
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS Fonts Color</h2>
<h3>Ini adalah header H3 menggunakan color: red</h3>
<h4>Ini adalah header H4 menggunakan color: #9000A1</h4>
<h5>Ini adalah header H5 menggunakan color: rgb(0, 220, 98)</h5>
</body>
</html>
```

**Kode Sumber 4.15.** Contoh Sederhana CSS Font Color

Pada kode sumber 4.15 untuk mendefinisikan format color dapat digunakan dengan tiga cara, yaitu : color name, nilai hexadecimal (dimulai dengan tanda (#) dan terdiri dari 6 nilai hexsadesima (0 - F) ), dan RGB (adalah Red Blue Green dengan range untuk setiap nilai adalah 0-255).

**Nama File : cssLatih15.html**

```
<html>
```

```

<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { background-image: url(bg-kot.gif);}
 h3 { font-family: sans-serif; }
 h4 { font-family: serif; }
 h5 { font-family: arial; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS Fonts Family</h2>
<h3>Ini adalah header H3 menggunakan font-family : sans-serif</h3>
<h4>Ini adalah header H4 menggunakan font-family : serif</h4>
<h5>Ini adalah header H5 menggunakan font-family : arial</h5>
</body>
</html>

```

**Kode Sumber 4.16.** Contoh Sederhana CSS Font Family

Font family dalam CSS dibagi menjadi dua kelompok : sans-serif yang merupakan font dengan ukuran besar, dan serif yang merupakan font dengan ukuran kecil, seperti yang diberikan pada kode sumber 4.16. Untuk san-serif lebih mudah dibaca di monitor computer dibandingkan dengan serif.

**Nama File : cssLatih16.html**

```

<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { background-image: url(bg-kot.gif);}
 p { font-size: 120%; }
 ol { font-size: 14px; }
 ul { font-size: x-large; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS Fonts Size</h2>
<p>Paragraph menggunakan ukuran font-size 120%</p>

 Ini adalah Ordered
 Dengan font-size adalah
 14px

 Ini adalah List
 Dengan font-size adalah
 x-large

</body>
</html>

```

**Kode Sumber 4.17.** Contoh Sederhana CSS Font Size

Pada kode sumber 4.17. untuk melakukan manipulasi ukuran font dapat digunakan nilai (ukuran font tidak terpengaruh pada saat browser dirubah ukuran font-nya), persentase (ukuran font terpengaruh pada saat browser dirubah ukuran font-nya), atau key terms (ukuran font terpengaruh pada saat browser dirubah ukuran font-nya). Key terms dapat meliputi xx-large, x-large, large, medium, small, x-small, dan xx-small.



**Nama File : cssLatih17.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { background-image: url(bg-kot.gif);}
 p { font-style: italic; }
 h4{ font-style: oblique; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS Fonts Style</h2>
<p>Paragraph menggunakan font-style : italic</p>
<h4>Ini adalah header H4 menggunakan font-style : oblique</h4>
</body>
</html>
```

**Kode Sumber 4.18.** Contoh Sederhana CSS Font Style

Pada kode sumber 4.18. untuk mendefinisikan font style italic atau bukan, key terms yang digunakan dapat meliputi italic, oblique, dan normal.

**Nama File : cssLatih18.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { background-image: url(bg-kot.gif);}
 p { font-weight: 100; }
 ul{ font-weight: bolder; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS Fonts Weight</h2>
<p>Paragraph menggunakan font-weight : 100</p>

 Ini adalah List
 Dengan font-weight adalah
 bolder(Tebal)

</body>
</html>
```

**Kode Sumber 4.19.** Contoh Sederhana CSS Font Weight

Pada kode sumber 4.19. untuk mendefinisikan ketebalan font dapat digunakan font-weight, yang nilainya pada range 100(thin) – 900(thick), atau dapat menggunakan key terms untuk font-weight bold, bolder, dan normal.

**Nama File : cssLatih19.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { background-image: url(bg-kot.gif);}
 p { font-variant: small-caps; }
</style>
</head>
```

```
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS Fonts Variant</h2>
<p>Paragraph menggunakan font-variant : small-caps</p>
</body>
</html>
```

**Kode Sumber 4.20.** Contoh Sederhana Variant CSS Font

Variant CSS font mengizinkan untuk mengkonversi font huruf kecil menjadi font huruf besar (caps lock) seperti pada kode sumber 4.20., tetapi tidak semua font mendukung variant CSS font, sehingga sebelum di publish sebaiknya dilakukan pengecekan terlebih dahulu.

#### 4.7. CSS Text

CSS text digunakan untuk melakukan pengontrolan terhadap spacing, decoration, dan alignment dari suatu text.

Pada halaman web underline biasanya merupakan halaman link, tetapi untuk membuat underline tanpa merupakan halaman link dapat menggunakan text-decoration, dimana underline dapat diletakan di atas, di tengah, atau di bawah dari text, seperti yang diberikan pada kode sumber 4.21.

##### Nama File : cssLatih20.html

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 h3{ text-decoration: line-through; }
 h4{ text-decoration: overline; }
 h5{ text-decoration: underline; }
 a { text-decoration: none; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS text-decoration</h2>
<h3>Ini adalah header H3 menggunakan text-decoration : line-
through</h3>
<h4>Ini adalah header H4 menggunakan text-decoration : overline</h4>
<h5>Ini adalah header H5 menggunakan text-decoration : underline</h5>
Ini adalah CSS Link menggunakan text-decoration :
none!
</body>
</html>
```

**Kode Sumber 4.21.** Contoh Sederhana CSS Text-Decoration

Untuk mendefinisikan indent pada suatu paragraph tanpa menggunakan format tag HTML atau menggunakan spasi secara manual, dapat digunakan CSS text-indent yang dapat direpresentasikan dengan nilai exact atau dengan persentase, disarankan lebih baik menggunakan nilai exact, seperti yang diberikan pada kode sumber 4.22.

##### Nama File : cssLatih21.html

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
```

```
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 p { text-indent: 20px; }
 h5 { text-indent: 30%; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS text-indent</h2>
<p>Ini adalah paragraph P menggunakan text-indent : 20px. Ini adalah
paragraph P menggunakan text-indent : 20px. Ini adalah paragraph P
menggunakan text-indent : 20px. Ini adalah paragraph P menggunakan
text-indent : 20px.</p>
<h5>Ini adalah header H5 menggunakan text-indent : 30%. Ini adalah
header H5 menggunakan text-indent : 30%. Ini adalah header H5
menggunakan text-indent : 30%. Ini adalah header H5 menggunakan text-
indent : 30%.</h5>
</body>
</html>
```

**Kode Sumber 4.22.** Contoh Sederhana CSS Text-Indent

Nilai default yang diberikan untuk menuliskan text adalah rata kiri, tetapi dapat dilakukan perubahan dengan menggunakan text-align, yang memiliki atribut right, justify, and left, seperti yang diberikan pada kode sumber 4.23.

**Nama File : cssLatih22.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 p { text-align: right; }
 h5{ text-align: justify; }
 h6{ text-align: left; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS text-align</h2>
<p>Ini adalah paragraph P menggunakan text-align: right. Ini adalah
paragraph P menggunakan text-align: right. Ini adalah paragraph P
menggunakan text-align: right. Ini adalah paragraph P menggunakan
text-align: right.</p>
<h5>Ini adalah header H5 menggunakan text-align: justify. Ini adalah
header H5 menggunakan text-align: justify. Ini adalah header H5
menggunakan text-align: justify. Ini adalah header H5 menggunakan
text-align: justify. Ini adalah header H5 menggunakan text-align:
justify. Ini adalah header H5 menggunakan text-align: justify. Ini
adalah header H5 menggunakan text-align: justify.</h5>
<h6>Ini adalah header H6 menggunakan text-align: left. Ini adalah
header H6 menggunakan text-align: left. Ini adalah header H6
menggunakan text-align: left. Ini adalah header H6 menggunakan text-
align: left. Ini adalah header H6 menggunakan text-align: left. Ini
adalah header H6 menggunakan text-align: left.</h6>
</body>
</html>
```

**Kode Sumber 4.23.** Contoh Sederhana CSS Text-Align

Untuk melakukan transformasi (capitalize, uppercase, dan lowercase) secara cepat terhadap format penulisan text dapat digunakan text-transform, seperti yang diberikan pada kode sumber 4.24.

**Nama File : cssLatih23.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 p { text-transform: capitalize; }
 h3{ text-transform: uppercase; }
 h4{ text-transform: lowercase; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS text-transform</h2>
<p>Ini adalah paragraph P menggunakan text-transform :
capitalize.</p>
<h3>Ini adalah header H3 menggunakan text-transform : uppercase.</h3>
<h4>Ini adalah header H4 menggunakan text-transform : lowercase.</h4>
</body>
</html>
```

**Kode Sumber 4.24.** Contoh Sederhana CSS Text-Transform

Untuk menuliskan text dan memberikan efek ganti baris tanpa harus memberikan tag HTML <BR> dapat dilakukan dengan menggunakan white-space, yang memiliki atribut nowrap (tanpa efek ganti baris), dan wrap (efek ganti baris), seperti yang diberikan pada kode sumber 4.25.

**Nama File : cssLatih24.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 p { white-space: nowrap; }
 h1 { white-space: wrap; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS white-space attribute</h2>
<p>Ini adalah paragraph P menggunakan white-space: nowrap. Ini adalah
paragraph P menggunakan white-space: nowrap. Ini adalah paragraph P
menggunakan white-space: nowrap. Ini adalah paragraph P menggunakan
white-space: nowrap.</p>
<h1>Ini adalah header H1 menggunakan white-space: wrap. Ini adalah
header H1 menggunakan white-space: wrap. Ini adalah header H1
menggunakan white-space: wrap.</h1>
</body>
</html>
```

**Kode Sumber 4.25.** Contoh Sederhana CSS White-Space Attribute

Untuk menspesifikan ukuran spasi antar kata dan karakter dapat digunakan word spacing dan letter spacing dengan menggunakan nilai exact, seperti yang diberikan pada kode sumber 4.26.

**Nama File : cssLatih25.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 p { word-spacing: 20px; }
 h1 { letter-spacing: 5px; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS white-space attribute</h2>
<p>Ini adalah paragraph P menggunakan word-spacing (jarak antar
kata): 20px. </p>
<h1>Ini adalah header H1 menggunakan letter-spacing (jarak antar
karakter) : 5px.</h1>
</body>
</html>
```

**Kode Sumber 4.26.** Contoh Sederhana CSS Word dan Letter Spacing

#### 4.8. CSS Padding

Padding adalah jarak antara elemen-elemen border dan isinya. CSS padding dapat digunakan untuk merubah default padding elemen-elemen HTML yang tampak (paragraph, tabel, dan sebagainya), seperti yang diberikan pada kode sumber 4.27. yang menggambarkan padding yang diberikan akan berlaku sama dalam satu halaman untuk top, bottom, right, dan left.

**Nama File : cssLatih26.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 p { padding: 15px; border: 1px solid black; }
 h3 { padding: 0px; border: 1px solid red; }
 h4 { padding: 2%; border: 1px solid black; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS padding one value</h2>
<p>Ini adalah paragraph P menggunakan padding : 15px; border : 1px
solid black. </p>
<h3>Ini adalah header H3 menggunakan padding : 0px; border : 1px
solid red.</h3>
<h4>Ini adalah header H4 menggunakan padding : 2%; border : 1px solid
black.</h4>
</body>
</html>
```

**Kode Sumber 4.27.** Contoh Sederhana CSS Padding One Value

Setiap elemen HTML secara aktual memiliki empat padding yang berbeda : top, bottom, right, dan left. Hal ini memungkinkan untuk mendefinisikan masing-masing padding secara berbeda dengan menambahkan suffix pada atribut (top, bottom, right, dan left) padding, seperti yang diberikan pada kode sumber 4.28.

**Nama File : cssLatih27.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 p { padding-left: 5px; border: 1px solid black; }
 h3{ padding-top: 0px;
 padding-right: 2px;
 padding-bottom: 13px;
 padding-left: 21px;
 border: 1px solid red; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS padding (direction)</h2>
<p>Ini adalah paragraph P menggunakan padding: 15px; border : 1px
solid black. Paragraph ini hanya menggunakan satu deklarasi padding
direction secara spesifik pada left dan tiga yang lainnya tidak
dispesifikan</p>
<h3>Ini adalah header H3 yang memiliki spesifikasi untuk setiap
padding dengan deklarasi padding direction sendiri-sendiri.</h3>
</body>
</html>
```

**Kode Sumber 4.28.** Contoh Sederhana CSS Padding Direction

Nilai empat padding dapat dideklarasikan dengan menggunakan dua atau empat nilai. Pada saat menggunakan dua nilai, berarti yang pertama mendefinisikan top dan bottom dan yang kedua mendefinisikan left dan right. Sedangkan pada saat menggunakan empat nilai, secara langsung berhubungan dengan top, right, bottom, dan left, seperti yang diberikan pada kode sumber 4.29.

**Nama File : cssLatih28.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 p { padding: 5px 15px;
 border: 1px solid black; }
 h3{ padding: 0px 5px 10px 3px;
 border: 1px solid red;}
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS padding two and four value</h2>
<p>Ini adalah paragraph P memiliki padding top dan bottom adalah 5
pixel; padding left dan right adalah 15 pixel.</p>
<h3>Ini adalah header H3 memiliki padding top adalah 0 pixel; padding
```

```
right adalah 5 pixel; padding bottom adalah 10 pixel; dan padding
left adalah 3 pixel.</h3>
</body>
</html>
```

**Kode Sumber 4.29.** Contoh Sederhana CSS dengan Two and Four Value

#### 4.9. CSS Margin

CSS margin adalah identik dengan CSS padding termasuk atribut yang dimilikinya adalah sama. Perbedaan yang terpenting pada definisi margin adalah jarak antara elemen-elemen border HTML. Untuk contoh CSS margin sama seperti pada CSS padding dengan cara melakukan perubahan semua atribut **padding** menjadi **margin** pada kode sumber 4.27, 4.28., dan 4.29., dan kemudian amati efek perubahan yang terjadi.

#### 4.10. CSS Border

CSS border digunakan untuk membuat border yang tampak pada sekitar elemen HTML. Banyak variasi untuk membuat type border seperti yang diberikan pada kode sumber 4.30., dengan nilai default border style adalah hidden.

**Nama File : cssLatih29.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 p.solid {border-style: solid; }
 p.double {border-style: double; }
 p.groove {border-style: groove; }
 p.dotted {border-style: dotted; }
 p.dashed {border-style: dashed; }
 p.inset {border-style: inset; }
 p.outset {border-style: outset; }
 p.ridge {border-style: ridge; }
 p.hidden {border-style: hidden; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS border style</h2>
<p class="solid">Ini adalah paragraph P dengan border-style :
solid</p>
<p class="double">Ini adalah paragraph P dengan border-style :
double</p>
<p class="groove">Ini adalah paragraph P dengan border-style :
groove</p>
<p class="dotted">Ini adalah paragraph P dengan border-style :
dotted</p>
<p class="dashed">Ini adalah paragraph P dengan border-style :
dashed</p>
<p class="inset">Ini adalah paragraph P dengan border-style :
inset</p>
<p class="outset">Ini adalah paragraph P dengan border-style :
outset</p>
<p class="ridge">Ini adalah paragraph P dengan border-style :
ridge</p>
<p class="hidden">Ini adalah paragraph P dengan border-style :
hidden</p>
</body>
```

---

</html>
**Kode Sumber 4.30.** Contoh Sederhana CSS Border Style

Untuk ketebalan border dapat digunakan border-width, dengan disertai border-style. Border-width didefinisikan dengan nilai exact atau dengan key terms : thin, medium, dan thick, seperti yang diberikan pada kode sumber 4.31.

