

## BAB II

### DASAR TEORI

Dalam bab ini akan dijelaskan tentang dasar teori yang digunakan sebagai acuan dalam mendesain dan mengimplementasikan pembangunan *game online multiplayer* berbasis web. Teori yang dijelaskan disini mengenai konsep dasar *game* dan sejarah perkembangan *game*, *internet*, *CSS*, *server side* dan *client side*, *HTML*, *JavaScript*, *PHP*, *AJAX* dan *MySQL*.

#### II.1 *Game*

Kata *Game* berasal dari Bahasa Inggris. Dalam Bahasa Indonesia *game* diartikan sebagai permainan atau pertandingan. Banyak filosofi yang dapat diambil dari sebuah *game*. Misalnya, bagaimana mengatur strategi ataupun mengambil keputusan. Namun yang utama adalah kesenangan dan keasyikan yang dihadapkannya.

*Game* yang berkembang saat ini, yang begitu menarik, menantang, dan canggih diiringi dengan perkembangan komputer, seperti *Final Fantasy*, *Everquest*, pembuatannya didukung oleh biaya jutaan US\$. Riset dan ujicoba berkali-kali pun telah mereka lakukan demi kesempurnaan. Para penciptanya pun adalah para pembuat *game* yang sangat berpengalaman (*telset*, 2004).

*Game* seperti itu dikembangkan menggunakan bahasa pemrograman tertentu untuk memberikan nuansa baru di dalam dunia permainan, dimana para pemain ikut berinteraksi dalam permainan tersebut untuk menyelesaikan misi yang sudah ditentukan (*Chandra*, 1996).

### II.1.1 Elemen-elemen *Game*

*Game* dibagi menjadi beberapa elemen. Elemen-elemen tersebut adalah (Chandra,1996):

a. *Desain Game*

*Desain game* merupakan elemen *game* yang sangat penting, dikarenakan suatu *game* haruslah dibuat semenarik mungkin. *Desain game* tidak sama dengan pemrograman *game*, karena cakupannya lebih luas dari pemrograman dimana isinya merupakan ide cerita, rancangan *level-level*, musuh yang akan dihadapi dan penghargaan yang diberikan. *Desain game* boleh dikatakan mirip dengan skenario film karena skenario tersebut mencakup seluruh aspek film. Pendek kata, *desain* adalah langkah awal untuk membuat semua elemen *game*. *Desain game* dibuat semenarik mungkin agar pemain tidak cepat bosan yang dampaknya membuat *game* tersebut cepat ditinggalkan.

b. *Pemrograman Game*

*Pemrograman game* juga merupakan faktor penting karena menggabungkan elemen-elemen lain untuk dibentuk menjadi sebuah *game*. *Pemrograman game* sebagian besar digunakan untuk mengontrol gerakan obyek di layar. Selain itu, *pemrograman game* juga digunakan untuk pemrograman suara, *input device*, deteksi keadaan dan lain-lain.

Untuk *pemrograman game online multiplayer* ini, penulis menggunakan bahasa pemrograman *PHP* dan bahasa pendukung lainnya seperti *Javascript* *AJAX*, *CSS* dan *HTML*.

### c. Grafis *Game*

Grafis pada *game* merupakan elemen *game* yang cukup penting, karena grafis *game* merupakan tampilan luar dari *game* tersebut. Grafis *game* memegang peranan penting dalam pembuatan tampilan, tampilan haruslah dibuat semenarik mungkin, sehingga dengan melihatnya saja *end user* dapat langsung tertarik untuk memainkannya.

### II.1.2 Sejarah dan Perkembangan *Game*

*Game* tidaklah asing bagi kita apalagi di jaman yang modern ini. Seiring dengan perkembangan jaman, *game* juga semakin berkembang dari *Atari*, *Computer Space*, *Nintendo*, *Sega*, *PlayStation*, dan sekarang *Game Komputer* sendiri.

*Atari* merupakan sebuah fenomena besar dalam dunia *video game*. Pada tahun 1972, *Atari* telah memperkenalkan permainan 'Pong' yang sangat terkenal sehingga menjadi *best selling item* pada tahun 1975 ketika musim cuti sekolah. *Atari* seterusnya mengeluarkan konsol *video game* yang dikenal sebagai *Atari 2600* yang telah menjadi pelopor *video game* sekarang.

Walaupun *Atari* dianggap sebagai pelopor kejayaan *video game*. Sebaliknya, *video game* yang benar-benar dibangun untuk umum adalah *Computer Space* oleh *Nutting Associates* pada tahun 1971. Namun *Atari* dan *Nutting Associates* mempunyai kaitan erat karena *Atari* dikeluarkan oleh orang yang membuat *Computer Space* yaitu Nolan Bushnell.

*Nintendo*, satu lagi nama besar dalam dunia *video game*, telah menggariskan beberapa konsep yang juga menjadi dasar infrastruktur *video game* pada masa sekarang misalnya penggunaan *pad* serta membuat *video game* yang sesuai dengan *home theater system* (bisa dihubungkan pada TV atau radio). *Nintendo* merupakan sistem permainan yang paling berjaya di dunia dengan lebih daripada 100 juta unit terjual di seluruh dunia.

Konsol pertama yang membuat user bisa bermain secara *online* adalah *Dreamcast* keluaran *Sega*. Pada tahun 1999, majalah *Popular Science* mengumumkan *Dreamcast* sebagai produk yang paling inovatif. Saat itu, *Sega* mengutamakan isi dari *game* dimana *game* dibuat serasi dengan kebanyakan platform *video game* seperti *PS2*, *Xbox*, dan *GameCube*.

