

## BAB 2 LANDASAN TEORI

### 2.1 Web Desktop

*Web Desktop* atau lebih dikenal dengan sistem operasi berbasis web (*webOS*) adalah sebutan populer untuk sistem aplikasi berbasis web yang dapat dijalankan melalui jaringan internet tanpa harus menginstall aplikasi tersebut pada komputer. *Web desktop* yaitu *virtual desktop* yang berjalan secara *online* di atas jaringan *internet* dengan menggunakan *browser*. Jika *desktop* berjalan secara *offline* di sebuah komputer, maka *web desktop* beroperasi secara *online* dan berjalan beriringan dengan *browser internet*. *Web desktop* diakses melalui komputer dan *notebook* yang terhubung dengan *internet*.

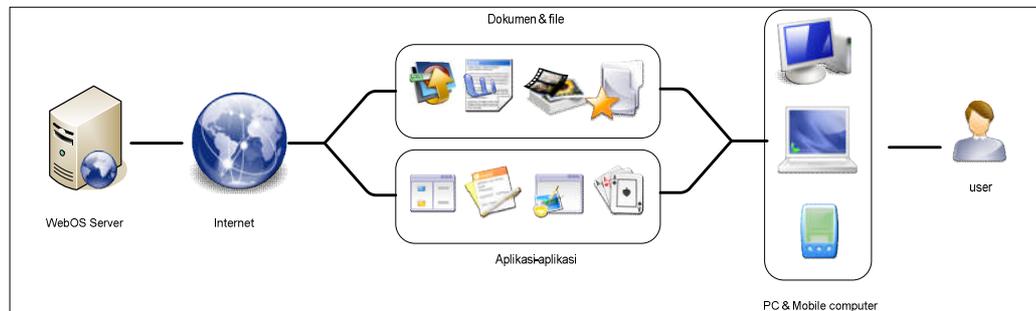
Tampilan utama dari *web desktop* atau sistem operasi berbasis web seperti *desktop personal computer* (PC) (tampilan *desktop* yang berjalan di atas web). *Web desktop* bukanlah sebuah *desktop personal computer* (PC) yang sebenarnya, *web desktop* hanyalah meniru tampilan, *look-and-feel*, dan beberapa fitur yang biasanya ditemukan pada sebuah *desktop personal computer* (PC). (*List of The Most Great Web operating systems (webtop)*, [www.ntt.cc](http://www.ntt.cc) dan dan *WebOS-Sistem Operasi Berbasis Web*, [www.Konsultasi Komputer dan Internet Online.com](http://www.KonsultasiKomputer.com), 2009)

*Web desktop* mencoba memberikan layanan fungsional baik untuk manajemen proses, manajemen berkas atau direktori, bahkan manajemen pemakainya. *Web desktop*

memiliki kemampuan definisi GUI (*Graphical User Interface*) yang menarik dengan menggunakan *javascript*, *HTML*, *CSS*, dan teknologi *Web 2.0*, sehingga menjadikan aplikasi *web desktop* lebih menawan. *Web desktop* memiliki menu-menu dan icon menu aplikasi yang lengkap, semua aplikasi yang tersedia bisa langsung diakses dari *web desktop*. Aplikasi *mailer* serupa *Outlook Express*, kalender, *bookmark* web, *web publishing*, *RSS feed*, *web browsing*, pengolah tabel, kalkulator, *chat*, penyusun presentasi, pemutar musik, pemutar video dengan dukungan aneka format video, pengolah foto, dan tentu saja aplikasi pengolah kata. Aplikasi Pengolah kata atau *word*, tidak berbeda dengan aplikasi pengolah kata pada umumnya, pengolah kata ini memiliki fasilitas pengelolaan huruf, format, pemeriksa ejaan, hingga fasilitas *thesaurus*. Setiap dokumen secara *default*, disimpan dalam format *HTM*, namun bisa juga diubah menjadi format *DOC*, *PDF*, *RTF*, dan *TXT*. *Web desktop* juga menyediakan *space* untuk menyimpan semua data yang dibutuhkan pada *server*. Beberapa contoh *web desktop* yang sudah dibuat yaitu *eyeOS*, *desktoptwo*, *Glide OS*, *youos*, dan *CorneliOS*.

*Web desktop* atau yang lebih dikenal dengan sistem operasi berbasis web (*web-based operating system*) merupakan salah satu jenis aplikasi web terkini yang memanfaatkan kemampuan jaringan dan bahasa pemrograman *internet* dalam mengimplementasikan sebuah desktop atau sistem operasi pada *website*. *Web desktop* atau sistem operasi berbasis web juga diyakini akan mampu menggeser kepopuleran sistem operasi masa kini. Arsitektur *web*

*desktop* seperti ditunjukkan pada gambar 2.1 berikut ini.