**Nama File : cssLatih30.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 table { border-width: 7px;
 border-style: outset; }
 td { border-width: medium;
 border-style: ridge; }
 p { border-width: thick;
 border-style: solid; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS border width</h2>
<table>
 <tr>
 <td>Table menggunakan outset border</td>
 <td>dan setiap cell menggunakan</td>
 <td>ridge border.</td>
 </tr>
 <tr>
 <td>Table menggunakan outset border</td>
 <td>dan setiap cell menggunakan</td>
 <td>ridge border.</td>
 </tr>
</table>
<p>Ini adalah paragraph P dengan border-width : thick; border-style :
solid</p>
</body>
</html>
```

**Kode Sumber 4.31.** Contoh Sederhana CSS Border Width

Warna border dapat dilakukan perubahan dengan menggunakan border color, dengan representasi dapat didefinisikan dengan RGB, hexadecimal, atau key terms, seperti yang diberikan pada kode sumber 4.32.

**Nama File : cssLatih31.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 table { border-color: rgb(100, 100, 255);
 border-style: dashed; }
 td { border-color: #FFBD32;
 border-style: ridge; }
 p { border-width : thick;
 border-color: blue;
```



```
border-style: solid; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS border color</h2>
<table>
 <tr>
 <td>Table menggunakan dashed border</td>
 <td>dan setiap cell menggunakan</td>
 <td>ridge border.</td>
 </tr>
 <tr>
 <td>Table menggunakan dashed border</td>
 <td>dan setiap cell menggunakan</td>
 <td>ridge border.</td>
 </tr>
</table>
<p>Ini adalah paragraph P dengan border-width : thick; border-color:
blue; border-style : solid</p>
</body>
</html>
```

**Kode Sumber 4.32.** Contoh Sederhana CSS Border Color

Setiap sisi pada border dapat diberlakukan secara berbeda-beda, dengan menspesifikan border secara direction yang diikuti dengan atribut border : top, right, bottom, dan left, seperti yang diberikan pada kode sumber 4.33.

**Nama File : cssLatih32.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 p { border-bottom-style: dashed ;
 border-bottom-color: yellow;
 border-bottom-width: 5px; }
 h4 { border-top-style: double;
 border-top-color: purple;
 border-top-width: thick; }
 h5 { border-left-style: groove;
 border-left-color: green;
 border-left-width: 15px;
 border-bottom-style: ridge;
 border-bottom-color: yellow;
 border-bottom-width: 25px; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS border (direction)</h2>
<p>Ini adalah paragraph P dengan border-bottom-style : dashed ;
border-bottom-color : yellow; border-bottom-width : 5px;</p>
<h4>Ini adalah header H4 dengan border-top-style : double; border-
top-color : purple; border-top-width : thick;</h4>
<h5>Ini adalah header H5 dengan border-left-style : groove; border-
left-color: green; border-left-width : 15px; border-bottom-
style : ridge; border-bottom-color : yellow; border-bottom-width
: 25px;</h5>
</body>
```

```
</html>
```

#### Kode Sumber 4.33. Contoh Sederhana CSS Border Direction

Untuk mempersingkat cara penulisan, dalam representasi border dapat dituliskan dalam satu baris, seperti yang diberikan pada kode sumber 4.34.

#### Nama File : cssLatih33.html

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 p { border: 20px outset blue ;}
 h4{ border: 5px solid; }
 h5{ border: dotted; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS border all in one</h2>
<p>Ini adalah paragraph P dengan border : 20px outset blue</p>
<h4>Ini adalah header H4 dengan border : 5px solid</h4>
<h5>Ini adalah header H5 border : dotted</h5>
</body>
</html>
```

#### Kode Sumber 4.34. Contoh Sederhana CSS Border All in One

### 4.11. CSS List

List terdiri dari dua, yaitu : unordered list dan ordered list. CSS mengijinkan image digunakan sebagai bullets pada unordered list. Atribut list style terdiri dari, seperti yang diberikan pada kode sumber 4.35. :

1. Unordered list style : square, circle, disc(default), dan none.
2. Ordered list style : upper-alpha, lower-alpha, upper-roman, lower-roman, decimal(default), dan none.

#### Nama File : cssLatih34.html

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 ol { list-style-type: upper-roman; }
 ul { list-style-type: circle; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS list style type</h2>

 Ini adalah ordered list
 dengan Upper Roman style
 menggunakan CSS

 Ini adalah unordered list
 dengan bullets circle style

```

```
menggunakan CSS

</body>
</html>
```

**Kode Sumber 4.35.** Contoh Sederhana CSS List Style Type

Pada Unordered list untuk bullets style dapat menggunakan image, sebaiknya ukuran image-nya jangan terlalu besar disesuaikan dengan ukuran font, seperti yang diberikan pada kode sumber 4.36.

**Nama File : cssLatih35.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 ol { list-style-type: upper-alpha; }
 ul { list-style-image: url("18_integration.gif"); }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS list with image (unordered list)</h2>

 Ini adalah ordered list
 dengan upper-alpha style
 menggunakan CSS

 Ini adalah unordered list
 dengan bullets menggunakan image "18_integration.gif"
 dan spasinya terlihat tidak bagus

</body>
</html>
```

**Kode Sumber 4.36.** Contoh Sederhana CSS List With Image

CSS memungkinkan untuk mengatur indent terhadap list item, yang representasinya hanya menggunakan key terms : outside dan inside, dengan nilai default adalah outside, seperti yang diberikan pada kode sumber 4.37.

**Nama File : cssLatih36.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 ul { list-style-position: inside; }
 ol { list-style-position: outside; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS list positioning</h2>

 Ini adalah ordered list position outside
 dengan decimal(default) style
 menggunakan CSS list position
```

```


 Ini adalah unordered list position inside
 dengan disc(default) style
 menggunakan CSS list position

</body>
</html>

```

**Kode Sumber 4.37.** Contoh Sederhana CSS List Position

CSS memungkinkan untuk mengkombinasikan semua teknik CSS yang diberikan pada bagian sebelumnya dalam satu kode baris CSS yang biasanya akan sangat berguna pada saat menggunakan list image dengan image yang tidak dapat dilakukan load, seperti yang diberikan pada kode sumber 4.38.

**Nama File : cssLatih37.html**

```

<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 ul { list-style: upper-roman inside url("18_quick.gif");}
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS list all in one</h2>

 Ini adalah unordered list all in one
 dengan bullets menggunakan image "18_quick.gif"
 dan spasinya terlihat tidak bagus

</body>
</html>

```

**Kode Sumber 4.38.** Contoh Sederhana CSS All in One**4.12. CSS Links (Pseudo-classes)**

Link memiliki empat state yang berbeda, yang masing-masing dapat state dapat dilakukan customize sendiri-sendiri. State link meliputi sebagai berikut :

1. link

Efek link sebelum digunakan

2. visited

Efek link sesudah digunakan

3. hover

Efek mouse pointer pada link

4. active

Efek link pada saat dilakukan proses (klik)

Nama state didefinisikan secara pseudo-class berdasarkan state yang digunakan dengan urutan link-visited-hover-active, seperti yang diberikan pada kode sumber 4.39.

**Nama File : cssLatih38.html**

```

<html>
<head>
<!--Pendefinisian format tag HTML untuk-->

```

```
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 a:link { color: red; }
 a:visited { color: red; }
 a:hover { color: blue; }
 a:active { color: yellow; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS links</h2>
Ini adalah CSS Link! Menggunakan state link color: red;
visited color: red; hover color: blue; active color: yellow
</body>
</html>
```

**Kode Sumber 4.39.** Contoh Sederhana CSS Link

Underline pada link dapat dihilangkan dengan menggunakan `text-decoration: none`, seperti yang diberikan pada kode sumber 4.40.

**Nama File : cssLatih39.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 a:link { color: red; text-decoration: none; }
 a:visited { color: red; text-decoration: none; }
 a:hover { color: blue; text-decoration: underline;}
 a:active { color: yellow; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS links: Removing the Default Underline</h2>
Ini adalah CSS Link! Menggunakan state link { color: red;
text-decoration: none; } visited { color: red; text-decoration: none;
} hover { color: blue; text-decoration: underline;} active { color:
yellow; }
</body>
</html>
```

**Kode Sumber 4.40.** Contoh Sederhana CSS Link Tanpa Underline

Kode sumber 4.41., 4.42., dan 4.3. adalah merupakan contoh untuk melakukan manipulasi state suatu hyperlinks (links couple) dengan menggunakan beberapa form CSS.

**Nama File : cssLatih40.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}

 a:link { color: white;
 background-color: black;
 text-decoration: none;
```

```

 padding: 3px 3px 3px 3px;
 border: 2px solid white; }

a:visited { color: white;
background-color: black;
text-decoration: none;
padding: 3px 3px 3px 3px;
border: 2px solid white; }

a:hover { color: black;
background-color: white;
text-decoration: none;
padding: 3px 3px 3px 3px;
border: 2px solid black; }

a:active { color: yellow; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS links: Menggunakan Couple</h2>
Variant Form CSS Link-1

Variant Form CSS Link-2
</body>
</html>

```

**Kode Sumber 4.41.** Contoh Sederhana Variant-1 CSS Link

**Nama File : cssLatih41.html**

```

<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
background-image: url(bg-kot.gif);}

 a:link { color: blue;
background-color: red;
font-size: 26px;
border: 10px outset blue;
font-family: sans-serif;
text-transform: lowercase;
text-decoration: none;}

 a:visited {
color: blue;
background-color: red;
font-size: 26px;
border: 10px outset blue;
font-family: sans-serif;
text-transform: lowercase;
text-decoration: none;}

 a:hover{ color: blue;
background-color: red;
font-size: 27px;
border: 10px inset blue;
font-family: serif;
text-transform: uppercase;
text-decoration: line-through;
letter-spacing: 3px;
word-spacing: 6px;
font-weight: normal; }

```

```
 a:active { color: yellow; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS links: Menggunakan Couple</h2>
Variant Form CSS Link-1

Variant Form CSS Link-2
</body>
</html>
```

**Kode Sumber 4.42.** Contoh Sederhana Variant-2 CSS Link

**Nama File : cssLatih42.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}

 a:link{ text-decoration: none;
 color: gray; }

 a:visited{ text-decoration: none;
 color: gray; }

 a:hover{ text-decoration: none;
 color: green;
 font-weight: bolder;
 letter-spacing: 2px; }

 a:active { color: yellow; }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS links: Menggunakan Couple</h2>
Variant Form CSS Link-1

Variant Form CSS Link-2
</body>
</html>
```

**Kode Sumber 4.43.** Contoh Sederhana Variant-3 CSS Link

#### 4.13. CSS Mouse Cursor

Pada saat bekerja dengan system operasi Windows, Linux, atau Macintosh akan dirasakan perbedaan pada cursor icon mouse. Dalam kondisi normal cursor icon mouse berbentuk arrow, "|" ketika dilakukan pemilihan text, dan beberapa logo animasi pada saat komputer bekerja. CSS dapat digunakan untuk merubah cursor icon mouse yang sesuai dengan keinginan. Beberapa properties cursor icon adalah, seperti yang diberikan pada kode sumber 4.44. :

- Default : menampilkan cursor icon mouse secara normal
- Wait : cursor icon mouse pada saat komputer bekerja
- Crosshair : icon tanda plus
- Text : icon "|" ketika dilakukan pemilihan text
- Pointer : icon tangan pada saat hover pada halaman link
- Help : question mark

**Nama File : cssLatih43.html**

```
<html>
<head>
<!--Pendefinisian format tag HTML untuk-->
<style>
 body { font-family: serif;
 background-image: url(bg-kot.gif);}
 p { cursor: default }
 h1 { cursor: wait }
 h3 { cursor: crosshair }
 h4 { cursor: text }
 h5 { cursor: pointer }
 h6 { cursor: help }
</style>
</head>
<body>
<!--Penggunaan format tag HTML berdasarkan style-->
<h2>CSS Cursor Icon Mouse</h2>
<p>Cursor Icon Mouse : default</p>
<h1>Cursor Icon Mouse : wait</h1>
<h3>Cursor Icon Mouse : crosshair</h3>
<h4>Cursor Icon Mouse : text</h4>
<h5>Cursor Icon Mouse : pointer</h5>
<h6>Cursor Icon Mouse : help</h6>
</body>
</html>
```

**Kode Sumber 4.44.** Contoh Sederhana CSS Cursor Icon Mouse



## BAB 5

### Java Script (JS)

JS adalah bermafaat untuk menambahkan interaksi antara halaman web dengan pengunjung halaman web. JS memberikan kemampuan fitur-fitur tambahan halaman web yang lebih baik dibandingkan dengan fasilitas fitur-fitur yang terdapat pada HTML dan CSS yang telah diberikan pada bagian sebelumnya. JS merupakan bahasa client-side scripting, dimana scripts-nya akan dijalankan pada sisi client (oleh browser).

#### 5.1. Sintak JS

Sintak penulisan kode JS diawali dengan tag `<script type="text/javascript">` (untuk memberitahu kepada browser bahwa ini merupakan kode-kode yang dituliskan dalam JS) dan diakhiri dengan tag `</script>`, dan kemudian semua perintah-perintah kode JS dituliskan diantara tag script tersebut dan untuk setiap akhir perintah kode JS tidak membutuhkan tanda semicolons (membutuhkan tanda semicolons jika dua perintah atau lebih dituliskan dalam satu baris), seperti yang diberikan pada kode sumber 5.1.

##### Nama File : jsLatih01.html

```
<html>
<body>
<script type="text/javascript">
// ini adalah "//" sebagai komentar pada kode JS
// Isi Perintah-perintah Kode Java Script
//menampilkan string pada browser
 document.write("Hello world, Java Script!")
</script>
</body>
</html>
```

**Kode Sumber 5.1.** Representasi Penulisan Kode JS pada Tag BODY

#### 5.2. Head Body

Terdapat tiga area halaman HTML yang dapat digunakan untuk menempatkan kode-kode JS, yaitu : antara tag HEAD, antara tag BODY, dan menggunakan file eksternal (merupakan bagian yang akan dibahas pada buku ini). Jika script ingin dijalankan pada beberapa event maka sebaiknya kode JS diletakkan pada tag HEAD seperti yang diberikan pada kode sumber 5.2. (fungsi alert), tetapi jika script dijalankan pada saat halaman web di load maka sebaiknya kode JS diletakkan pada tag BODY seperti yang diberikan pada kode sumber 5.1.

##### Nama File : jsLatih02.html

```
<html>
<head>
<script type="text/javascript">
 //fungsi popup yang dituliskan dalam kode JS
 function popup() {
 alert("Hello world, Java Script")
 }
</script>
</head>
<body>
 <input type="button" onclick="popup()" value="popup">
</body>
</html>
```

**Kode Sumber 5.2.** Representasi Penulisan Kode JS pada Tag HEAD

#### 5.3. JS File Eksternal

Bekerja dengan file eksternal dibutuhkan untuk melakukan proses import terhadap file eksternal tersebut. Hal yang harus diperhatikan pada saat import file eksternal adalah :

file yang di-import adalah merupakan file JS yang valid dan hanya file JS, file harus berekstensi file **.js**, dan harus diketahui lokasi dari file eksternal tersebut.