*Playstation* adalah *game* yang cukup populer sekarang ini. *Playstation* terbaru yaitu *Playstation 3* (*PS3*) merupakan *video game* pertama yang menggunakan *DVD*. Grafik yang menarik serta jumlah *game* yang banyak terdapat di pasaran menjadikan *PS3* cukup populer.

*Game* pada komputer diperkenalkan oleh *IBM PC* pada tahun 1981 dimana komputer jenis ini lebih meningkatkan kreativitas pembuat program mainan di komputer, karena semakin lama komputer *IBM Compatible* ini semakin cepat, grafiknya semakin baik, dan tempat penyimpanannya semakin besar.

Pada komputer yang berdiri sendiri atau *stand alone* kita hanya bisa bermain melawan komputer atau maksimum dua orang dengan menggunakan *joystick*. Sedangkan pada komputer yang berdiri dengan sistem jaringan, bisa lebih dari tiga orang bermain, dan

bahkan kemajuan teknologi jaringan *internet*, memungkinkan ribuan orang bermain di satu server komputer.

Elemen-elemen *Game* yaitu jenis *game*, karakter dalam *game*, *background*, elemen *sound* / suara, dan gerakan-gerakan. Klasifikasi *game* yang dapat dimainkan ada beberapa tipe yaitu :

1. *Side scrolling game*, pada *game* ini, karakter dapat bergerak ke samping diikuti dengan gerakan *background*. (Contoh : *Super Mario*)
2. *Shooting game*, ada beberapa macam *game shooting*, diantaranya *First Person Shooting* dan *Third Person Shooting*. (Contoh : *Counter Strike*)
3. *RPG (RolePlay Game)*, adalah *game* memainkan peran suatu karakter dalam menjalankan misi tertentu. (Contoh : *Final Fantasy*)
4. *RTS (RealTime Strategy)*, pada *game* ini, pemain memerintahkan beberapa karakter untuk melakukan strategi perang. (Contoh : *Red Alert 2*)
5. *Simulation*, *game* ini merupakan simulasi dari keadaan *real*. Ada beberapa pembagian *game* simulasi antara lain adalah *game* simulasi manusia (*the SIMS*), *game* simulasi kendaraan (*Ace Combat*), dan *game* simulasi lainnya (*Sim City*, dll).
6. *Racing*, *game* ini merupakan *game* balapan otomotif untuk melatih ketangkasan. (contoh : *Need for Speed*, *Grand Turismo*, dll).

7. *Fighting*, dalam *game* ini, pemain memainkan satu atau lebih karakter untuk bertarung dengan karakter lain. (contoh : *Tekken*)
8. *Text Base*, dalam *game* ini biasanya dimainkan secara online dengan *interface* berbentuk *text* berbasis *web*. Meskipun dengan *interface text* tetapi dapat dimainkan oleh banyak orang. *Game* ini termasuk sederhana dan meminimalkan spesifikasi hardware yang biasanya digunakan untuk *game* grafik 2D maupun 3D. (contoh: *Kurusetra*).

### II.1.3 *Game Online Multiplayer*

Dahulu main *game* lebih enak dimainkan sendiri. Makin lama, perkembangan teknologi *game* memungkinkan pemain bermain dengan beberapa orang sekaligus. Kini dengan adanya perkembangan di dunia *game online*, pemain *game online* dapat saling berinteraksi ribuan orang dari penjuru dunia bukanlah hal yang mustahil. Kemampuan *game* jaringan yang memungkinkan puluhan orang bermain sekaligus mulai mendapatkan pesaing ketika *game online* muncul. Dengan adanya *game online* memungkinkan pemain bermain bersama ribuan orang sekaligus dari berbagai lokasi. Infrastruktur yang dibutuhkan dalam *game online* hanya diperlukan sambungan internet untuk melakukan koneksi pada *game* yang akan dimainkan.

Jenis *game*-nya pun tidak beda jauh dengan *game* yang biasa ditemukan di *konsole* atau di pusat *game* jaringan. Ada yang berjenis strategi, *role playing*, atau bahkan *sports*.

Yang masuk *game online* jenis baru macam ini ada beberapa macam. Contohnya, *Ragnarok* yang bergenre *massive multiplayer online role playing game (MMORPG)*, *Pangya* yang bergenre *fantasy sport*, *Gunbound* yang masuk kategori *turn based*, atau *Kurusetra* yang masuk kategori *text based*. Jenis *genre* macam itu yang kemudian mengotak-ngotakkan beberapa *game* sehingga mempunyai penggemarnya sendiri. Hal ini yang membedakan dengan *game online* standard yang jamak dijumpai di situs-situs macam Yahoo atau MSN.

Perkembangan *game online* tidak hanya memiliki hiburan semata. *Xian Online*, misalnya, *game* yang dilisensi oleh *BolehGame* ini mengajarkan dasar-dasar ilmu ekonomi. *Game* ini mengajari para pemainnya hal apa saja yang memengaruhi harga suatu barang. Lewat *game* ini pula pemain dapat belajar investasi. Selain itu, ada *TS Online*, yang mengajak para pemainnya mempelajari sejarah Sam Kok, plus mengasah kepekaan menghadapi suatu masalah. Selain beberapa *game online* yang memiliki sifat interaksi langsung, ada juga jenis *game online* yang didasarkan pada *web based*. Pada jenis *game online* macam ini pemain tidak berinteraksi langsung dengan karakter pemain. Strategi dan ketekunan yang membuat karakter pemain semakin kuat. *Kurusetra*, *The Crims* atau *Neopets* bisa jadi contoh *game* dengan berbasis *web*.