**Gambar 2.1 Arsitektur Desktop Web**

Menggunakan aplikasi *web desktop* atau sistem operasi berbasis web memiliki keuntungan dan kerugian. (*List of The Most Great Web operating systems (webtop)*, [www.ntt.cc](http://www.ntt.cc), 2009 dan *WebOS-Sistem Operasi Berbasis Web*, [www.Konsultasi Komputer dan Internet Online.com](http://www.Konsultasi Komputer dan Internet Online.com), 2009) yaitu:

1. Keuntungan menggunakan *web desktop* atau sistem operasi berbasis web :
  - a. Data yang disimpan dalam sistem operasi jenis ini dapat diakses dari mana saja dari seluruh dunia.
  - b. Data yang disimpan juga dapat diakses oleh orang lain pada saat yang sama (*multiple file sharing*).
  - c. Sangat berguna apabila kita sering mengakses *internet* dari warung *internet* namun sistem operasi yang digunakan tidak familiar.
  - d. Sebagian besar sistem operasi berbasis web menyediakan fasilitas enkripsi khusus untuk setiap pelanggannya apabila mereka hendak menyimpan *file* mereka di sini.

- e. Tidak perlu direpotkan tentang masalah instalasi *software* ataupun *update* karena sistem operasi berbasis web telah menyediakannya untuk anda.
  - f. Sistem operasi berbasis web aman dari virus dan *threat* jahat lainnya.
  - g. Tidak diperlukan pengetahuan mendalam tentang sistem operasi untuk menggunakan sistem operasi berbasis web.
2. Kerugian menggunakan *web desktop* atau sistem operasi berbasis web :
- a. Karena berbasis web, sistem operasi ini hanya akan efektif penggunaannya melalui akses *internet*. Besarnya volume data seringkali menyebabkan beberapa sistem operasi *virtual* mengalami *crash/hang*, disamping akses ke *desktop* yang juga lambat.
  - b. Data yang disimpan dalam ruang *virtual* sistem operasi dapat sewaktu-waktu hilang apabila sistem operasi yang bersangkutan menghilang dari dunia maya.
  - c. Ada beberapa sistem operasi yang menggunakan elemen *flash* untuk *desktop environment* dan navigasinya sehingga cukup menyulitkan pengguna yang kurang begitu memahami karakteristik elemen *flash*.
  - d. Tidak semua aplikasi berjalan secara lancar. Ada beberapa aplikasi yang mungkin akan mengalami *lag/crash* dalam penggunaannya mengingat sistem operasi jenis ini juga dipengaruhi oleh jenis *browser* yang digunakan.

e. Minimnya jenis aplikasi yang kompatibel dengan sistem operasi berbasis web yang bersangkutan.

## 2.2 Web 2.0

*Web 2.0* lahir dari hasil wacana antara *O'Reilly* dan *MediaLive International*, *Web 2.0* makin hari makin bergulir, hanya dalam satu setengah tahun, lebih dari 9,5 juta halaman web dicatat *Google* mengandung nama ini. Menurut definisi oleh *Tim O'Reilly* (pendiri dari *O'Reilly Media*), *Web 2.0* adalah sebuah revolusi bisnis pada industri komputer yang dikarenakan oleh perpindahan internet menjadi sebuah *platform*. Contoh untuk mempermudah pemahaman konsep *Web 2.0* adalah dengan melihat contoh perbandingan antara *website* konvensional (*Web 1.0*) dengan *website* yang telah termasuk kedalam *Web 2.0*.

Tabel 2.1 Evolusi Web 1.0 ke Web 2.0

Web 1.0		Web 2.0
<i>DoubleClick</i>	->	<i>Google AdSense</i>
<i>Ofoto</i>	->	<i>Flickr</i>
<i>Akamai</i>	->	<i>BitTorrent</i>
<i>mp3.com</i>	->	<i>Napster</i>
<i>Britannica Online</i>	->	<i>Wikipedia</i>
<i>personal websites</i>	->	<i>Blogging</i>
<i>Evite</i>	->	<i>upcoming.org</i> and <i>EVDB</i>
<i>domain name speculation</i>	->	<i>search engine optimization</i>
<i>page views</i>	->	<i>cost per click</i>
<i>screen scraping</i>	->	<i>web services</i>
<i>Publishing</i>	->	<i>Participation</i>
<i>content management systems</i>	->	<i>Wikis</i>
<i>directories (taxonomy)</i>	->	<i>tagging ("folksonomy")</i>
<i>Stickiness</i>	->	<i>syndication</i>