Menggunakan kode sumber 5.2. akan dilakukan perubahan dengan meletakkan kode-kode JS sebagai file eksternal dan disimpan dengan nama file jsLatih03.js yang berada pada folder jscript dan kemudian di import pada file jsLatih04.html, seperti yang diberikan pada kode sumber 5.3. dan 5.4. File eksternal digunakan pada saat script yang sama digunakan pada beberapa halaman web, tetapi tidak dibutuhkan untuk menuliskan kembali script tersebut.

#### Nama File : jsLatih03.js

```
//fungsi popup sebagai file eksternal
function popup() {
 alert("Hello world, Java Script")
}
```

**Kode Sumber 5.3.** File Eksternal JS

#### Nama File : jsLatih04.html

```
<html>
<head>
<!--memanggil file eksternal jsLatih03.js pada folder jscript-->
<script src="jscript/jsLatih03.js">
</script>
</head>
<body>
 <input type="button" onclick="popup()" value="Click Me!">
</body>
</html>
```

**Kode Sumber 5.4.** JS Menggunakan File Eksternal

### 5.4. JS Operator

Operator dalam JS sama dengan definisi operator pada kebanyakan bahasa-bahasa pemrograman (pascal, C ansi, Java, dan lain sebagainya). Operator aritmatika yang digunakan pada JS diberikan pada tabel 5.1., sedangkan contoh penggunaan operator diberikan pada kode sumber 5.5.

**Tabel 5.1.** Operator Aritmatika

Operator	Keterangan	Contoh
+	Penjumlahan	2 + 5 = 7
-	Pengurangan	7 - 3 = 4
*	Perkalian	3 * 4 = 12
/	Pembagian	10 / 2 = 5
%	Sisa Hasil Bagi (modulus)	5 % 2 = 1

#### Nama File : jsLatih05.html

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
//deklarasi variabel sekaligus inisialisasi
 var two = 2
 var ten = 10
 var linebreak = "
"
//penjumlahan
 document.write("two + ten = ")
 result = two + ten
 document.write(result)
 document.write(linebreak)
//perkalian
 document.write("ten * ten = ")
 result = ten * ten
```

```

document.write(result)
document.write(linebreak)
//pembagian
document.write("ten / two = ")
result = ten / two
document.write(result)
document.write(linebreak)
//sisa hasil bagi
document.write("ten % two = ")
result = ten % two
document.write(result)
</script>
</body>
</html>

```

**Kode Sumber 5.5.** Operator Aritmatika

Operator perbandingan digunakan untuk melakukan pengecekan hubungan antara variabel-variabel dan/atau dengan suatu nilai. Untuk operator perbandingan diberikan pada tabel 5.2.

**Tabel 5.2.** Operator Perbandingan

Operator	Keterangan	Contoh	Hasil
==	Sama dengan	\$X == \$Y	False
!=	Tidak sama dengan	\$X != \$Y	True
<	Kurang dari	\$X < \$Y	True
>	Lebih besar dari	\$X > \$Y	False
<=	Kurang dari atau sama dengan	\$X <= \$Y	True
>=	Lebih besar dari atau sama dengan	\$X >= \$Y	False

### 5.5. JS Variabel

Variabel dalam JS adalah sama dengan definisi variabel pada kebanyakan bahasa-bahasa pemrograman (C, Java, dan lain sebagainya), dan variabel pada JS bersifat case sensitive. Variabel digunakan untuk untuk menyimpan informasi yang dapat digunakan pada bagian berikutnya. Pendefinisian variabel dalam JS tidak harus diawali dengan menggunakan kata "var", tetapi untuk alasan pemrograman yang baik disarankan untuk menggunakan kata "var", seperti yang diberikan pada kode sumber 5.6.

#### Nama File : jsLatih06.html

```

<html>
<head>
</head>
<body>
<script type="text/javascript">
//deklarasi menggunakan var dan sekaligus inisialisasi
var linebreak = "
"
var my_var = "Hello world!"
//deklarasi tanpa menggunakan var
tanpaVar = "Variable Declaration without var"
document.write(my_var)
document.write(linebreak)
document.write(tanpaVar)
document.write(linebreak)
//assignment
my_var = "I am learning javascript!"
document.write(my_var)
document.write(linebreak)
//assignment
my_var = "Script is Finishing up..."
document.write(my_var)
</script>
</body>
</html>

```

**Kode Sumber 5.6. Variabel****5.6. JS Function**

Function digunakan untuk menspesifikan suatu kode-kode JS yang banyak/sering kali dimanfaatkan pada suatu halaman web, sehingga dalam penulisan kode-kode JS tersebut cukup sekali tidak perlu secara berulang-ulang. Untuk penggunaan function tersebut, cukup memanggil nama dari function (dapat dilakukan secara berulang-ulang), dan memberikan efek yang sama terhadap pemanggilan yang berulang-ulang tersebut. Function tidak dieksekusi pada saat halaman web di-load, sehingga sebaiknya function tersebut diletakkan pada tag HEAD, seperti yang diberikan pada kode sumber 5.2, 5.3., dan 5.4. Bentuk penulisan function secara umum :

```
function function_name() {
 //body function
}
```

**5.7. Events**

Events dalam JS adalah sesuatu yang terjadi dengan atau pada suatu halaman web. Beberapa contoh events adalah sebagai berikut :

- Mouse click
- Pada saat melakukan load halaman web
- Pada saat mouse dipindahkan ke suatu objek
- Memilih input box pada form HTML
- Keystroke (tombol)

Untuk meng-capture dan membuat suatu kejadian pada saat events terjadi dibutuhkan untuk menspesifikan events tersebut, kemudian elemen HTML menunggu events tersebut, dan fungsinya akan dijalankan pada saat events-nya terjadi, seperti yang diberikan pada kode sumber 5.7.

**Nama File : jsLatih07.html**

```
<html>
<head>
<script type="text/javascript">
 //mendefinisikan fungsi
 function popup() {
 alert("Hello World, Java Script")
 }
</script>
</head>
<body>
<h2>Events yang digunakan: onClick,onMouseMove,onMouseOut</h2>
<!--pendefinisian events: onClick,onmouseover,onMouseout-->
<input type="button" value="Click Me!" onclick="popup()">

<a href="#" onMouseMove="window.status= 'Menggunakan onMouseMove yang
diisikan pada window.status'" return true
onMouseOut="window.status=''">Gerakan Cursor disini!
</body>
</html>
```

**Kode Sumber 5.7. Events****5.8. JS Alert**

JS alert adalah suatu dialog box yang tampil dan fokusnya keluar dari current window, dan melakukan pemaksaan terhadap web browser untuk membaca message. Untuk contoh alert diberikan pada kode sumber 5.2., 5.3., 5.4., dan 5.7. Kondisi ideal menggunakan JS alert adalah :

- Jika secara mutlak message harus dibaca terlebih dahulu sebelum melakukan eksplorasi terhadap halaman web
- Jika ingin memberikan suatu peringatan tertentu terhadap pengunjung halaman web
- Pada saat terjadinya error dan error tersebut diinformasikan kepada pengunjung halaman web
- Pada saat menanyakan konfirmasi pengunjung halaman web terhadap suatu action tertentu

### 5.9. JS Confirm

JS confirm pada prinsipnya sama dengan JS alert. Pada confirm box terdapat dua pilihan, yaitu **OK** (jika pengunjung web setuju dengan message yang tampilkan pada window (bernilai **1** jika **OK**)) dan **Cancel** (jika pengunjung web tidak setuju dengan message yang tampilkan pada window (bernilai **0** jika **Cancel**)), seperti yang diberikan pada kode sumber 5.8.

**Nama File : jsLatih08.html**

```
<html>
<head>
<script type="text/javascript">
 //pendefinisian fungsi
 function confirmation() {
 //deklarasi dan sekaligus inisialisasi
 //variabel answer bernilai 1 jika OK atau 0 jika CANCEL
 var answer = confirm("Apakah kamu ingin keluar dari
arispw.t35.com?")
 if (answer){
 alert("Terima Kasih!")
 window.close();//menutup browser
 }
 else{//!answer
 alert("Terima Kasih, untuk tidak keluar dari
arispw.t35.com!")
 }
 }
</script>
</head>
<body>
<form>
<input type="button" onclick="confirmation()" value="konfirmasi untuk
menutup arispw.t35.com">
</form>
</body>
</html>
```

**Kode Sumber 5.8.** Confirm

### 5.10. JS Prompt

Point utama dari prompt adalah digunakan untuk mengumpulkan informasi yang dimasukan oleh pengunjung halaman web dan kemudian untuk dilakukan proses lebih lanjut, seperti yang diberikan pada kode sumber 5.9.

**Nama File : jsLatih09.html**

```
<html>
<head>
<script type="text/javascript">
 //pendefinisian fungsi
 function prompter() {
 //deklarasi sekaligus inisialisasi
 var reply = prompt("Masukan Nama Anda?", "Default Nama (Karena
nama anda belum diisikan)")
 alert (reply)
 }
</script>
</head>
<body>
<input type="button" onclick="prompter()" value="Masukan Nama">
</body>
</html>
```

```

</script>
</head>
<body>
<input type="button" onclick="prompter()" value="Siapa Nama Anda!">
</body>
</html>

```

**Kode Sumber 5.9.** Prompt

### 5.10. JS Print

Fungsi JS print menjalankan operasi yang sama dengan icon gambar printer pada web browser. Fungsi JS print akan mengirimkan isi halaman web ke device printer, seperti yang diberikan pada kode sumber 5.10.

**Nama File : jsLatih10.html**

```

<html>
<head>
</head>
<body>
<script type="text/javascript">
 //deklarasi menggunakan var dan sekaligus inisialisasi
 var linebreak = "
"
 var my_var = "Hello world!"
 //deklarasi tanpa menggunakan var
 tanpaVar = "Variable Declaration without var"
 document.write(my_var)
 document.write(linebreak)
 document.write(tanpaVar)
 document.write(linebreak)
 //assignment
 my_var = "I am learning javascript!"
 document.write(my_var)
 document.write(linebreak)
 //assignment
 my_var = "Script is Finishing up..."
 document.write(my_var)
</script>
<form>
<!--fungsi print ditempatkan pada event onClick-->
<input type="button" value="Print Halaman ini"
onClick="window.print()" />
</form>
</body>
</html>

```

**Kode Sumber 5.10.** Print

### 5.11. JS Redirecting

Redirect digunakan untuk berpindah ke lokasi halaman web lain, dengan melakukan setting pada alamat URL halaman web tujuan. Properties JS yang digunakan untuk melakukan redirect adalah `window.location="alamatUrlTujuan"`, seperti yang diberikan pada kode sumber 5.11. (semua pengunjung web dilakukan redirect pada halaman `jsLatih11.html`). Redirect dapat dikombinasikan dengan pemberian waktu delay pada saat dilakukan perpindahan ke lokasi baru, seperti yang diberikan pada kode sumber 5.12. Untuk berpindah ke alamat domain (URL) baru sebaiknya menggunakan redirect, tetapi jika hanya merupakan link biasa sebaiknya jangan digunakan redirect tetapi menggunakan hyperlink.

**Nama File : jsLatih11.html**

```

<html>
<head>
<script type="text/javascript">
 alert("Membuka file jsLatih11.html karena efek

```

```
window.location=jsLatih10.html,Maka yang terbuka adalah file
jsLatih10.html")
 //melakukan redirect halaman jsLatih10.html
 window.location = "jsLatih10.html"
</script>
</head>
<body>
</body>
</html>
```

**Kode Sumber 5.11.** Redirect**Nama File : jsLatih12.html**

```
<html>
<head>
<script type="text/javascript">
 //pendefinisian fungsi
 function delayer(){
 //melakukan redirect halaman jsLatih10.html
 document.location = "jsLatih10.html"
 }
</script>
</head>
<!--setting lamanya waktu delay 5000 milisecond/5 second-->
<!--setTimeout adalah fungsi untuk mengatur waktu delay-->
<body onLoad="setTimeout('delayer()', 5000)">
<h2>Mempersiapkan untuk REDIRECT halaman jsLatih10.html!</h2>
<p>Halaman ini menggunakan waktu delay sebesar 5000 pada saat
dilakukan REDIRECT halaman jsLatih10.html!</p>
</body>
</html>
```

**Kode Sumber 5.12.** Redirect dengan Waktu Delay**5.12. JS Popup**

Popup digunakan pada saat menampilkan informasi tambahan yang dianggap penting, atau pada saat menampilkan membuka window baru tanpa menggunakan tag HTML <a>. Fungsi JS yang digunakan untuk membuka window baru adalah menggunakan properties window.open("namaDanUrlNewPage"), seperti yang diberikan pada kode sumber 5.13.

**Nama File : jsLatih13.html**

```
<html>
<head>
<script type="text/javascript">
 function myPopup() {
 //alamat URL yang akan di buka pada window baru
 window.open("jsLatih07.html")
 }
</script>
</head>
<body>
<form>
<input type="button" onClick="myPopup()" value="POPUP">
</form>
<p onClick="myPopup()">CLICK ME TOO!</p>
</body>
</html>
```

**Kode Sumber 5.13.** Popup untuk Membuka Window Baru

Beberapa argumen yang terdapat pada fungsi window.open() adalah :

- **Dependent** : sub window akan di tutup jika parent (window yang sedang terbuka) window juga di tutup
- **Fullscreen** : ditampilkan pada browser secara mode full screen
- **Height** : tinggi untuk window baru dalam pixels
- **Width** : lebar untuk window baru dalam pixels

- **Left** : pixel offset dari sisi kiri screen
- **Top** : pixel offset dari sisi atas screen
- **Resizable** : window dapat diubah ukurannya atau tidak (0 = tidak dapat diubah, dan 1 = dapat diubah)
- **Status** : menampilkan status bar atau tidak (0 = status bar tidak ditampilkan, dan 1 = status bar ditampilkan)

Dependent, fullscreen, resizable, and status dapat dilakukan set ON/OFF, jika bernilai 1 maka dalam kondisi ON, dan jika bernilai 0 maka dalam kondisi OFF.

#### Nama File : jsLatih14.html

```
<html>
<head>
<script type="text/javascript">
 function myPopup() {
 //alamat URL yang akan di buka pada window baru
 //menggunakan beberapa argumen
 window.open("jsLatih07.html","mywindow","status =
1,height=300, width=300,resizable=0")
 }
</script>
</head>
<body>
<form>
<input type="button" onClick="myPopup()" value="POPUP">
</form>
<p onClick="myPopup()">CLICK ME TOO!</p>
</body>
</html>
```

**Kode Sumber 5.14.** Popup untuk Membuka Window Baru Menggunakan Argumen

### 5.13. JS Date

Objek date digunakan pada saat dibutuhkan untuk menampilkan tanggal atau melakukan perhitungan lamanya waktu tertentu pada suatu halaman web. Objek date memiliki fungsi untuk menampilkan informasi tanggal sebagai berikut, seperti yang diberikan pada kode sumber 5.15. dan 5.16. :

- **getTime()** : menunjukkan angka millisecond dari 1/1/1970 @12:00 AM
- **getSecond()** : menunjukkan angka detik (0-59)
- **getMinutes()** : menunjukkan angka menit (0-59)
- **getHours()** : menunjukkan angka jam (0-23)
- **getDay()** : hari dalam satu minggu (0 = Sunday, ..., 6 = Saturday)
- **getDate()** : hari dalam satu bulan (0-31)
- **getMonth()** : menunjukkan angka bulan (0-11)
- **getFullYear()** : menunjukkan angka empat digit tahun (1970-9999)

#### Nama File : jsLatih15.html