([www.kompas.com/kompas-cetak/0510/21/muda/2143888.htm](http://www.kompas.com/kompas-cetak/0510/21/muda/2143888.htm))

## II.2 Internet

*Internet* menghubungkan banyak jaringan yang ada di dunia, jaringan tersebut bisa terdiri dari jaringan komputer pendidikan, jaringan komputer komersial, jaringan komputer pemerintah, jaringan komputer militer dan sebagainya. Jadi *internet* adalah jaringan komputer seluruh dunia.

*Internet* pertama kali diciptakan oleh Departemen Pertahanan AS pada tahun 1969 dibawah nama APRANet (*Advanced Research Project Agency*). Pada awalnya *internet* hanya digunakan untuk penelitian militer, industri, komunikasi dan pendidikan.

Seiring dengan kemajuan teknologi maka *internet* semakin berkembang dan memiliki banyak fungsi, diantaranya :

- a. *World wide Web* yaitu bagian dari *internet* yang terdiri dari situs-situs atau *server* dari seluruh dunia yang dapat menyimpan dan menyajikan informasi.
- b. *Email* yaitu komunikasi surat menggunakan *internet*.
- c. *Usenet New Group* yaitu menggunakan *internet* untuk mendiskusikan masalah.
- d. *Mailing List*, yaitu gabungan dari email dan *usenet new groups*.
- e. *FTP (File Transfer Protocol)* digunakan untuk mengirimkan *file* ke seluruh dunia.
- f. *Telnet*, digunakan untuk menghubungkan dan mengendalikan komputer dari jarak jauh.

HTTP (*HyperText Transfer Protocol*) adalah protokol yang digunakan untuk layanan WWW (*World Wide Web*) di jaringan TCP/IP. Protokol HTTP bersifat *request-response*, yaitu dalam protokol ini *client* menyampaikan pesan *request* ke *server* dan *server* kemudian memberi *response* yang sesuai dengan *request* tersebut.

Pada protokol HTTP terdapat tiga jenis hubungan dengan perantara, yaitu *proxy*, *gateway*, dan *tunel*. *Proxy* bertindak sebagai agen penerus, menerima *request* dalam bentuk *Uniform Resource Identifier (URI)* absolut, mengubah format *request*, dan mengirimkan *request* ke *server* yang ditunjukkan oleh URI. *Gateway* bertindak sebagai agen penerima dan menerjemahkan *request* ke protokol *server* yang dilayani. *Tunnel* bertindak sebagai titik *relay* antara dua hubungan HTTP tanpa mengubah *request* dan *response* HTTP.

### **II.3 Konsep Pembuatan Web**

Untuk membuat suatu situs *web* yang menarik dan interaktif yang tidak hanya sekedar menampilkan informasi, tetapi dapat berinteraksi dengan *user*, sehingga suatu situs tidak bersifat statis tetapi dapat bersifat menjadi dinamis, membutuhkan kemampuan pemrograman *web*.

### **II.4 Server Side dan Client Side**

Teknologi yang digunakan dalam pemrograman *web* dibagi menjadi dua, yaitu *server side* dan *client side*.

Pada *server side*, perintah-perintah program dijalankan di *server* dan dikirimkan ke *browser* dalam bentuk HTML. Sedangkan *client side*, proses akan dilakukan di *web browser*. Biasanya *client side* digunakan untuk hal-hal yang membutuhkan interaksi *user* tetapi data yang ditampilkan tetap dan seragam.

Aplikasi *web* berjalan pada protokol HTTP, dan semua protokol di internet selalu melibatkan antara *Server* dan *Client*. Ketika seseorang mengetikkan suatu alamat di *browser*, maka *browser* akan mengirimkan perintah tersebut ke *web server*. Jika yang diminta oleh *client* adalah *file* yang mengandung perintah *server side* maka *server* akan menjalankan dahulu program tersebut lalu mengirimkannya kembali ke *browser* dalam bentuk HTML sehingga dapat diterjemahkan oleh *browser*. Sedangkan jika yang diminta oleh *client* adalah *file* yang mengandung *file client side* maka oleh *server side* tersebut akan langsung dikirimkan ke *browser*.

Banyak sekali bahasa pemrograman yang dipakai pada pemrograman *web*, diantaranya: CGI/PERL, *Active Server Pages* (ASP), *Java Server Pages* (JSP), PHP, *Cold Fusion* (CFM), dan sebagainya, yang semuanya itu adalah program *server side*. Sedangkan untuk *client side* misalnya *Java Script*, *VB Script*, *HTML*.

## II.5 HTML (Hypertext Markup Language)

*Hypertext Markup Language* adalah suatu sistem untuk menambahkan *tags* pada dokumen yang menandakan bagaimana teks di dokumen harus disajikan dan bagaimana dokumen dihubungkan bersama-sama. *Tags* dalam HTML selalu berada di dalam tanda lebih kecil (<) dan tanda

lebih besar (>) serta boleh ditulis dalam huruf besar maupun huruf kecil. *Tags* biasanya merupakan suatu pasangan, yang biasanya disebut tag awal dan tag akhir di mana tag awal dinyatakan dalam bentuk `<nama_tag>`, sedangkan tag akhir dinyatakan dalam bentuk `</nama_tag>`.

### II.5.1 Struktur Dasar HTML

Sebuah *file* yang sedang memuat teks yang ditambahkan pada suatu Web Page dinamakan *file* HTML. *File* ini dimulai dan diakhiri dengan tag `<html>` dan `</html>`. *File* ini dibagi menjadi dua bagian, yaitu kepala dan batang tubuh. Bagian kepala berisi informasi tentang dokumen dan batang tubuh berfungsi untuk menyimpan informasi atau data yang akan ditampilkan dalam dokumen HTML. Bagian kepala diawali dan diakhiri dengan tag `<head>` dan `</head>`, sedangkan bagian batang tubuh diawali dan diakhiri dengan tag `<body>` dan `</body>`. Berikut ini adalah contoh sebuah *file* HTML dasar :

```
<html>
<head>
<title>Halaman HTML Sederhana</title>
</head>
<body>
<p><!--Ini adalah komentar yang tidak ditampilkan--></p>
<p>Teks ini akan ditampilkan pada halaman browser.</p>
</body>
</html>
```

Tulisan yang terletak antara tag `<title>` dan `</title>` akan dicetak pada *Titlebar* jendela *Browser* dan tag `<title>` harus diletakkan pada bagian kepala. Sedangkan tulisan yang terletak di antara tag `<body>` dan `</body>` akan dicetak pada halaman *browser*. Komentar dituliskan pada tag `<!-- teks -->` di mana *teks* yang dituliskan di dalam tag ini tidak akan ditampilkan pada *browser*. Struktur Dasar *text* dari HTML adalah sebagai berikut :

#### II.5.1.1 *Heading*

*Heading* digunakan untuk membuat judul atau kepala bagian di mana ada 6 tingkat *heading* yaitu *heading 1* hingga *heading 6*, yang ditandai dengan pasangan tag `<h1>` dan `</h1>` sampai pasangan tag `<h6>` dan `</h6>`. Pada *browser*, *heading* akan ditampilkan dengan huruf tebal. *Heading 1* adalah *heading* yang ditampilkan dengan ukuran huruf terbesar dan *heading 6* adalah *heading* yang ditampilkan dengan ukuran huruf terkecil. *Teks* yang diletakkan di antara tag *heading* akan dianggap sebagai paragraf tersendiri.

#### II.5.1.2 *Paragraph*

Sebuah *paragraf* ditandai dengan pasangan tag `<p>` dan `</p>` di mana *teks* yang ditulis diantara tag ini akan dianggap sebagai satu kesatuan paragraf. Perpindahan baris akan dilakukan secara otomatis oleh *browser* sehingga perpindahan baris pada *teks* hanya dianggap sebagai spasi.

### II.5.1.3 *Line Break*

*Line break* digunakan untuk memaksakan perpindahan baris pada teks di dalam sebuah paragraf dengan menggunakan tag `<br>` tanpa tag penutup `</br>`.

### II.5.1.4 *Blockquote*

*Blockquote* digunakan untuk menuliskan paragraf yang menjorok ke dalam pada kedua sisinya.

### II.5.1.5 *List*

#### a. *Unordered Lists*

*Unordered lists* adalah sebuah daftar tanpa nomor, tetapi hanya menggunakan kotak (`type="square"`) atau lingkaran tanpa isi (`type="circle"`) atau lingkaran dengan isi (`type="disc"`) sebagai tanda anggotanya. Sebuah *unordered lists* ditandai dengan pasangan tag `<ul type="tipe_yg_dipilih">` dan `</ul>`, dan masing-masing anggotanya ditandai dengan pasangan tag `<li>` dan `</li>`.

#### b. *Ordered Lists*

*Ordered lists* adalah sebuah daftar dengan nomor yang ditandai dengan pasangan tag `<ol type="tipe_yg_dipilih">` dan `</ol>`, dan masing-masing anggotanya diawali dengan tag `<li>` dan diakhiri dengan tag `</li>`. Tipe-tipe yang dapat dipilih pada *ordered lists* antara lain a, A, 1, I, atau i.

### **c. Definition Lists**

*Definition lists* digunakan untuk membuat suatu daftar definisi dari suatu istilah di mana istilah ditulis rapat kiri dan pengertian dari istilah tersebut ditulis agak menjorok ke dalam pada baris berikutnya. *Definition lists* ditandai dengan pasangan tag `<dl>` dan `</dl>`. Istilah diawali dengan tag `<dt>` ("*Definition Lists Term*") dan definisi dari istilah tersebut diawali dengan tag `<dd>` ("*Definition Lists Definition*").

#### **II.5.1.6 Tag Spesial Efek**

Tag-tag berikut digunakan untuk memberikan efek *tipografi* tertentu pada teks yang akan ditampilkan pada *browser*. Ada dua kategori untuk tag ini yaitu *Logical Style Tags* dan *Forced Style Tags*.

##### **1. Logical Style Tags**

Tag ini tidak langsung menunjukkan bagaimana teks ditampilkan.

###### **a. Emphasis**

Ditandai dengan pasangan tag `<em>` dan `</em>`

###### **b. Strong**

Ditandai dengan pasangan tag `<strong>` dan `</strong>`

###### **c. Citation**

Ditandai dengan pasangan tag `<cite>` dan `</cite>`

## 2. *Forced Style Tags*

Tag ini akan langsung menunjukkan bagaimana teks ditampilkan.

### a. *Boldface*

Teks di antara pasangan tag `<b>` dan `</b>` akan dicetak tebal.