```
<html>
<body>
<h4>Sekarang Tanggal :
<script type="text/javascript">
 //deklarasi sekaligus inisialisasi
 var currentTime = new Date()
 var day = currentTime.getDate()
 //month=month+1 , karena array month mulai dari 0
 var month = currentTime.getMonth()+1
 var year = currentTime.getFullYear()

 document.write(day + "/" + month + "/" + year)
 document.write("
" + "Menggunakan Fungsi date()" + " "+Date())
</script>
```



```
</h4>
</body>
</html>
```

**Kode Sumber 5.15.** Object Date**Nama File : jsLatih16.html**

```
<html>
<body>
<h4>Sekarang Jam :
<script type="text/javascript">
 //deklarasi sekaligus inisialisasi
 var currentTime = new Date()
 var hours = currentTime.getHours()
 var minutes = currentTime.getMinutes()

 if (minutes < 10){
 minutes = "0" + minutes
 }
 document.write(hours + ":" + minutes + " ")
 if(hours > 11){
 document.write("PM")
 }
 else {
 document.write("AM")
 }
</script>
</h4>
</body>
</html>
```

**Kode Sumber 5.16.** Variant Object Date**5.14. JS Form Validation**

Form validation adalah metoda yang digunakan untuk melakukan pengecekan informasi yang di-entry oleh pengguna halaman web sebelum dilakukan submit, dengan cara menampilkan alert untuk memberikan informasi kepada pengguna terhadap kesalahan entry data yang dilakukan. Misalnya pada suatu form isian semua fields harus diisi atau tidak boleh ada fields yang kosong, seperti yang diberikan pada kode sumber 5.17.

**Nama File : jsLatih17.html**

```
<html>
<head>
<script type="text/javascript">
 // fungsi validasi form
 function formValidation(form){
 if(notEmpty(form.req1)){
 if(notEmpty(form.req2)){
 return true;
 }
 }
 return false;
 } //end function formValidation

 // fungsi cek elemen tidak boleh kosong terhadap argumen elem
 function notEmpty(elem){
 //deklarasi sekaligus inisialisasi
 var str = elem.value;
 if(str.length == 0){
 alert("Semua yang bertanda(*) harus diisi / tidak boleh
kosong");
 return false;
 }
 else {(str.length != 0)
 return true;
 }
 } //end function notEmpty
</script>
</head>
```

```

<body>
<!--onSubmit dapat diganti dg menggunakan onBlur-->
<form name="example" onSubmit="return formValidation(this)">
Nama 1 : <input type="text" name="req1" />*

Nama 2 : <input type="text" name="req2" />*

<input type="submit" value="Check Validasi"/>
</form>
</body>
</html>

```

**Kode Sumber 5.17.** Validasi Form

### 5.15. JS Array

Array dapat dilihat sebagai banyak variabel yang disatukan. Anda dapat mengaksesnya melalui satu nama dan penomoran. Misalnya array diberi nama *names*, maka mengakses nama pertama dapat melalui *names[0]*, nama kedua dengan *names[1]* dan seterusnya. Setelah JS 1.1 (Netscape Navigator 3.0) Anda dapat menggunakan Array-object. Array baru dibuat dengan perintah *myArray = new Array()*. Harganya diisi dengan kode berikut:

```

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

```

JS array sangat fleksibel, ukurannya dapat ditentukan secara dinamis. Jika Anda menuliskan *myArray[99]="xyz"*, ukuran array akan menjadi seratus elemen. Tidak jadi masalah apakah yang disimpan adalah angka, string atau objek lain dalam array, seperti yang diberikan pada kode sumber 5.18.

#### Nama File : **jsLatih18.html**

```

<html>
<head>
</head>
<body>
<script type="text/javascript">
//deklarasi sekaligus inisialisasi
var myArray = new Array()
myArray[0]=17 // elemen ke-1
myArray[1]="Ambar" //elemen ke-2
myArray[2]="Miranthi" //elemen ke-3

document.write("myArray[0] adalah elemen ke-1 = " + myArray[0]+
"
")
document.write("myArray[1] adalah elemen ke-2 = " + myArray[1]+
"
")
document.write("myArray[2] adalah elemen ke-3 = " + myArray[2])
</script>
</body>
</html>

```

**Kode Sumber 5.18.** Array

### 5.16. JS Image

#### 5.16.1. Image pada Halaman Web

Image-object bisa menggantikan gambar image pada satu halaman web, dan memungkinkan juga untuk membuat animasi.

Bagaimana image dalam halaman web dapat diakses melalui array, Array disebut *images* yang merupakan property dari *document-object*. Setiap image dalam halaman web memiliki nomor. Image pertama bernomor 0, image kedua bernomor 1 dan seterusnya. Jadi Anda bisa mengakses image pertama melalui *document.images[0]*.

Setiap image dalam dokumen HTML dianggap sebagai sebuah *image-object*. *Image-object* tentunya memiliki property yang dapat diakses melalui JavaScript. Ukuran yang dimiliki image ditentukan dengan property `WIDTH` dan `HEIGHT`. Jadi `document.images[0].width` akan memberi lebar (dalam pixel) image pertama dalam halaman web yang ada.

Jika terdapat banyak image dalam satu halaman, tentu akan sulit untuk memberi nomor seluruh image yang ada. Dengan memberikan nama untuk setiap image akan jauh mempermudah hal ini.

```

```

Anda akan bisa mengaksesnya melalui `document.myImage` atau `document.images["myImage"]`, seperti yang diberikan pada kode sumber 5.19.

#### **Nama File : jsLatih19.html**

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

<script type="text/javascript">
 //melakukan penimpaan images/tilepic03.gif dg
 //images/logo.png dengan ukuran gambar tidak dapat dirubah
 document.myImage.src="images/logo.png"
</script>
</body>
</html>
```

**Kode Sumber 5.19.** Image pada Halaman Web

#### **5.16.2. Melakukan Load Image**

Untuk mengganti isi (file gambar) sebuah image pada web-page digunakan property `SRC`. Property `SRC` mewakili alamat dari file gambar yang ditampilkan. Dengan JavaScript 1.1 kita bisa mengisi alamat file gambar yang baru ke image yang telah di-load pada web-page. Hasilnya adalah gambar pada lokasi yang baru akan di-load.

Gambar baru ini akan menggantikan gambar lama. Perhatikan contoh berikut ini:

```

```

Image dengan file gambar `img1.gif` akan di-load dan bernama `myImage`. Baris kode berikut akan menggantikan gambar lama `img1.gif` dengan gambar baru `img2.gif`:

```
document.myImage.src="img2.src"
```

Gambar yang baru akan berukuran sama dengan gambar lama. Ukuran tempat gambar ditampilkan tidak bisa diubah, seperti pada kode sumber 5.19.

#### **5.16.3. Pre-Load Image**

Preloading image, yaitu me-load gambar lebih dahulu sebelum ditampilkan, sehingga bisa mempersingkat waktu untuk menampilkan gambar yang baru. Untuk ini Anda harus membuat *Image-object* tambahan. Lihat baris kode berikut :

```
hiddenImg=new Image()
hiddenImg.src=img3.gif
```

Baris pertama membuat sebuah *Image-object* dengan nama `hiddenImg`. Baris kedua mendefinisikan alamat gambar yang akan diwakili melalui object `hiddenImg`. Untuk menampilkan gambar yang baru ini gunakan baris berikut:

```
document.myImage.src=hiddenImg.src
```

Sekarang gambar diambil dari cache dan akan ditampilkan dengan jauh lebih cepat. Inilah yang disebut *preloading image*, seperti yang diberikan pada kode sumber 5.20. Tentu saja browser harus terlebih dahulu menyelesaikan pengambilan gambar yang dimaksud.

#### Nama File : jsLatih20.html

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

<script type="text/javascript">
 //image terlebih dahulu disimpan di cache sebelum di buka
 hiddenImg=new Image()
 hiddenImg.src="images/logo.png"
 document.myImage.src=hiddenImg.src
</script>
</body>
</html>
```

**Kode Sumber 5.20.** Pre-Load Image

### 5.17. Studi Kasus

#### 5.17.1. Pembuatan Tanggal

##### Nama File : jsLatih21.html

```
<!-- nama file : jsLatih21.html -->
<!-- Deskripsi : pembuatan tanggal sistem dalam bahasa indonesia -->

<html>
<head>
 <TITLE> Konversi Tanggal Dalam Bhs Indonesia </TITLE>
</head>
<body>
<script language="JavaScript">
 //deklarasi sekaligus inisialisasi
 var sekarang=new Date();
 var namaHari=new
 Array("Minggu","Senin","Selasa","Rabu","Kamis","Jumat","Sabtu");
 var namaBulan=new Array
 ("Januari","Februari","Maret","April","Mei","Juni","Juli","Agustus",
 "September","Oktober","November","Desember");
 var x=0;
 if(sekarang.getYear()<2000) {
 x=1900;
 }
 var Tahun=sekarang.getYear()+x
 document.write(sekarang.getDate()+"
 "+namaBulan[sekarang.getMonth()]+" "+Tahun)
</script>
</body>
</html>
```

**Kode Sumber 5.21.** Konversi Tanggal Dalam Bahasa Indonesia

#### 5.17.2. Multi Selection dan Menampilkan Source Code

##### Nama File : jsLatih22.html

```
<html>
<head>
<title>JavaScript Selection</title>
<script>

 // Fungsi untuk menampilkan semua list item yg dipilih
 function showSelections() {
 var i;
```

```

var str = "";

for (i = 0; i < document.forms[0].list.length; i++)
 if (document.forms[0].list[i].selected)
 str = str + document.forms[0].list[i].value + "\n";

alert(str);
}
</script>
</head>
<body>

<!-- A multiple selection list. -->
Contoh menggunakan multiple selection list:
<p>
Silahkan pilih satu atau lebih kategori berikut:
<p>
<form>
<select name="list" multiple size=4>
 <option value="news">News</option>
 <option value="sports">Sports</option>
 <option value="weather">Weather</option>
 <option value="entertainment">Entertainment</option>
 <option value="community">Community</option>
 <option value="events">Events</option>
 <option value="dining">Dining</option>
 <option value="home">Home and Garden</option>
 <option value="real estate">Real Estate</option>
</select>
<p>
<input type="button" value="Tampilkan Pilihan"
onClick="showSelections();">
<input type="reset" value="Hapus Pilihan">
</form>

<p>

<!-- menampilkan source code -->
<form onSubmit="location.href = 'view-source:' + location; return
false;">
<input type="submit" value="Tampilkan Source Code">
</form>

</body>
</html>

```

**Kode Sumber 5.22.** Multi Selection dan Menampilkan Source Code

### 5.17.3. Open Window Baru

**Nama File : jsLatih23.html**

```

<html>
<head>
<title>JavaScript Membuka Windows Baru</title>
<script>
 // Fungsi membuka windows baru dg menspesifikasikan URL
 function showAnswer(url) {
 window.open(url, "answer",
 "height=110,width=200,directories=no,location=no,menubar=no," +
 "resizeable=no,status=no,toolbar=no");
 return false;
 }
</script>
</head>
<body bgcolor="#ffffff">

<!-- Setiap link memanggil showAnswer() utk menampilkan windows baru
yang sesuai -->
Membuka Windows Baru
<p>
Pop Quiz:

```

```

<p>

 Mata Kuliah Yang sedang Anda Pelajari?

 <a href="#" onClick="return
showAnswer('answer1.html');">Answer
 <p>
 Mata Kuliah Yang paling Anda Sukai?

 <a href="#" onClick="return
showAnswer('answer1.html');">Answer
 <p>

<p>

<!-- Menampilkan source code -->
<form onSubmit="location.href = 'view-source:' + location; return
false;">
<input type="submit" value="View Source">
</form>

</body>
</html>

```

**Kode Sumber 5.23.** Open Window Baru**Nama File : answer1.html**

```

<html>
<head>
<title>JavaScript Tutorial</title>
</head>
<body bgcolor="#000000" text="#ffffff">
<center>
Internet
Programming
<p>
<form>
<input type=submit value="Close" onClick="window.close(); return
false;">
</form>
</center>
</body>
</html>

```

**Kode Sumber 5.24.** Window Baru**5.17.4. Object Window****Nama File : jsLatih24.html**

```

<html>
<head>
<title>JavaScript Windows Object</title>
<script>
 function alertTest() {
 alert("Press OK to continue.")
 document.forms["message"].text.value = "";
 return false;
 }
 function confirmTest() {
 if(confirm("Continue?"))
 document.forms["message"].text.value = "OK pressed.";
 else
 document.forms["message"].text.value = "Cancel pressed.";
 return false;
 }
 function promptTest() {
 var name;
 name = prompt("Please enter your name:", "");
 document.forms["message"].text.value = "Hello " + name + "!";
 return false;
 }

```

```
</script>
</head>
<body bgcolor="#ffffff">

Dialog windows.
<p>

<!-- Links menggunakan event handlers utk memanggil dialog function -->
Select a dialog box to display:
<p>

 Alert Dialog
 Confirm Dialog
 Prompt Dialog

<p>
<form name="message">
Result: <input name="text" type="input" value="">
</form>

<!-- menampilkan source code -->
<form onSubmit="location.href = 'view-source:' + location; return
false;">
<input type="submit" value="View Source">
</form>

</body>
</html>
```

**Kode Sumber 5.25.** Object Window

## BAB 6

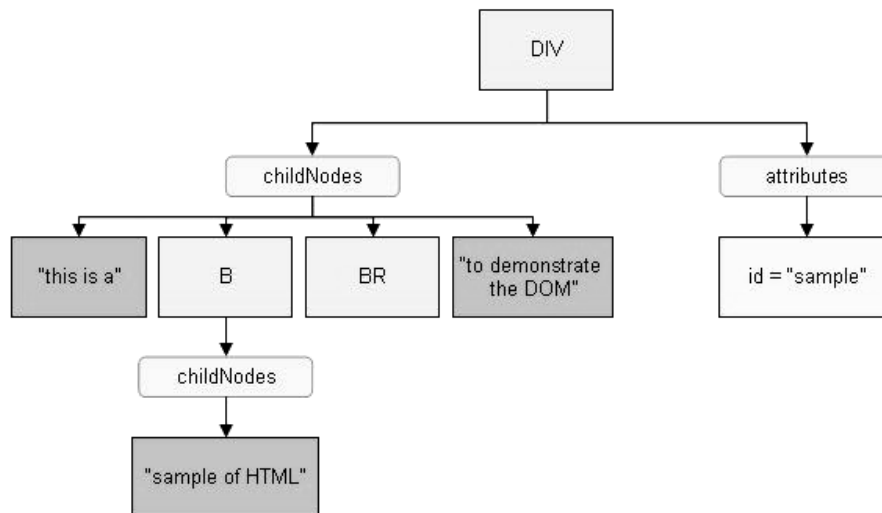
### Document Object Model (DOM)

#### 6.1 DOM dan Cara Modifikasinya dengan Javascript

Document Object Model (DOM) merupakan suatu struktur pohon(*tree*) yang mempresentasikan halaman HTML. Perhatikan contoh kode HTML berikut dan korespondensinya dengan pohon DOM.