### b. *Italics*

Teks di antara pasangan tag `<i>` dan `</i>` akan dicetak miring.

### c. *Underline*

Teks di antara pasangan tag `<u>` dan `</u>` akan diberi garis bawah.

### d. *Typewriter text*

Teks di antara `<tt>` dan `</tt>` akan dicetak dengan huruf seperti mesin ketik.

### e. *Horisontal Rule*

Tag `<hr>` ini tidak termasuk *style*, tetapi digunakan untuk membuat garis mendatar pada dokumen.

## II.5.1.7 Hubungan

Keistimewaan sebuah dokumen HTML adalah kemampuannya untuk dihubungkan dengan dokumen HTML lain, yaitu *file gambar, movie, dan suara*. Di dalam dokumen HTML terdapat tag tanda jangkar (*anchor*), yaitu `<a>` dan `</a>`. Didalam tag `<a>` masih memerlukan atribut yaitu *HREF* atau *NAME*.

### a. *HREF*

*HREF* adalah singkatan dari *hypertext reference* yaitu atribut yang menunjukkan tempat *file* yang

akan dimuat ke dalam *browser*. Atribut ini diletakkan di dalam *anchor* sebagai berikut : `<a href = "url" > ..... </a>`. *URL (Uniform Resource Locator)* adalah cara standard untuk menyatakan lokasi dan tipe data dari sumber pada internet. Secara umum *URL* berbentuk "*protocol://address*". *Protocol* dapat berupa *file, ftp, http, gopher, telnet, dll*. Jika kata tersebut dipilih, maka *browser* akan mengambil *file* contoh.htm yang berada di *directory windows drive* lokal C. Perhatikan bahwa sesudah kata *file:* terdapat tiga buah garis miring. Karena yang diambil adalah *file local*, maka sesudah garis miring kedua tidak ada alamat lokasi. Yang disebut *file* lokal adalah *file* yang terdapat pada komputer sendiri ataupun yang terdapat pada komputer lain dalam LAN. Perhatikan penulisan direktori pada alamat menggunakan *forward-slash (/)*, bukan *back-slash (\)*.

#### **b. NAME**

*NAME* digunakan untuk memberi label pada dokumen yang kita buat. Label tersebut dapat dituju dengan tanda hubungan.

#### **II.5.1.8 Tabel**

Tabel adalah serangkaian baris yang masing-masing terdiri dari serangkaian sel. Sel-sel dalam tabel ada dua macam, yaitu sel *header* dan sel *data*. Sel *header* biasanya berada pada baris teratas atau kolom terdepan. Suatu tabel diawali dan diakhiri dengan tag `<table>` dan `</table>`.

Setiap baris pada tabel diawali dan diakhiri dengan tag `<tr>` dan `</tr>`. Sel *header* diapit oleh tag `<th>` dan `</th>` dan sel data diapit oleh tag `<td>` dan `</td>`. Tag untuk sel mempunyai atribut `align="left|right|center"`, untuk mengatur posisi *horizontal* teks dalam sel, dan `valign="top|middle|bottom"`, untuk mengatur posisi vertikal teks dalam sel. Untuk memberi *caption* pada tabel digunakan tag `<caption>` judul `</caption>`. Tag ini mempunyai atribut `align="top|bottom"`, untuk mengatur letak *caption* di atas atau di bawah tabel.

#### II.5.1.9 Gambar

Gambar disisipkan ke dalam dokumen dengan tag `<img>`. Tag ini tidak mempunyai tag penutup. Atribut yang terdapat dalam tag `<img>` adalah *src*, *alt*, dan *align*.

##### a. SRC

SRC digunakan untuk menunjukkan sumber-sumber gambar. Formatnya adalah sebagai berikut: `<img src= "alamat_gambar">`. Format penulisan alamat gambar sama seperti pada *href*. Gambar yang dapat disisipkan ke dalam dokumen HTML adalah gambar dengan *format* GIF atau JPG.

##### b. ALT

ALT digunakan untuk memberikan teks alternatif jika gambar tidak dimuat ke dalam *browser*. Pada beberapa *browser* ada fasilitas untuk tidak memuat gambar. Hal ini dimaksudkan untuk mempercepat proses pemuatan dokumen HTML.

### c. ALIGN

ALIGN digunakan untuk mengatur posisi gambar terhadap teks. *Format* penulisan dapat dilihat pada contoh di atas. Posisi dapat diganti dengan *top*, *middle*, atau *bottom*. Jika *top*, maka bagian atas gambar akan rata dengan bagian atas teks. Jika *bottom*, maka bagian bawah gambar akan rata dengan bagian bawah teks. Jika *middle*, maka ketinggian gambar tepat di tengah-tengah.

#### II.5.1.10 Formulir

Sebuah formulir adalah suatu daerah HTML page yang ditandai, seringkali diubah dengan batas yang ada di sekeliling, yang berisi field masukan dan obyek-obyek interaktif lain, seperti menu pop-up, kotak periksa, dan tombol. Formulir diawali dan diakhiri dengan tag/label `<form>` `</form>`. Pada label `<form>` awal terdapat dua atribut yaitu *action* yang menunjukkan lokasi tempat program yang hendak dijalankan dan *methode* yang menandakan bagaimana isi formulir akan disajikan ke *script* atau e-mail. Biasanya selalu menggunakan *value* = "post" untuk menentukan bahwa isi disajikan sebagai masukan standar.

##### a. Scrolling Text Box

Elemen ini merupakan salah satu elemen *form* yang dipergunakan dengan `<form>`, dipakai bila ingin memasukkan informasi/komentar yang panjangnya beberapa baris.

**b. Drop-Down**

Akan menampilkan suatu daftar pilihan, baik dalam bentuk menu *pop-up* maupun menu gulung. *SIZE* menunjukkan jumlah pilihan yang ingin ditampilkan. Apabila *size* ditiadakan atau nilai *size*="1", menu akan ditampilkan dalam bentuk *pop-up*, tetapi bila *size* ditentukan lebih dari 2 dengan sendirinya *browser* akan menampilkan menu gulung (*scrollable*).

**c. Text Box**

dipakai untuk memasukan isian berupa text.

**d. Radio Button**

dipakai untuk memilih salah satu dari banyak pilihan.

**e. Checkbox**

dipakai untuk memilih bias lebih dari satu dari beberapa pilihan.

**f. Push Button "Submit"**

Merupakan suatu tombol yang digunakan untuk mengirimkan data-data yang telah dimasukkan ke dalam *form*.

**g. Pust Button "Reset"**

Tombol ini disediakan untuk menghapus semua informasi yang telah ada di dalam *form*, sehingga pengisian *form* dapat dimulai lagi dari awal.

**II.6 Java Script**

*JavaScript* adalah *script* yang tidak berbeda dengan bahasa pemrograman karena *script* juga berupa kumpulan instruksi program, hanya saja *script* mempunyai aturan yang tidak terlalu ketat dan tidak dikompilasi. Dengan

*JavaScript*, *webpage* yang tadinya hanya bersifat statis dapat dibuat interaktif dan dinamis. *Java Script* dapat menambahkan operasi aritmatika, menampilkan waktu lokal, menerima masukan dari user, menambahkan *counter* pengunjung, animasi sederhana dan banyak hal lain pada *web page*.

*JavaScript* dikembangkan oleh *Netscape Communications* bersama dengan *Sun Microsystems*. *JavaScript* sebenarnya khusus dirancang untuk browser *Netscape Navigator*, namun dalam perkembangannya browser *Internet Explorer* dari *Microsoft* juga dapat menjalankan *JavaScript*. *Microsoft* sendiri sebenarnya telah merilis sendiri *JavaScript* yaitu yang dinamakan *Jscript*.

Kelebihan *JavaScript* diantaranya adalah :

a. *Cepat*

Anda meletakkan *JavaScript* di dalam file HTML, jadi bisa sekaligus dan cepat dalam memanipulasi halaman *Web*. Program yang dibuat oleh *JavaScript* juga dapat langsung dicoba pada browser tanpa harus dikompilasi terlebih dahulu, sehingga memotong waktu pembuatan jadi lebih cepat.

b. *Mudah*

Karena *JavaScript* hanya memiliki sedikit *sintaks*, penghafalannya tidak sulit dilakukan. Semua perintahnya juga memakai bahasa Inggris sederhana yang mudah dihafal dan dimengerti. Walaupun *JavaScript* memiliki banyak persamaan dengan *Java*, ia tidak memiliki aturan maupun *sintaks* yang serumit *Java*. Setiap orang yang memiliki kemampuan programming terbatas pun dapat menggunakannya.

*c. Terbuka*

*JavaScript* tidak terikat oleh sistem operasi, perangkat keras, atau program tertentu. Ia bersifat terbuka dan fleksibel. *JavaScript* dapat dibuat pada komputer PC yang menggunakan sistem operasi Windows dan dibaca oleh komputer *mainframe* yang memakai UNIX.

*d. Ukurannya kecil*

Program *JavaScript* relatif berukuran sangat kecil jika dibandingkan dengan program-program yang dibuat dengan Java. Hal ini dapat mengurangi beban penyimpanan pada komputer *server* dan mempersingkat waktu *download*-nya.

Sedangkan kekurangan *JavaScript* diantaranya :

*a. Kemampuan terbatas*

*JavaScript* walaupun memperindah sebuah halaman *web*, ia tidak dapat digunakan untuk melakukan tugas-tugas yang lain seperti membuat program aplikasi sendiri.

*b. Object Terbatas*

*JavaScript* hanya memiliki beberapa *object* yang sudah *built-in* pada struktur bahasanya. Anda tidak dapat membuat kelas-kelas yang bisa menampung *object-object* tambahan seperti Java.

*c. Tidak dapat disembunyikan*

Setiap orang dapat membaca program *JavaScript* yang Anda buat. Hal ini dapat mengakibatkan pencontekan atau penggunaan program Anda secara luas dan tidak terkendali.

*JavaScript* adalah *script* yang dituliskan (*embedded*) pada *file* HTML. Penulisan *JavaScript* pada *file* HTML (.html atau .htm) dengan menggunakan tag container `<SCRIPT>`. Dengan kata lain program *JavaScript* tidak perlu dituliskan pada *file* terpisah.

Ingat bahwa tag *container* adalah tag yang diawali dengan `<NAMA_TAG>` dan diakhiri dengan `</NAMA_TAG>`. Beberapa contoh tag *container* adalah `<HTML></HTML>`, `<HEAD></HEAD>`, `<BODY></BODY>`.

Tag *container* `<SCRIPT>` mempunyai dua atribut tetapi yang harus diisi hanya satu atribut, yaitu *Language* dengan *JavaScript*. Hal ini memberitahukan *browser* bahwa *script* yang ditulis adalah dalam *JavaScript*.