```
<div id="sample">
This is a sample of HTML
 to demonstrate the DOM
</div>
```

Struktur DOM yang berkorespondensi:



Pada contoh di tersebut, tag DIV merupakan elemen akar(*root*), yang berisi tag-tag dan teks. Elemen DIV punya dua objek yang berasosiasi dengannya, atribut yang mereferensi pada seluruh tag (elemen) dan elemen anak(*childNodes*) yang mereferensi selainnya (berisi tag dan teks).

Penting untuk dipahami bahwa teks dianggap sebagai elemen pula. Suatu teks yang tidak mengandung elemen beserta elemen anak, maka dianggap sebagai sebuah elemen (sebagai contoh, "this is a" merupakan sebuah elemen).

Jadi bagaimana cara mengakses DOM dengan memakai Javascript? Baik IE dan Firefox mendukung fungsi DOM `getElementById`:

```
var elm=document.getElementById("sample");
// mengakses referensi pada elemen DIV
```

Fungsi `getElementById` menempatkan sebuah elemen dengan memanfaatkan atribut ID-nya (pada kasus ini, elemen ID pada DIV sebagai contoh). Dengan demikian, pastikanlah terdapat atribut ID pada elemen yang akan diakses dengan Javascript.

Perhatikan contoh pengaksesan DOM:

```
alert(elm.attributes.id.value); // memunculkan alert "sample"
alert(elm.childNodes[0].nodeType); // memunculkan alert 3 - TEKS
// NODE
alert(elm.childNodes[0].nodeValue); // memunculkan alert "this is a"
```



```

alert(elm.childNodes[1].nodeType); // memunculkan alert 1 - ELEMEN
NODE
alert(elm.childNodes[1].nodeValue); // memunculkan alert null -
elemen nodes don't have a value
alert(elm.childNodes[1].childNodes[0].nodeValue); //memunculkan
alert "sample of HTML"
elm.childNodes[0].nodeValue="what a "; // mengubah konten elemen
TEXT menjadi "what a "

```

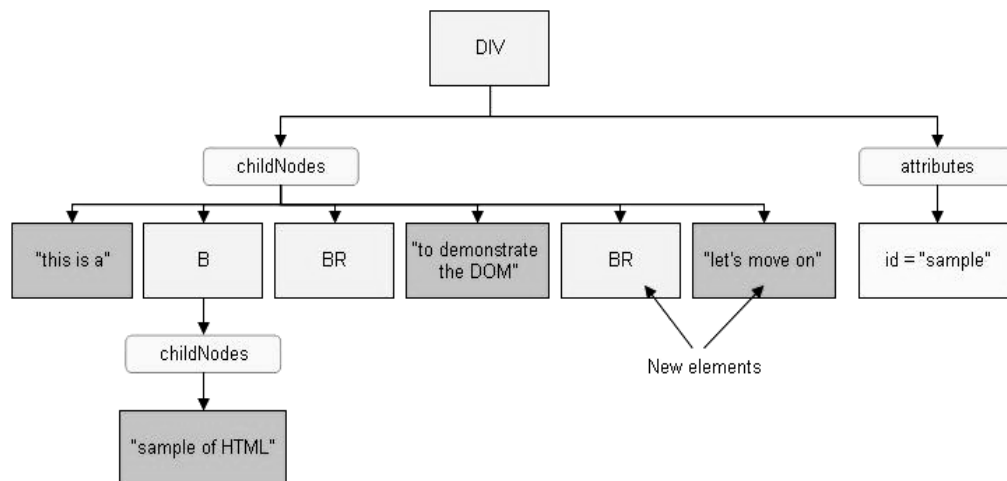
Kita dapat pula mengakses konten suatu elemen sebagai teks:

```

var elm=document.getElementById("sample");
alert(elm.innerHTML);
// memunculkan alert "This is a sample of HTML
 to
demonstrate the DOM"
elm.innerHTML+="
let's move on";

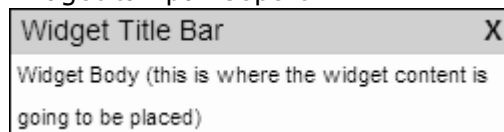
```

innerHTML memberikan akses langsung pada level teks ke dalam apapun yang terkandung dalam elemen. Kita dapat menggantikan atau menambahkan konten dan DOM akan membangunnya kembali secara *on the fly*. Berikut kondisi DOM setelah kode di atas dieksekusi:



## 6.2 Membuat Widget DOM secara Dinamis

Widget tampak seperti ini:



Kode HTML-nya adalah sebagai berikut:

```

<style>
.widget{
width: 250px;
min-height: 50px;
border: 1px solid #CACACA;
background-color: #FFFEEA;
}
</style>
...
<body>
<div id='id1' class='widget' style='width: 250px;'>
<div id='wdrag' style='width: 100%; height: 16; background-color:
#CCCCFF; border-bottom: 1px solid #CACACA;'>
<table id='d' style='font-family: arial; font-size: 8pt;'
width=100% cellpadding=0 cellspacing=0 border=0>
<tr>

```

```

 <td align=left><span id='id1_title' style='padding-left:
5px;'>Widget Title Bar</td>
 <td align=right>
 <a href='#' style='teks-decoration:
none;'>X</td>
 </tr>
</table>
</div>
<div id='id1_konten' style='padding: 5px; font-family: arial; font-
size: 8pt; width: 100%'>
 Widget Body (this is where the widget konten is going to be
placed)</div>
</div>

```

Widget memiliki dua bagian, Judul(*title bar*) dan badan(*body*). Judul berisi HREF('X') yang akan kita gunakan untuk membangkitkan event menutup widget. Berikut ini merupakan cara dinamis menambahkan widget.

```

...
<input type=button value="Add widget" onclick="addwidget()">
<div id="widget1"></div>
...
<script language="JavaScript">
<!--

var counter=10;

function addwidget(){
 var ref=document.getElemenById("widget1");
 // mencarikan tempat untuk widget
 var wid="id"+counter;
 // membuat ID yang unik untuk widget
 counter++;
 var wstr="<div id='"+wid+"' class='widget' style='top: 130; left:
300;
 width: 250px;'>" + "<div id='wdrag' style='cursor: move; width:
100%;
 height: 16; " + "background-color: #CCCCFF; border-bottom: 1px
solid
 #CACACA; '>" + "<table id='d' style='font-family: arial;
font-size: 8pt;' width=100% cellpadding=0 cellspacing=0
border=0>" +
 "<tr><td ALIGN=left><span id='id1_title' style='padding-left:
5px;'>
 widget Title Bar" + "</td><td align=right> " + "<a href=javascript:closewidget(
'" + wid + "') style='text-decoration: none;'>" +
 "X
 </td> </tr></table> </div>" + "<div id='id1_konten'
style='padding: 5px; font-family: arial; font-size: 8pt;
width: 100%'>" + "Widget Body (this is where the widget content
is going to be placed) </div></div>";
 ref.innerHTML+=wstr; // menempatkan widget pada layar
}

function closewidget(id){
 var widget1=document.getElemenById('widget1');
 var node=document.getElemenById(id);
 widget1.removeChild(node);
}
//-->
</script>
...

```

Pertama kita tambahkan sebuah tombol(button) pada layar, dan memanggil fungsi `addWidget` ketika diklik. Kemudian kita definisikan elemen DIV dengan ID `'widget1'` yang akan disajikan sebagai objek – inilah dimana widget akan ditempatkan.

Fungsi `addWidget` dalam Javascript amat sederhana: kita cukup mereferensi pada objek elemen DIV, membuat *string*(=`'untai','kalimat'`) yang berisi definisi widget (kita dapat menuliskan tag HTML sebagai sebuah string panjang) dan lalu kita menuliskannya pada elemen DIV, sehingga browser menambahkan sebuah widget pada layar.

Catatan penting:

1. Untuk mengubah pointer mouse menjadi bentuk `'drag'` ketika pengguna mendekatkannya di title bar (*hover event*), maka kita tambahkan: `'cursor: move'` dalam definisi CSS definition pada `<div id='wdrag' style='cursor: move; width: 100%;... elemen.`  
Terdapat banyak bentuk pointer mouse yang dapat dipilih.
2. Untuk memanggil fungsi `closewidget`, kita tambahkan: `href=javascript:closewidget('"+wid+"'')` pada HREF `'X'`.  
`wid` merupakan ID untuk widget, memungkinkan fungsi `closewidget` untuk menutup widget sesuai yang dimaksud.

Fungsi `closewidget` dalam Javascript menempatkan elemen DOM berkorespondensi dengan ID widget dan `'widget1'` (objek) serta menghapus elemen DOM dengan fungsi `removeChild`.

### 6.3 Menggeser DOM Widget

Untuk menggeser suatu objek terdapat langkah-langkah sebagai berikut:

1. Mendeteksi klik-kiri mouse pada suatu objek
2. Mendeteksi pergerakan mouse selama dalam kondisi klik-kiri
3. Menggeser objek mengikuti gerakan mouse
4. Mendeteksi lepas-klik mouse untuk menghentikan penggeseran objek

Perhatikan kode HTML berikut:

```
<div id='widget1' class='widget' style='position: absolute; left: 150; top: 150; width: 250; '>
<div style='cursor: move; width: 100%; height: 16; background-color: #CCCCFF; border-bottom: 1px solid #CACACA; '>
 <table style='font-family: arial; font-size: 8pt;' width=100% cellpadding=0 cellspacing=0 border=0>
 <tr>
 <td ALIGN=left id='titlebar' style='padding-left: 5px;'>widget Title Bar</td>
 <td align=right style='padding-right: 5px;'> X</td>
 </tr>
 </table>
</div>
<div id='' style='padding: 5px; font-family: arial; font-size: 8pt; width: 100%'>
 widget Body (this is where the widget konten is going to be placed)</div>
</div>
```

Buat widget pada layar, sebagaimana yang pada contoh sebelumnya. Perlu diperhatikan bahwa agar dapat memposisikan secara bebas widget dimanapun pada layar, kita perlu menggunakan `position: absolute` dalam definisi CSS. Sekarang kita dapat memerintahkan browser dimana harus meletakkan widget, dan kita menggunakan `left: 150; top: 150;` untuk menempatkan widget pada 150 pixel dari kiri ( $x=150$ ) dan 150 pixel dari atas ( $y=150$ ).

Perhatikan kode Javascript berikut:

```
var x,y;
var tx, ty;
var isDrag=false;
var firefox=document.getElementById&&!document.all;
document.onmousemove=mouseMove;
document.onmousedown=selectMouse;
document.onmouseup=fungsi() {
 isDrag=false;
}

fungsi selectMouse(e) {
 if (firefox) {
 // memeriksa bahwa user mengklik elemen title bar
 var p=e.target;
 if(p.attributes['id']&&p.attributes['id'].value=="titlebar")
 {
 isDrag=true;
 x=e.clientX;
 y=e.clientY;
 // mengambil nilai koordinat widget
 tx=parseInt(document.getElementById('widget1').style.left);
 ty=parseInt(document.getElementById('widget1').style.top);
 }
 }
 else { //IE
 // memeriksa bahwa user mengklik elemen title bar
 var p=event.srcElement;
 if (p.attributes['id']&&p.attributes['id'].value=="titlebar")
 {
 isDrag=true;
 x=event.clientX;
 y=event.clientY;
 // mengambil nilai koordinat widget
 tx=parseInt(document.getElementById('widget1').style.left);
 ty=parseInt(document.getElementById('widget1').style.top);
 }
 }
}

fungsi mouseMove(e) {
 if (isDrag) { // geser box widget menurut gerakan mouse
 var box=document.getElementById("widget1");
 if (firefox) {
 box.style.left = e.clientX + (tx - x);
 box.style.top = e.clientY + (ty - y);
 }
 else { //IE
 box.style.left = event.clientX + (tx - x);
 box.style.top = event.clientY + (ty - y);
 }
 }
}
```

`isDrag` bernilai `true` ketika user menggeser widget (ketika user mengklik mouse dan belum melepaskannya). Setiap kali mouse berpindah fungsi `mousemove` akan menanganinya, ketika mouse diklik fungsi `selectMouse` akan menanganinya, dan ketika mouse dilepaskan maka `isDrag` diset bernilai `false`.

`x,y` menyimpan nilai koordinat mouse ketika diklik, `tx,ty` menyimpan nilai koordinat widget ketika mouse diklik (`x=left`, `y=top`). Hal ini memungkinkan kita memperhitungkan keseimbangan gerakan mouse jika dibandingkan dengan gerakan widget agar pergeseran berlaku secara tepat.

Ketika terjadi event mouse diklik, fungsi `selectMouse` dipanggil (`document.onmousedown=selectMouse`). Fungsi `selectMouse` akan memeriksa bahwa mouse diklik pada titlebar widget. Pada Firefox hal ini dilakukan dengan mengakses atribut `target` dari objek event (menyimpan referensi kepada elemen DOM yang diklik) dan kemudian memeriksa bahwa elemen memiliki atribut `ID` dengan value `'titlebar'`. Pada IE hal ini dilakukan dengan mengecek atribut `srcElement` dari objek event global dan kemudian menjalani proses yang sama.

Jika user melakukan klik pada titlebar, maka *dragflag* diset menjadi bernilai `true` (`isDrag=true`) dan menyimpan koordinat klik mouse pada `x,y`. Kita kemudian menyimpan koordinat widget atribut `DIV widget1` dengan mencacah `left`, `top` pada atribut CSS dan menyimpan nilainya pada variabel `tx,ty`.

Ketika mouse berpindah, terlebih dahulu dicek apakah `isDrag` dalam kondisi `true` (dengan kata lain, user masih menahan klik-kiri mouse). Jika demikian, kita mengeset atribut CSS `left` dan `top` pada `DIV widget` untuk meng-update koordinat mouse dengan mengambil nilai koordinat mouse saat ini dan menambahkan selisih antara koordinat mouse dan sudut kiri atas(top left corner) dari widget.

## Bab 7

### Dynamic HTML (DHTML)

#### 7.1 Pengantar

DHTML adalah HTML dengan tambahan format-format khusus dan script untuk membuat halaman-halaman Web lebih interaktif. Meski baik Microsoft Internet Explorer 4.0 dan Netscape Communicator 4.0 mendukung fitur-fitur DHTML, kedua cara yang digunakan tidak kompatibel, sehingga sebuah halaman yang ditulis dengan DHTML mungkin terlihat baik pada salah satu browser tetapi tidak bisa dibuka dengan baik pada browser lainnya.

DHTML merupakan suatu kumpulan teknologi yang digunakan bersama-sama untuk menciptakan websites yang interaktif dan animatif dengan memanfaatkan kombinasi dari *Markup-Language* (seperti HTML), bahasa *scripting* klien (seperti JavaScript), bahasa pendefinisi tampilan (CSS), serta Document Object Model.

Secara umum, suatu konsep web dinamis berarti halaman web yang di-*generate* secara berbeda untuk masing-masing pengguna, memuat peristiwa, atau nilai variabel yang spesifik. Termasuk diantaranya halaman yang tercipta oleh script klien, dan di sisi lain oleh script server (semisal PHP atau Perl) dimana web server meng-*generate* isi sebelum dikirimkan kepada komputer klien.

DHTML sering digunakan untuk membuat tombol *rollover* atau menu drop-down pada halaman web.

Kegunaan lain yang tidak umum yaitu untuk membuat browser berbasis *action games*. Selama akhir dekade 1990-an hingga awal dekade 2000-an, sejumlah game diciptakan dengan DHTML. Namun perbedaan diantara masing-masing browser membuat hal ini sulit diterapkan; banyak teknik yang harus diimplementasikan dalam kode program supaya game dapat berjalan pada multi platform. Browser-browser baru telah berkumpul untuk mengikuti standardisasi web, dibuat dengan desain game DHTML. Game dapat dijalankan pada kebanyakan browser serta terhubung pada *Widgets for Mac OS X* dan *Gadgets for Windows Vista*, yang berbasis pada kode DHTML.

Beberapa kelemahan DHTML yaitu sulitnya dalam proses pengembangan dan *debug*(testing) oleh derajat perubahan dari dukungan browser terhadap teknologi yang bersangkutan, dan variasi ukuran layar monitor menjadikan tampilan final hanya dapat dinikmati jika menemui kondisi *fine-tuned*(ukuran layar spesifik dan dukungan teknologi browser). Pengembangan bagi browser-browser terbaru, seperti Internet Explorer 5.0+, Mozilla Firefox 2.0+, and Opera 7.0+, terbantu oleh pembagiaksesan(*shared*) Document Object Model.

Secara khusus suatu halaman web DHTML disiapkan dengan cara berikut:

```
<!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>DHTML example</title>
 <script type="text/javascript">
 function init() {
 myObj = document.getElementById("navigation");
 // more code
 }
 window.onload=init;
 </script>
 </head>
 <body>
 <div id="navigation"></div>
 </body>
</html>
```

Kode berikut mengilustrasikan fungsi-fungsi yang sering dipakai. Bagian tambahan halaman web hanya akan ditampilkan jika pengguna memintanya. Dalam e-learning, fungsi-fungsi dapat digunakan untuk menampilkan petunjuk tambahan yang pada awalnya tidak ditampilkan.