Contoh:

```
<SCRIPT LANGUAGE="JavaScript">
    //Program Javascript anda tuliskan disini
</SCRIPT>
```

### II.7 CSS (*Cascading Style Sheets*)

*CSS* (*Cascading Style Sheets*) merupakan bagian terpenting perancangan suatu website, yaitu di dalam membuat gaya atau *style* dari halaman web. Gaya atau *style* yang dibuat dapat digunakan oleh seluruh halaman web dengan cara menghubungkannya dengan HTML. Suatu teks pada HTML yang membutuhkan banyak tag, tentunya akan terlihat sangat sederhana apabila menggunakan *CSS*. *CSS* dibuat pada halaman terpisah dari HTML. Tag-tag yang terlihat rumit, yang digunakan untuk mengatur tampilan, dibuat pada halaman *CSS*. Dengan demikian, update pada halaman yang menggunakan *CSS* akan mudah dilakukan yaitu cukup dengan mengganti model, warna

serta *style* pada halaman CSS, sehingga secara otomatis halaman yang menggunakan CSS akan berubah.

CSS merupakan konsep sederhana yang berfungsi untuk membuat *style* atau gaya yang lebih diprioritaskan kepada bagaimana data yang ada pada HTML ditampilkan ke *browser*. CSS akan membuat suatu format pada teks yang dibuat pada halaman tersendiri, kemudian dengan menggunakan *link* maka akan dapat digunakan secara bersama-sama. *Update* akan mudah dilakukan karena hanya cukup mengganti model ataupun warna serta gaya lain pada halaman CSS tersebut sehingga secara otomatis halaman yang menggunakannya akan berubah, biasanya digunakan untuk menciptakan suatu set gaya yang dapat diaplikasikan pada font, table dan atribut *web* sehingga memudahkan dalam melakukan perubahan secara global.

Dalam memberikan CSS, tiap halaman yang terkait harus diberikan *keyword* CSS untuk menjalankan perintah yang ada di dalam CSS. Banyak hal yang dapat diatur dalam CSS, seperti menyesuaikan font dan warna atribut, spasi (pengaturan jarak) serta *alignment*. Dalam CSS, hal penting yang harus diperhatikan adalah penempatan *keyword* CSS di dalam *script* dan hubungan link-linknya. Dalam membuat eksternal CSS dan menghubungkannya dari halaman *web*, dapat menggunakan *style file* seperti (`<link href= "style.css" rel= "stylesheet" type= "text/css">` ). Dengan menyesuaikan atribut dalam suatu *eksternal* yang dihubungkan dengan CSS file, dapat meminimalkan halaman *web* dan lebih memudahkan dalam pengupdatean.

(sumber : <http://www.devarticles.com/c/a/style-sheets/the-power-of-css.htm> )

## II.8 PHP

Pada sub-bab ini akan dijelaskan mengenai definisi PHP, sejarah perkembangan PHP, dan kelebihan menggunakan PHP.

### II.8.1 Definisi PHP

PHP merupakan singkatan dari *Personal Home Page*, yaitu *script* untuk pemrograman web yang bekerja disisi *server* dan mampu untuk membuat dokumen HTML secara *on the fly*, dalam arti dokumen HTML yang dihasilkan berasal dari suatu aplikasi, bukan berasal dari dokumen HTML yang dibuat dengan menggunakan *teks editor* atau *editor HTML*.

Bahasa ini mempunyai kelebihan yaitu kompatibilitasnya dengan berbagai macam jenis *database*, dukungan dengan berbagai macam jenis sistem operasi. PHP lebih cocok dan umum digunakan jika digabungkan dengan *database mysql*. *Mysql* dan PHP seakan-akan dua hal yang tidak dapat dipisahkan.

### II.8.2 Sejarah PHP

PHP pertama kali ditulis oleh Rasmus Lerdorf pada tahun 1994 dengan tujuan untuk membuat *tool* pelacak data pengguna yang telah membaca daftar riwayat hidupnya. Pada tahun 1995, Rasmus membuat FI (*Form Interpreter*), yaitu *tool* untuk memasukkan *query SQL* secara mudah ke dalam halaman *web*.

Berikut ini adalah perintah SQL yang sering digunakan:

- a. *SELECT*, menyatakan kolom, konstanta dan ekspresi yang akan ditampilkan sebagai hasil *query*.
- b. *FROM*, berisikan tabel-tabel yang berisi data yang akan ditampilkan.
- c. *WHERE*, menyatakan syarat dari hasil *query*.
- d. *INSERT*, digunakan untuk menambah baris tunggal pada tabel.
- e. *DELETE*, untuk menghapus satu atau lebih baris dari tabel.
- f. *UPDATE*, untuk memperbaiki satu atau lebih baris didalam tabel.

MySQL merupakan sebuah *Database Management System* atau manajemen basis data yang digunakan untuk menambah, menghapus, mengambil nilai, dan memproses data yang disimpan di dalam suatu basis data komputer. Kemampuan komputer untuk menangani jumlah data yang sangat besar menjadikan manajemen basis data memegang peranan yang penting dalam komputasi.

Manajemen basis data dapat digunakan dalam perangkat lunak *stand alone* maupun sebagai bagian dari sebuah perangkat lunak.

MySQL termasuk jenis *Relational Database Management System* atau basis data relasional yang menyimpan data dalam tabel - tabel terpisah sehingga akan menambah kecepatan dan fleksibilitas pengaksesan data. Itulah sebabnya istilah seperti tabel, baris, dan kolom digunakan pada MySQL. Pada MySQL, sebuah basis

data mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom. Tabel - tabel yang ada dihubungkan dengan mendefinisikan relasi-relasi sehingga memungkinkan untuk mengkombinasikan data dari beberapa tabel.

MySQL merupakan perangkat lunak yang bersifat *Open Source*, yang berarti memperbolehkan bagi semua orang untuk menggunakan dan memodifikasinya. MySQL dapat digunakan oleh semua orang dan *source code*-nya dapat dipelajari untuk dimodifikasi sesuai kebutuhan si pengguna tanpa perlu membayar.

#### **II.9.2 Kelebihan Menggunakan MySQL**

Terdapat beberapa kelebihan yang dapat diperoleh jika menggunakan MySQL antara lain performa yang diberikan MySQL sangat cepat, stabil, dan mudah digunakan, MySQL juga memiliki sekumpulan fitur - fitur yang dikembangkan untuk memenuhi kebutuhan pemakai. Konektivitas, kecepatan, dan keamanan yang diberikan oleh MySQL menjadikannya sangat cocok untuk mengakses basis data di Internet. (Abdul Kadir, 2002)

#### **II.10 Apache Web Server**

*Web server* merupakan server yang diperlukan untuk melaksanakan proses transaksi antara *user* atau *computer* pengguna dan juga server. Pada awalnya versi pertama Apache ini hanya bisa berjalan di lingkungan UNIX, tetapi kini telah ada versi yang dapat beroperasi di bawah platform OS/2, Windows dan sebagainya. (Onno, 2000)

Apache merupakan *web server* yang paling banyak digunakan saat ini. Menurut *survey netcraft*, Apache digunakan pada lebih dari 62% *web server* di Internet. Hal ini disebabkan oleh beberapa sebab, diantaranya adalah karena sifatnya yang *open source* atau bisa didownload secara gratis dari situs-situs yang mengulas Apache. Diantaranya dengan konektivitasnya dengan *database server* melalui bahasa scripting PHP.

### II.11 Macromedia Dreamweaver MX

Macromedia Dreamweaver merupakan *software web design* yang menyediakan banyak obyek *website* dan alat-alat bantu untuk menciptakan *script client side* serta menciptakan efek-efek *website* yang dinamis, seperti *rollover images*, animasi *layer*, *hidden menu* dan yang lainnya dalam waktu yang singkat dan tanpa pengetahuan *script client side*. *Software* ini memang sangat unggul dalam teknologi dinamisnya, sebab memiliki fasilitas untuk menangani *script client side*.

### II.12 AJAX (Asynchronous JavaScript dan XML / WEB 2.0)

*Asynchronous JavaScript* dan XML (AJAX) adalah teknik pembangunan *web* untuk membuat aplikasi *web* interaktif menggunakan paduan beberapa teknologi sebagai berikut:

- a. HTML (*HyperText Markup Language*) digunakan dalam membuat halaman *web* dan dokumen-dokumen lain yang dapat ditampilkan dalam *browser*. HTML merupakan standar internasional dengan spesifikasi yang

ditetapkan oleh *World Wide Web Consortium (W3C)*. Versi terakhir saat tulisan ini dibuat adalah HTML 4.01.

- b. *Extensible HyperText Markup Language (XHTML)*, adalah bahasa penanda (*markup*) sebagaimana HTML, tetapi dengan gaya bahasa lebih baik.
- c. *Cascading Style Sheets (CSS)* adalah sebutan bahasa *stylesheet* yang digunakan untuk memaparkan presentasi sebuah dokumen yang ditulis dalam HTML atau XML (termasuk beberapa variasi bahasa XML seperti XHTML dan SVG).
- d. JavaScript adalah bahasa *scripting* kecil, ringan, berorientasi-objek dan lintas *platform*. JavaScript tidak dapat berjalan dengan baik sebagai bahasa mandiri, melainkan dirancang untuk ditanamkan pada produk dan aplikasi lain seperti *browser web*.
- e. DOM (*Document Object Model*) adalah sebuah API (*Application Program Interface*) untuk dokumen HTML dan XML. DOM menyediakan representasi dokumen secara terstruktur, dimungkinkan untuk merubah isi dan presentasi *visual*. Pada dasarnya, DOM menghubungkan halaman *web* dengan *script* atau bahasa pemrograman.
- f. XML (*Extensible Markup Language*) adalah bahasa penanda untuk keperluan umum yang disarankan oleh W3C untuk membuat bahasa penanda keperluan khusus. Keperluan utama XML adalah untuk membagi data antar sistem yang berbeda, sebagian sistem terhubung melalui Internet.
- g. *Extensible Stylesheet Language Transformations (XSLT)* adalah sebuah bahasa berbasis-XML untuk

transformasi dokumen XML. Walaupun proses merujuk pada transformasi, dokumen asli tidak berubah melainkan dokumen XML baru dibuat dengan basis isi dokumen yang sudah ada. XSLT biasanya digunakan untuk merubah skema XML ke halaman web.

- h. Objek *XMLHttpRequest* untuk melakukan pertukaran data secara *asinkron* dengan *server web*. Beberapa kerangka-kerja *AJAX* dan dalam beberapa situasi, objek *IFrame* digunakan selain objek *XMLHttpRequest* untuk melakukan pertukaran data dengan *server web*.

Perpaduan teknologi-teknologi tersebut dapat meningkatkan kinerja aplikasi web dan lebih responsif terhadap aksi pengguna sehingga dapat berjalan lebih cepat, pemutakhiran antarmuka pengguna secara berkala tanpa memuat kembali keseluruhan halaman.