```
<!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" xml:lang="en">
 <head>
 <title>Test</title>
 <style type="text/css">
 h2 {background-color: lightblue; width: 100%}
 a {font-size: larger; background-color: goldenrod}
 a:hover {background-color: gold}
 #example1 {display: none; margin: 3%; padding: 4%;
background-color: limegreen}
 </style>
 <script type="text/javascript">
 function changeDisplayState (id) {
 d=document.getElementById("showhide");
 e=document.getElementById(id);
 if (e.style.display == 'none' || e.style.display == "") {
 e.style.display = 'block';
 d.innerHTML = 'Hide example';
 } else {
 e.style.display = 'none';
 d.innerHTML = 'Show example';
 }
 }
 </script>
 </head>
 <body>
 <h2>How to use a DOM function</h2>
 <div><a id="showhide"
href="javascript:changeDisplayState('example1')">Show example</div>
 <div id="example1">
 This is the example.
 (Additional information, which is only displayed on request)...
 </div>
 <div>The general text continues...</div>
 </body>
</html>
```

## 7.2. DHTML sebagai Perpaduan Teknologi Web

DHTML mampu mengerjakan hal-hal tertentu yang tersembunyi seperti mengirimkan *cookies* (berkas yang menyimpan sesi autentifikasi dan otorisasi) atau mendeteksi informasi-informasi dalam komputer pengunjung. Ide dasarnya adalah suatu halaman web yang cerdas namun sederhana dengan memanfaatkan teknologi server dan script sekaligus.

### XHTML

XHTML atau HTML 4.x digunakan untuk menciptakan halaman itu sendiri dan membangun unsur-unsur untuk CSS dan DOM untuk bekerja terpasang. Tidak ada yang khusus tentang XHTML untuk DHTML, tetapi mempunyai XHTML yang valid.

### Client-side Script

Penulisan script menggunakan JavaScript atau ActiveX yang merupakan bahasa pemrograman paling umum yang digunakan untuk mengaktifkan DHTML. Disamping itu menggunakan suatu script untuk mengendalikan object itu untuk menetapkan DOM.

### CSS (Cascading StyleSheet)

Ini merupakan bagian dari DHTML (Dynamic HTML), di mana kita dapat menentukan rancangan/desain layout tampilan dari HTML yang kita buat, dari jenis tulisan, warna background, dan sebagainya. Semua di control dan disetting di CSS, memudahkan dalam menjaga konsistensi rancangan web kita.

Contoh untuk style text :

```
/* CSS Document */
Body {
 padding: 0;
 margin: 0;
 text-align:center;
}
#container {
 width:800px;
 text-align: left;
 position: relative;
}
// pembagian halaman untuk head
#head {
 width: 100%;
 background-image:url(image/head.jpg);
 height: 140px;
}
// pembagian halaman untuk link menu
#menu {
 background-color:#bde9fb;
 image-align: right;
 height: 35px;
 width: 100%;
}
// pembagian halaman untuk content
#content {
 background-color: #D6D3D6;
 background-image:url(image/semut.jpg);
 background-position: right bottom;
 background-repeat: no-repeat;
 height: 500px;
 width: 780px;
 padding: 10px 10px 10px 10px;
}
```

Setiap menggunakan CSS, tiap bagian dipanggil dengan tag <div id=""> dan ditutup dengan tag </div>.

### DOM

DOM adalah sesuatu yang memungkinkan kita untuk mengakses semua bagian dari Halaman web untuk mengubah isinya dengan DHTML. Tiap-Tiap bagian dari suatu Halaman web ditetapkan oleh DOM dan penggunaan yang konsisten menamai konvensi yang memungkinkan kita dapat mengakses halaman itu dan merubah isinya.

### XML (eXtensible Markup Language)

Kalau HTML itu tag-tagnya dapat dimengerti oleh browser dan mendefinisikan format tampilan, XML memiliki tag-tag bebas dan dapat didefinisikan sendiri oleh pengguna, di mana digunakan untuk merepresentasikan suatu data.

Contoh style untuk membuat halaman web :



## Bab 8

### Extensible Markup Language (XML)

#### 8.1. Pengertian XML

XML kependekan dari eXtensible Markup Language

XML adalah markup language / bahasa markup seperti HTML

XML didesain untuk menggambarkan data.

XML tags tidak didefinisikan dalam XML sendiri.

XML menggunakan DTD (Document Type Definition) untuk menggambarkan data

XML dengan DTD didesain untuk menggambarkan diri sendiri/self describing.

Contoh DTD :

```
<!DOCTYPE booklist
[
 <!ELEMENT booklist (book*)>
 <!ELEMENT book (title,authors,publisher,pages?,isbn,price+)>
 <!ELEMENT authors (author+)>
 <!ELEMENT title (#PCDATA)>
 <!ELEMENT author (#PCDATA)>
 <!ELEMENT publisher (#PCDATA)>
 <!ELEMENT pages (#PCDATA)>
 <!ELEMENT isbn (#PCDATA)>
 <!ELEMENT price (#PCDATA)>
 <!ATTLIST price currency CDATA #REQUIRED>
]>
```

#### 8.2. Tentang XML

XML dapat digunakan secara langsung di Internet, mendukung berbagai aplikasi secara luas, serta kompatibel dengan SGML. Dengan demikian XML mudah digunakan untuk membuat program yang melakukan proses parsing dokumen XML.

##### XML dan HTML

XML didesain untuk membawa/carry data

XML tidak untuk menggantikan HTML XML and HTML didesain dengan tujuan berbeda XML didesain untuk menggambarkan data dan fokusnya pada apakah data itu?. HTML didesain untuk menampilkan data, dan fokus pada bagaimana data tampak HTML mengenai menampilkan informasi, XML mengenai menggambarkan informasi.

##### XML tidak didesain untuk melakukan apapun

Mungkin hal ini sukar dimengerti tetapi XML adalah cara untuk membangun struktur serta menyimpan dan mengirim informasi. Contoh berikut adalah sebuah pesan untuk Tove dari Jani, disimpan sebagai dokumen XML:

```
<note>
 <to>Tove</to>
 <from>Jani</from>
 <heading>Reminder</heading>
 <body>Don't forget me this weekend!</body>
</note>
```

Pesan di atas memiliki header dan body juga memiliki informasi pengirim dan penerima, tetapi dokumen XML di atas tidak melakukan apapun. Semua di atas hanyalah sebatas informasi dan tidak melakukan apapun. Harus ada software untuk mengirimnya, menerima atau menampilkannya.

#### XML sebagai pelengkap HTML

Sangat penting untuk dimengerti bahwa XML tidak untuk menggantikan HTML. Di masa depan dapat dikatakan bahwa XML digunakan untuk menggambarkan data (definisi data) dan HTML digunakan untuk memformat dan menampilkan data tersebut. Dengan kata lain, XML adalah tool yang independen untuk mentransmisikan informasi yang memiliki sifat *cross platform* baik software maupun hardware.

#### XML akan ada di mana-mana

HTML merupakan fondasi dari Web sedangkan XML akan menjadi alat utama untuk manipulasi data dan transmisi data

### **8.3. Penggunaan XML**

#### XML dapat memisahkan data dari HTML

Dengan XML data dapat tersimpan terpisah dari HTML sehingga kita dapat berkonsentrasi menggunakan HTML hanya untuk layout data dan tampilannya sehingga kita tidak khawatir perubahan data akan mengubah HTML

#### XML digunakan untuk pertukaran data

Dengan XML, data dapat dipertukarkan antara sistem yang tidak kompatibel. Dalam kasus nyata, sistem komputer dan database menyimpan data dalam format yang tidak kompatibel satu sama lain. Hal itu menjadi salah satu hambatan bagi para developer untuk mempertukarkan data di Internet. Dengan mengkonversikan data ke XML kita dapat mengurangi kompleksitas masalah dan membuat data yang dapat dibaca oleh berbagai tipe aplikasi.

#### XML dan B2B

Dengan XML, informasi finansial dapat dipertukarkan melalui internet sehingga pertukaran aplikasi B2B akan menjadi lebih baik.

#### XML untuk Membagi Data

XML biasanya disimpan sebagai berkas teks biasa (*plaintext*) namun dapat menyatakan pembagian data. Dengan demikian, tersedia jalan yang independen untuk membagi data. Hal ini memudahkan untuk membuat data yang dapat digunakan oleh aplikasi-aplikasi berbeda, selain itu juga memudahkan untuk mengekspansi dan mengupgrade ke sistem operasi baru, server baru, aplikasi baru dan browsers baru.

#### XML untuk Menyimpan Data

Dengan XML, file teks biasa dapat untuk menyimpan data.

XML juga dapat digunakan untuk menyimpan data dalam file maupun database. Aplikasi tertentu dapat dibuat untuk menulis dan mengambil informasi dari tempat penyimpanan dan aplikasi umum dapat digunakan untuk menampilkan data.

#### XML Membuat Data Lebih Berguna

Dengan XML, data dapat digunakan oleh lebih banyak orang karena bersifat independen dan *cross platform* sehingga data dapat diakses oleh banyak orang dan tidak tergantung pada standar browser maupun HTML tertentu. Klien dan aplikasinya dapat mengakses file XML sebagai sumber data seperti mengakses database.

#### XML sebagai induk dari WAP dan WML

*Wireless Markup Language* digunakan untuk memarkup aplikasi Internet untuk peralatan mobile semacam telepon selular yang ditulis dalam XML. Semua aplikasi akan menyimpan datanya dalam XML, sehingga berbagai aplikasi dapat membaca data masing-masing dalam format teks murni tanpa adanya utiliti untuk konversi.

### **8.4. Sintaks XML**

Sintaks XML sangat sederhana dan sangat terbatas. Aturannya sangat mudah dipelajari dan sangat mudah digunakan. Karena itu, membuat software yang dapat membaca dan memanipulasi XML sangat mudah dilakukan

Contoh Dokumen XML

Dokumen XML yang menggambarkan dirinya sendiri dengan sintak sederhana:

```
<?xml version="1.0"?>
<note>
 <to>Tove</to>
 <from>Jani</from>
 <heading>Reminder</heading>
 <body>Don't forget me this weekend!</body>
</note>
```

Baris pertama dokumen merupakan deklarasi XML adalah untuk mendefinisikan versi XML pada dokumen, yaitu versi 1.0. Baris berikutnya menggambarkan struktur elemen dari dokumen ini

#### Tentang Struktur XML

Semua elemen XML harus memiliki tag penutup

Dalam XML adalah ilegal tidak memiliki tag penutup

Dalam HTML beberapa elements tidak harus memiliki tag penutup, seperti contoh di bawah ini:

```
<p>This is a paragraph
<p>This is another paragraph
```

Dalam XML contoh diatas tidak benar dan yang benar adalah seperti dibawah ini :

```
<p>This is a paragraph</p>
<p>This is another paragraph</p>
```

Catatan : Dari contoh sebelumnya deklarasi XML tidak memiliki tag penutup, hal ini tidak salah. Deklarasi tidak merupakan bagian dari dokumen XML itu sendiri, ini bukan elemen XML dan tidak harus memiliki tag penutup.

#### Case sensitive

Hal ini tidak seperti HTML yang case insensitive. Dalam XML , tag <Note> berbeda dari tag <note> Jadi tag awal dan akhir harus ditulis dalam case yang sama

```
<Message>This is incorrect</message>
<message>This is correct</message>
```

#### Semua elemen XML harus memiliki struktur yang benar

Dalam HTML beberapa element tidak harus memiliki struktur yang urut (memungkinkan *overlapping*), seperti di bawah ini

```
<i>This text is bold and italic</i>
```

Tetapi dalam XML harus memiliki urutan yang benar

```
<i>This text is bold and italic</i>
```

Semua dokumen XML harus memiliki tag akar

Tag pertama dalam dokumen XML adalah tag utama

Elemen dalam XML dapat memiliki sub elemen

```
<root>
 <child>
 <subchild>.....</subchild>
 </child>
</root>
```

Nilai dari atribut harus diberi tanda petik

Dalam XML adalah ilegal bila tidak memberi tanda petik pada nilai dari atribut

Bandingkan contoh yang salah dan contoh yang benar berikut ini:

```
<?xml version="1.0"?>
 <note date=12/11/99>
 <to>Tove</to>
 <from>Jani</from>
 <heading>Reminder</heading>
 <body>Don't forget me this weekend!</body>
 </note>
```

```
<?xml version="1.0"?>
<note date="12/11/99">
 <to>Tove</to>
 <from>Jani</from>
 <heading>Reminder</heading>
 <body>Don't forget me this weekend!</body>
</note>
```

\*Perhatikan atribut date pada tag note

Dalam XML, Spasi Kosong Diperhatikan

Dalam XML, spasi dalam dokumen tidak diabaikan

Hal ini tidak seperti HTML yang mengabaikan spasi. Dalam HTML, kalimat seperti ini:

Hello            my name is Tove,

akan ditampilkan seperti ini:

Hello my name is Tove,

Element XML adalah Ekstensibel

Dokumen XML dapat dikembangkan untuk membawa informasi lebih

Perhatikan contoh dokumen XML ini:

```
<?xml version="1.0"?>
<note>
 <to>Tove</to>
 <from>Jani</from>
 <heading>Reminder</heading>
 <body>Don't forget me this weekend!</body>
</note>
```

Misalkan kita membuat aplikasi yang mengeluarkan elemen <to>, <from>, dan <body> dari dokumen XML dapat menghasilkan output ini:

MESSAGE To: Tove From: Jani Don't forget me this weekend!

Elemen-elemen XML memiliki hubungan

Elemen memiliki hubungan semacam orangtua dan anak

Untuk memahami terminologi XML kita harus mengetahui hubungan antara penamaan elemen XML dan bagaimana isi elemen digambarkan

Misalnya di bawah ini adalah deskripsi dari suatu buku

Book Title: My First XML  
Chapter 1: Introduction to XML what is HTML what is XML  
Chapter 2: XML Syntax Elements must have a closing tag  
Elements must be correctly nested

Dalam XML data buku tersebut dapat dideskripsikan dalam format sebagai berikut:

```
<book>
 <title>My First XML</title>
 <prod id="33-657" media="paper"></prod>
 <chapter>Introduction to XML
 <para>what is HTML</para>
 <para>what is XML</para>
 </chapter>
 <chapter>XML Syntax
 <para>Elements must have a closing tag</para>
 <para>Elements must be properly nested</para>
 </chapter>
</book>
```

Book adalah root element. Title dan chapter adalah child elements dari book. Book adalah parent element dari title dan chapter. Title dan chapter adalah siblings (atau sister elements) karena mereka mempunyai parent yang sama.

**LAMPIRAN HTML****1. Symbols HTML**

<b>Symbols</b>	<b>Used</b>
URL	URL of an external file (or just file name if in the same directory)
?	Arbitrary number (i.e. <H?> means <H1>, <H2>, <H3>, etc.)
%	Arbitrary percentage (i.e. <HR WIDTH="%"> means <HR WIDTH="50%">, etc.)
***	Arbitrary text (i.e. ALT="***" means fill in with text)
\$\$\$\$\$	Arbitrary hex (i.e. BGCOLOR="#\$\$\$\$\$" means BGCOLOR="#00FF1C", etc.)
:::	Arbitrary date (i.e. DATETIME=":::" means "1994-11-05T08:15:30" etc.)
@	Email address (i.e. "mailto:@" means "mailto:kevin@werbach.com" etc.)
, , ,	Comma-delimited (i.e. COORDS=",,, " means COORDS="0,0,50,50", etc.)
	Alternatives (i.e. ALIGN=LEFT RIGHT CENTER means pick one of these)

**2. Komentar**

<b>Tags</b>	<b>Used</b>
<!-- . . . -->	Untuk komentar
< COMMENT > . . . </COMMENT>	Untuk komentar

**3. Structure Tags**

<b>Tag</b>	<b>Attribute</b>	<b>Function</b>
<HTML> . . . </HTML>		Encloses the entire HTML document
<HEAD> . . . </HEAD>		Encloses the head of the HTML document.
<BODY> . . . </BODY>		Encloses the body (text and tags) of the HTML document.
	BACKGROUND="..."	The name or URL of the image to tile on the page background.
	BGCOLOR="..."	The color of the page background.
	TEXT="..."	The color of the page's text.
	LINK="..."	The color of unfollowed links.
	ALINK="..."	The color of activated links.
	VLINK="..."	The color of followed links.
	BGPROPERTIES="..."(MS)	Properties of background image. Currently allows only the value FIXED, which prevents the background image from scrolling.
	TOPMARGIN="..."(MS)	Top margin of the page, in pixels.
	BOTTOMMARGIN="..."(MS)	Bottom margin of the page, in pixels.
<BASE>		Indicates the full URL of the current document. This optional tag is used within <HEAD>.
	HREF="..."	The full URL of this document.
<ISINDEX>		Indicates that this document is a gateway script that allows searches.
	PROMPT="..."	The prompt for the search field.
	ACTION="..."	Gateway program to which the search string should be passed.
<LINK>		Indicates a link between this document and some other document. Generally used only by HTML-generating tools. <LINK> represents a link from this entire document to another, as opposed to <A>, which can create multiple links in the document. Not commonly used.
	HREF="..."	The URL of the document to call when the link is activated.

	NAME="..."	If the document is to be considered an anchor, the name of that anchor.
	REL="..."	The relationship between the linked-to document and the current document; for example, "TOC" or "Glossary".
	REV="..."	A reverse relationship between the current document and the linked-to document.
	URN="..."	A Uniform Resource Number (URN), a unique identifier different from the URL in HREF.
	TITLE="..."	The title of the linked-to document.
	METHODS="..."	The method with which the document is to be retrieved; for example, FTP, Gopher, and so on.
<META>		Indicates meta-information about this document (information about the document itself); for example, keywords for search engines, special HTTP headers to be used for retrieving this document, expiration dates, and so on. Meta-information is usually in the form of a key/value pair. Used in the document <HEAD>.
	HTTP-EQUIV="..."	Creates a new HTTP header field with the same name as the attribute's value; for example, HTTP-EQUIV="Expires". The value of that header is specified by the CONTENT attribute.
	NAME="..."	If meta-data is usually in the form of key/value pairs, NAME indicates the key; for example, Author or ID.
	CONTENT="..."	The content of the key/value pair (or of the HTTP header indicated by HTTP-EQUIV).
<NEXTID>		Indicates the "next" document to this one (as might be defined by a tool to manage HTML documents in series). <NEXTID> is considered obsolete.

#### 4. Headings dan Title

Tag	Attribute	Function
<H1>...</H1>		A first-level heading.
<H2>...</H2>		A second-level heading.
<H3>...</H3>		A third-level heading.
<H4>...</H4>		A fourth-level heading.
<H5>...</H5>		A fifth-level heading.
<H6>...</H6>		A sixth-level heading.
<TITLE>...</TITLE>		Indicates the title of the document. Used within <HEAD>.
All heading tags accept the following attribute	ALIGN="..."	Possible values are CENTER, LEFT, and RIGHT.

#### 5. Paragraphs dan Regions

Tag	Attribute	Function
<P>...</P>		A plain paragraph. The closing tag (</P>) is optional.
	ALIGN="..."	Align text to CENTER, LEFT, or RIGHT.

<DIV>...</DIV>		A region of text to be formatted.
	ALIGN="..."	Align text to CENTER, LEFT, or RIGHT.

## 6. Links

Tag	Attribute	Function
<A>...</A>		With the HREF attribute, creates a link to another document or anchor; with the NAME attribute, creates an anchor that can be linked to.
	HREF="..."	The URL of the document to be called when the link is activated.
	NAME="..."	The name of the anchor.
	REL="..."	The relationship between the linked-to document and the current document; for example, "TOC" or "Glossary". REL="..." is not commonly used.
	REV="..."	A reverse relationship between the current document and the linked-to document (not commonly used).
	URN="..."	A Uniform Resource Number (URN), a unique identifier different from the URL in HREF (not commonly used).
	TITLE="..."	The title of the linked-to document (not commonly used).
	METHODS="..."	The method with which the document is to be retrieved; for example, FTP, Gopher, and so on (not commonly used).
	TARGET="..."	The name of a frame that the linked document should appear in.

## 7. Lists

Tag	Attribute	Function
<OL>...</OL>		An ordered (numbered) list.
	TYPE="..."	The type of numerals to label the list. Possible values are A, a, I, i, and 1.
	START="..."	The value with which to start this list.
<UL>...</UL>		An unordered (bulleted) list.
	TYPE="..."	The bullet dingbat to use to mark list items. Possible values are DISC, CIRCLE (or ROUND), and SQUARE.
<MENU>...</MENU>		A menu list of items.
<DIR>...</DIR>		A directory listing; items are generally smaller than 20 characters.
<LI>		A list item for use with <OL>, <UL>, <MENU>, or <DIR>.
	TYPE="..."	The type of bullet or number to label this item with. Possible values are DISC, CIRCLE (or ROUND) SQUARE, A, a, I, i, and 1.
	VALUE="..."	The numeric value this list item should have (affects this item and all below it in <OL> lists).
<DL>...</DL>		A definition or glossary list.
	COMPACT	The COMPACT attribute specifies a formatting that takes less whitespace to present.
<DT>		A definition term, as part of a definition list.
<DD>		The corresponding definition to a definition term, as part of a definition list.

## 8. Character Formatting

Tag	Attribute	Function
<EM>...</EM>		Emphasis (usually italic).
<STRONG>...</STRONG>		Stronger emphasis (usually bold).
<CODE>...</CODE>		Code sample (usually Courier).
<KBD>...</KBD>		Text to be typed (usually Courier).
<VAR>...</VAR>		A variable or placeholder for some other value.



<SAMP>...</SAMP>		Sample text (not commonly used).
<DFN>...</DFN>		A definition of a term.
<CITE>...</CITE>		A citation.
<B>...</B>		Boldface text.
<I>...</I>		Italic text.
<TT>...</TT>		Typewriter (monospaced) font.
<PRE>...</PRE>		Preformatted text (exact line endings and spacing will be preserved--usually rendered in a monospaced font).
<BIG>...</BIG>		Text is slightly larger than normal.
<SMALL>...</SMALL>		Text is slightly smaller than normal.
<SUB>...</SUB>		Subscript.
<SUP>...</SUP>		Superscript.
<STRIKE>...</STRIKE>		Puts a strikethrough line in text.

## 9. Other Elements

Tag	Attribute	Function
<HR>		A horizontal rule line.
	SIZE="..."	The thickness of the rule, in pixels.
	WIDTH="..."	The width of the rule, in pixels or as a percentage of the document width.
	ALIGN="..."	How the rule line will be aligned on the page. Possible values are LEFT, RIGHT, and CENTER.
	NOSHADE	Causes the rule line to be drawn as a solid line instead of a transparent bevel.
	COLOR="..." (MS)	Color of the horizontal rule.
 		A line break.
	CLEAR="..."	Causes the text to stop flowing around any images. Possible values are RIGHT, LEFT, and ALL.
<NOBR>...</NOBR>		Causes the enclosed text not to wrap at the edge of the page.
<WBR>		Wraps the text at this point only if necessary.
<BLOCKQUOTE>...</BLOCKQUOTE>		Used for long quotes or citations.
<ADDRESS>...</ADDRESS>		Used for signatures or general information about a document's author.
<CENTER>...</CENTER>		Centers text or images.
<BLINK>...</BLINK>		Causes the enclosed text to blink in an irritating manner.
<FONT>...</FONT>		Changes the size of the font for the enclosed text.
	SIZE="..."	The size of the font, from 1 to 7. Default is 3. Can also be specified as a value relative to the current size; for example, +2.
	COLOR="..."	Changes the color of the text.
	FACE="..."	Name of font to use if it can be found on the user's system. Multiple font names can be separated by commas, and the first font on the list that can be found will be used.
<BASEFONT>		Sets the default size of the font for the current page.
	SIZE="..."	The default size of the font, from 1 to 7. Default is 3.

**10. Images, Sounds, dan Embedded Media**

Tag	Attribute	Function
<IMG>		Inserts an inline image into the document.
	ISMAP	This image is a clickable image map.
	SRC="..."	The URL of the image.
	ALT="..."	A text string that will be displayed in browsers that cannot support images.
	ALIGN="..."	Determines the alignment of the given image. If LEFT or RIGHT, the image is aligned to the left or right column, and all following text flows beside that image. All other values such as TOP, MIDDLE, and BOTTOM, or the Netscape-only TEXTTOP, ABSMIDDLE, BASELINE, and ABSBOTTOM determine the vertical alignment of this image with other items in the same line.
	VSPACE="..."	The space between the image and the text above or below it.
	HSPACE="..."	The space between the image and the text to its left or right.
	WIDTH="..."	The width, in pixels, of the image. If WIDTH is not the actual width, the image is scaled to fit.
	HEIGHT="..."	The height, in pixels, of the image. If HEIGHT is not the actual height, the image is scaled to fit.
	BORDER="..."	Draws a border of the specified value in pixels to be drawn around the image. In the case of images that are also links, BORDER changes the size of the default link border.
	LOWSRC="..."	The path or URL of an image that will be loaded first, before the image specified in SRC. The value of LOWSRC is usually a smaller or lower resolution version of the actual image.
	USEMAP="..."	The name of an image map specification for client-side image mapping. Used with <MAP> and <AREA>.
	DYNSRC="..." (MS)	The address of a video clip or VRML world (dynamic source).
	CONTROLS (MS)	Used with DYNSRC to display a set of playback controls for inline video.
	LOOP="..." (MS)	The number of times a video clip will loop. (-1, or INFINITE, means to loop indefinitely.)
	START="..." (MS)	When a DYNSRC video clip should start playing. Valid options are FILEOPEN (play when page is displayed) or MOUSEOVER (play when mouse cursor passes over the video clip).
<BGSOUND> (MS)		Plays a sound file as soon as the page is displayed.
	SRC="..."	The URL of the WAV, AU, or MIDI sound file to embed.
	LOOP="..." (MS)	The number of times a video clip will loop. (-1, or INFINITE, means to loop indefinitely.)
<SCRIPT>		An interpreted script program.
	LANGUAGE="..."	Currently only JAVASCRIPT is supported by Netscape. Both JAVASCRIPT and VBSCRIPT are supported by Microsoft.
	SRC="..."	Specifies the URL of a file that includes the script program.
<OBJECT>		Inserts an image, video, Java applet, or ActiveX control into a document.
<APPLET>		Inserts a self-running Java applet.

	CLASS="..."	The name of the applet.
	SRC="..."	The URL of the directory where the compiled applet can be found (should end in a slash / as in http://mysite/myapplets/). Do not include the actual applet name, which is specified with the CLASS attribute.
	ALIGN="..."	Indicates how the applet should be aligned with any text that follows it. Current values are TOP, MIDDLE, and BOTTOM.
	WIDTH="..."	The width of the applet output area, in pixels.
	HEIGHT="..."	The height of the applet output area, in pixels.
<PARAM>		Program-specific parameters. (Always occurs within <APPLET> or <OBJECT> tags.)
	NAME="..."	The type of information being given to the applet or ActiveX control.
	VALUE="..."	The actual information to be given to the applet or ActiveX control.
	REF="..."	Indicates that this <PARAM> tag includes the address or location of the object.
<EMBED>		Embeds a file to be read or (Netscape only!) displayed by a plug-in application.
	SRC="..."	The URL of the file to embed.
	WIDTH="..."	The width of the embedded object in pixels.
	HEIGHT="..."	The height of the embedded object in pixels.
	ALIGN="..."	Determines the alignment of the media window. Values are the same as for the <IMG> tag.
	VSPACE="..."	The space between the media and the text above or below it.
	HSPACE="..."	The space between the media and the text to its left or right.
	BORDER="..."	Draws a border of the specified size in pixels to be drawn around the media.
<NOEMBED>...</NOEMBED>		Alternate text or images to be shown to users who do not have a plug-in installed.
<MAP>...</MAP>		A client-side image map, referenced by <IMG USEMAP="...">. Includes one or more <AREA> tags.
<AREA>		Defines a clickable link within a client-side image map.
	SHAPE="..."	The shape of the clickable area. Currently, only RECT is supported.
	COORDS="..."	The left, top, right, and bottom coordinates of the clickable region within an image.
	HREF="..."	The URL that should be loaded when the area is clicked.
	NOHREF	Indicates that no action should be taken when this area of the image is clicked.

**11. Forms**

Tag	Attribute	Function
<FORM>...</FORM>		Indicates an input form.
	ACTION="..."	The URL of the script to process this form input.
	METHOD="..."	How the form input will be sent to the gateway on the server side. Possible values are GET and POST.
	ENCTYPE="..."	Normally has the value application/x-www-form-urlencoded. For file uploads, use multipart/form-data.
	NAME="..."	A name by which JavaScript scripts can refer to the form.
<INPUT>		An input element for a form.
	TYPE="..."	The type for this input widget. Possible values are CHECKBOX, HIDDEN, RADIO, RESET, SUBMIT, TEXT, SEND FILE, or IMAGE.
	NAME="..."	The name of this item, as passed to the gateway script as part of a name/value pair.
	VALUE="..."	For a text or hidden widget, the default value; for a check box or radio button, the value to be submitted with the form; for Reset or Submit buttons, the label for the button itself.
	SRC="..."	The source file for an image.
	CHECKED	For check boxes and radio buttons, indicates that the widget is checked.
	SIZE="..."	The size, in characters, of a text widget.
	MAXLENGTH="..."	The maximum number of characters that can be entered into a text widget.
	ALIGN="..."	For images in forms, determines how the text and image will align (same as with the <IMG> tag).
<TEXTAREA>...</TEXTAREA>		Indicates a multi-line text entry form element. Default text can be included.
	NAME="..."	The name to be passed to the gateway script as part of the name/value pair.
	ROWS="..."	The number of rows this text area displays.
	COLS="..."	The number of columns (characters) this text area displays.
	WRAP="..."	Controls text wrapping. Possible values are OFF, VIRTUAL, and PHYSICAL.
<SELECT>...</SELECT>		Creates a menu or scrolling list of possible items.
	NAME="..."	The name that is passed to the gateway script as part of the name/value pair.
	SIZE="..."	The number of elements to display. If SIZE is indicated, the selection becomes a scrolling list. If no SIZE is given, the selection is a pop-up menu.
	MULTIPLE	Allows multiple selections from the list.
<OPTION>		Indicates a possible item within a <SELECT> element.
	SELECTED	With this attribute included, the <OPTION> will be selected by default in the list.
	VALUE="..."	The value to submit if this <OPTION> is selected when the form is submitted.

**12. Tables**

Tag	Attribute	Function
<TABLE>...</TABLE>		Creates a table that can contain a caption (<CAPTION>) and any number of rows (<TR>).
	BORDER="..."	Indicates whether the table should be drawn with or without a border. In Netscape, BORDER can also have a value indicating the width of the border.
	CELLSPACING="..."	The amount of space between the cells in the table.
	CELLPADDING="..."	The amount of space between the edges of the cell and its contents.
	WIDTH="..."	The width of the table on the page, in either exact pixel values or as a percentage of page width.
	ALIGN="..." (MS)	Alignment (works like IMG ALIGN). Values are LEFT or RIGHT.
	BGCOLOR="..."	Background color of all cells in the table that do not contain their own BACKGROUND or BGCOLOR attribute.
	BACKGROUND="..." (MS)	Background image to tile within all cells in the table that do not contain their own BACKGROUND or BGCOLOR attribute.
	BORDERCOLOR="..." (MS)	Border color (used with BORDER="...").
	BORDERCOLORLIGHT="..." (MS)	Color for light part of 3D-look borders (used with BORDER="...").
	BORDERCOLORDARK="..." (MS)	Color for dark part of 3D-look borders (used with BORDER="...").
	VALIGN="..." (MS)	Alignment of text within the table. Values are TOP and BOTTOM.
	FRAME="..." (MS)	Controls which external borders will appear around a table. Values are void (no frames), above (top border only), below (bottom border only), hside (top and bottom), lhs (left hand side), rhs (right hand side), vside (left and right sides), and box (all sides).
	RULES="..." (MS)	Controls which internal borders appear in the table. Values are none, basic (rules between THEAD, TBODY, and TFOOT only), rows (horizontal borders only), cols (vertical borders only), and all.
<CAPTION>...</CAPTION>		The caption for the table.
	ALIGN="..."	The position of the caption. Possible values are TOP and BOTTOM.
<TR>...</TR>		Defines a table row, containing headings and data (<TR> and <TH> tags).
	ALIGN="..."	The horizontal alignment of the contents of the cells within this row. Possible values are LEFT, RIGHT, and CENTER.
	VALIGN="..."	The vertical alignment of the contents of the cells within this row. Possible values are TOP, MIDDLE, BOTTOM, and BASELINE.

	BGCOLOR="..."	Background color of all cells in the row that do not contain their own BACKGROUND or BGCOLOR attributes.
	BACKGROUND="..." (MS)	Background image to tile within all cells in the row that do not contain their own BACKGROUND or BGCOLOR attributes.
	BORDERCOLOR="..." (MS)	Border color (used with BORDER="...").
	BORDERCOLORLIGHT="..." (MS)	Color for light part of 3D-look borders (used with BORDER="...").
	BORDERCOLORDARK="..." (MS)	Color for dark part of 3D-look borders (used with BORDER="...").
<TH>...</TH>		Defines a table heading cell.
	ALIGN="..."	The horizontal alignment of the contents of the cell. Possible values are LEFT, RIGHT, and CENTER.
	VALIGN="..."	The vertical alignment of the contents of the cell. Possible values are TOP, MIDDLE, BOTTOM, and BASELINE.
	ROWSPAN="..."	The number of rows this cell will span.
	COLSPAN="..."	The number of columns this cell will span.
	NOWRAP	Does not automatically wrap the contents of this cell.
	WIDTH="..."	The width of this column of cells, in exact pixel values or as a percentage of the table width.
	BGCOLOR="..."	Background color of the cell.
	BACKGROUND="..." (MS)	Background image to tile within the cell.
	BORDERCOLOR="..." (MS)	Border color (used with BORDER="...").
	BORDERCOLORLIGHT="..." (MS)	Color for light part of 3D-look borders (used with BORDER="...").
	BORDERCOLORDARK="..." (MS)	Color for dark part of 3D-look borders (used with BORDER="...").
<TD>...</TD>		Defines a table data cell.
	ALIGN="..."	The horizontal alignment of the contents of the cell. Possible values are LEFT, RIGHT, and CENTER.
	VALIGN="..."	The vertical alignment of the contents of the cell. Possible values are TOP, MIDDLE, BOTTOM, and BASELINE.
	ROWSPAN="..."	The number of rows this cell will span.
	COLSPAN="..."	The number of columns this cell will span.
	NOWRAP	Does not automatically wrap the contents of this cell.
	WIDTH="..."	The width of this column of cells, in exact pixel values or as a percentage of the table width.
	BGCOLOR="..."	Background color of the cell.
	BACKGROUND="..." (MS)	Background image to tile within the cell.
	BORDERCOLOR="..." (MS)	Border color (used with BORDER="...").
	BORDERCOLORLIGHT="..." (MS)	Color for light part of 3D-look borders (used with BORDER="...").
	BORDERCOLORDARK="..." (MS)	Color for dark part of 3D-look borders

		(used with BORDER="...").
--	--	---------------------------

### 13. Frames

Tag	Attribute	Function
<FRAMESET>...</FRAMESET>		Divides the main window into a set of frames that can each display a separate document.
	ROWS="..."	Splits the window or frameset vertically into a number of rows specified by a number (such as 7), a percentage of the total window width (such as 25%), or as an asterisk (*) indicating that a frame should take up all the remaining space or divide the space evenly between frames (if multiple * frames are specified).
	COLS="..."	Works similar to ROWS, except that the window or frameset is split horizontally into columns.
	BORDER="..."	Size of frame border in pixels (0 turns off borders). This tag is Netscape-specific--Microsoft IE uses FRAMEBORDER and FRAMESPACING instead.
	FRAMEBORDER="..." (MS)	Specifies whether to display a border for a frame. Options are YES and NO.
	FRAMESPACING="..." (MS)	Space between frames, in pixels.
<FRAME>		Defines a single frame within a <FRAMESET>.
	SRC="..."	The URL of the document to be displayed in this frame.
	NAME="..."	A name to be used for targeting this frame with the TARGET attribute in <A HREF> links.
	MARGINWIDTH="..."	The amount of space to leave to the left and right side of a document within a frame, in pixels.
	MARGINHEIGHT="..."	The amount of space to leave above and below a document within a frame, in pixels.
	SCROLLING="..."	Determines whether a frame has scrollbars. Possible values are YES, NO, and AUTO.
	NORESIZE	Prevents the user from resizing this frame (and possibly adjacent frames) with the mouse.
<NOFRAME>...</NOFRAME>		Provides an alternative document body in <FRAMESET> documents for browsers that do not support frames (usually encloses <BODY>...</BODY>).

### 14. Character Entities

Character Entities contains the possible numeric and character entities for the ISO-Latin-1 (ISO8859-1) character set. Where possible, the character is shown.

**Just a Minute:** Not all browsers can display all characters, and some browsers might even display characters different from those that appear in the table. Newer browsers seem to have a better track record for handling character entities, but be sure to test your HTML files extensively with multiple browsers if you intend to use these entities.

Character	Numeric Entity	Character Entity (if any)	Description
	&#00; -&#08;		Unused
	&#09;		Horizontal tab
	&#10;		Line feed
	&#11; -&#31;		Unused
	&#32;		Space

!	&#33;		Exclamation mark
"	&#34;	&quot;	Quotation mark
#	&#35;		Number sign
\$	&#36;		Dollar sign
%	&#37;		Percent sign
&	&#38;	&amp;	Ampersand
'	&#39;		Apostrophe
(	&#40;		Left parenthesis
)	&#41;		Right parenthesis
*	&#42;		Asterisk
+	&#43;		Plus sign
,	&#44;		Comma
-	&#45;		Hyphen
.	&#46;		Period (fullstop)
/	&#47;		Solidus (slash)
0-9	&#48;-&#57;		Digits 0-9
:	&#58;		Colon
;	&#59;		Semicolon
<	&#60;	&lt;	Less than
=	&#61;		Equal sign
>	&#62;	&gt;	Greater than
?	&#63;		Question mark
@	&#64;		Commercial "at"
A-Z	&#65;-&#90;		Letters A-Z
[	&#91;		Left square bracket
\	&#92;		Reverse solidus (backslash)
]	&#93;		Right square bracket
^	&#94;		Caret
--	&#95;		Horizontal bar
'	&#96;		Grave accent
a-z	&#97;-&#122;		Letters a-z
{	&#123;		Left curly brace
	&#124;		Vertical bar
}	&#125;		Right curly brace
~	&#126;		Tilde
	&#127;-&#160;		Unused
[exclamdown]	&#161;	&iexcl;	Inverted exclamation
[cents]	&#162;	&cent;	Cent sign
[sterling]	&#163;	&pound;	Pound sterling
[currency]	&#164;	&curren;	General currency sign
[yen]	&#165;	&yen;	Yen sign
	&#166;	&brvbar; or brkbar;	Broken vertical bar
[§]	&#167;	&sect;	Section sign
[dieresis]	&#168;	&uml;	Umlaut (dieresis)
©	&#169;	&copy; (Netscape only)	Copyright
ª	&#170;	&ordf;	Feminine ordinal
<	&#171;	&laquo;	Left angle quote, guillemot left
~	&#172;	&not;	Not sign
-	&#173;	&shy;	Soft hyphen
®	&#174;	&reg; (Netscape only)	Registered trademark



-	&#175;	&hibar;	Macron accent
deg.	&#176;	&deg;	Degree sign
+/-	&#177;	&plusmn;	Plus or minus
²	&#178;	&sup2;	Superscript two
³	&#179;	&sup3;	Superscript three
'	&#180;	&acute;	Acute accent
µ	&#181;	&micro;	Micro sign
¶	&#182;	&para;	Paragraph sign
.	&#183;	&middot;	Middle dot
[cedilla]	&#184;	&cedil;	Cedilla
¹	&#185;	&sup1;	Superscript one
º	&#186;	&ordm;	Masculine ordinal
>	&#187;	&raquo;	Right angle quote, guillemot right
¼	&#188;	&frac14;	Fraction one-fourth
½	&#189;	&frac12;	Fraction one-half
¾	&#190;	&frac34;	Fraction three-fourths
¿	&#191;	&iquest	Inverted question mark
À	&#192;	&Agrave;	Capital A, grave accent
Á	&#193;	&Aacute;	Capital A, acute accent
Â	&#194;	&Acirc;	Capital A, circumflex accent
[Atilde]	&#195;	&Atilde;	Capital A, tilde
Ä	&#196;	&Auml;	Capital A, dieresis or umlaut mark
Å	&#197;	&Aring;	Capital A, ring
Æ	&#198;	&AElig;	Capital AE diphthong (ligature)
Ç	&#199;	&Ccedil;	Capital C, cedilla
È	&#200;	&Egrave;	Capital E, grave accent
É	&#201;	&Eacute;	Capital E, acute accent
Ê	&#202;	&Ecirc;	Capital E, circumflex accent
Ë	&#203;	&Euml;	Capital E, dieresis or umlaut mark
Ì	&#204;	&Igrave;	Capital I, grave accent
Í	&#205;	&Iacute;	Capital I, acute accent
Î	&#206;	&Icirc;	Capital I, circumflex accent
Ï	&#207;	&Iuml;	Capital I, dieresis or umlaut mark
Ð	&#208;	&ETH;	Capital Eth, Icelandic
Ñ	&#209;	&Ntilde;	Capital N, tilde
Ò	&#210;	&Ograve;	Capital O, grave accent
Ó	&#211;	&Oacute;	Capital O, acute accent
Ô	&#212;	&Ocirc;	Capital O, circumflex accent
[Otilde]	&#213;	&Otilde;	Capital O, tilde
Ö	&#214;	&Ouml;	Capital O, dieresis or umlaut mark
×	&#215;		Multiply sign
Ø	&#216;	&Oslash;	Capital O, slash
Ù	&#217;	&Ugrave;	Capital U, grave accent
Ú	&#218;	&Uacute;	Capital U, acute accent
Û	&#219;	&Ucirc;	Capital U, circumflex accent
Ü	&#220;	&Uuml;	Capital U, dieresis or umlaut mark
Ý	&#221;	&Yacute;	Capital Y, acute accent
Þ	&#222;	&THORN;	Capital THORN, Icelandic
ß	&#223;	&szlig;	Small sharp s, German (sz ligature)

à	&#224;	&agrave;	Small a, grave accent
á	&#225;	&aacute;	Small a, acute accent
â	&#226;	&acirc;	Small a, circumflex accent
ã	&#227;	&atilde;	Small a, tilde
ä	&#228;	&auml;	Small a, dieresis or umlaut mark
å	&#229;	&aring;	Small a, ring
æ	&#230;	&aelig;	Small ae diphthong (ligature)
ç	&#231;	&ccedil;	Small c, cedilla
è	&#232;	&egrave;	Small e, grave accent
é	&#233;	&eacute;	Small e, acute accent
ê	&#234;	&ecirc;	Small e, circumflex accent
ë	&#235;	&euml;	Small e, dieresis or umlaut mark
ì	&#236;	&igrave;	Small i, grave accent
í	&#237;	&iacute;	Small i, acute accent
î	&#238;	&icirc;	Small i, circumflex accent
ï	&#239;	&iuml;	Small i, dieresis or umlaut mark
ú	&#240;	&eth;	Small eth, Icelandic
ñ	&#241;	&ntilde;	Small n, tilde
ò	&#242;	&ograve;	Small o, grave accent
ó	&#243;	&oacute;	Small o, acute accent
ô	&#244;	&ocirc;	Small o, circumflex accent
õ	&#245;	&otilde;	Small o, tilde
ö	&#246;	&ouml;	Small o, dieresis or umlaut mark
/	&#247;		Division sign
ø	&#248;	&oslash;	Small o, slash
ù	&#249;	&ugrave;	Small u, grave accent
ú	&#250;	&uacute;	Small u, acute accent
û	&#251;	&ucirc;	Small u, circumflex accent
ü	&#252;	&uuml;	Small u, dieresis or umlaut mark
ÿ	&#253;	&yacute;	Small y, acute accent
w	&#254;	&thorn;	Small thorn, Icelandic
ÿ	&#255;	&yuml;	Small y, dieresis or umlaut mark