Pada era booming *dot com*, sedemikian banyak orang yang membuat *website* pribadi sehingga halaman pertamanya rata-rata menuliskan "*Welcome to My Personal Homepage*", tidak lupa menambahkan fasilitas buku tamu

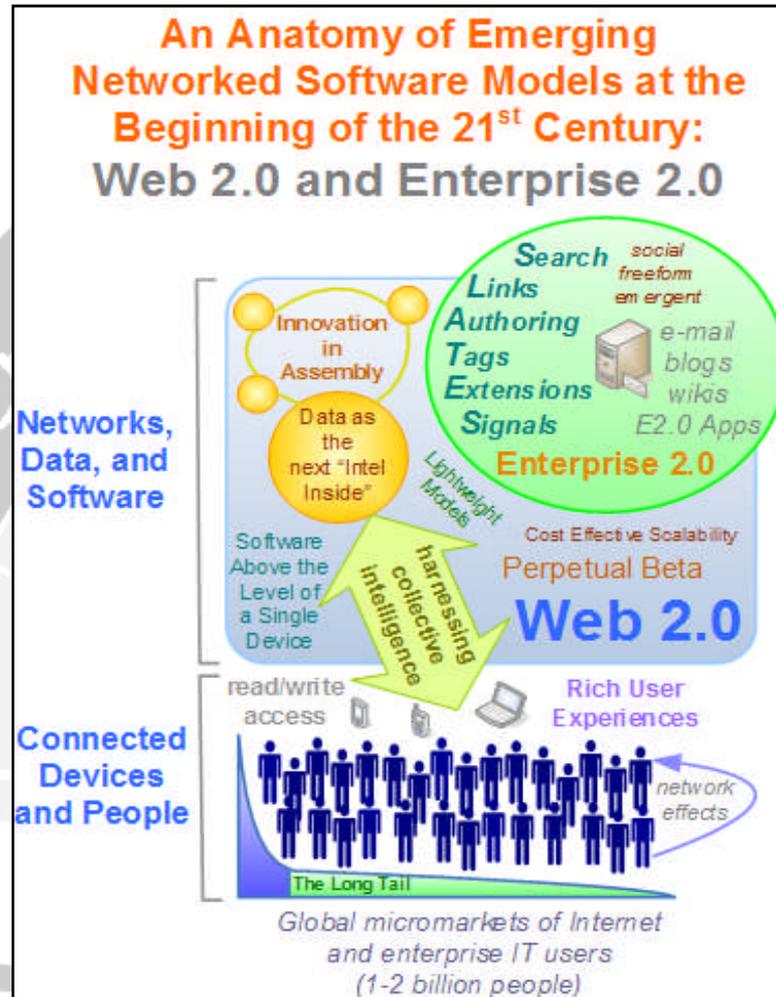
dan *web counter*. Dengan perubahan kearah konsep Web 2.0, website-website pribadi itu menjelma menjadi blog. Meski diterima banyak kalangan, tak sedikit yang mencibir bahwa istilah ini cuma kerjaan orang teknik yang sedang tertarik dengan teknologi sehingga melahirkan kategorisasi yang bodoh dan tidak dipahami publik. Perdebatan akan masih terus berlanjut. Para pelaku bisnis *internet* memahami bahwa tahun 2001 adalah tahun kelam *dotcom*. Saat itu memang sebagian besar *dotcom* yang menghiasi pentas bisnis tiga tahun sebelumnya dan menjadi primadona investasi dunia tiba-tiba rontok bertumbangan dan mati. *Dotcom boom* berubah menjadi *Dotcom crash* atau *dotcom doom*. Banyak yang menangis terutama para investor. Namun di balik kehancuran itu *O'Reilly* dan *MediaLive International* melihat bahwa masih ada *dotcom* yang lolos dari masa kritis tersebut. Setelah dianalisa, *dotcom* tersebut memiliki ciri yang sama. Dan ciri-ciri itu tidak dimiliki oleh para mantan *dotcomers*. Apakah kehancuran *dotcomers* lama dan lahirnya jenis *dotcomer* baru menandai lahirnya generasi baru web? Begitulah pertanyaan *Dale Dougherty*. Mungkin pionir web dan VP *O'Reilly* ini terinspirasi oleh proses seleksi alam *Charles Darwin*: begitu ada generasi yang punah, akan muncul generasi baru yang lebih tangguh. Untuk mempermudah kategorisasi, *Dougherty* menyebut generasi baru itu dengan sebutan *Web 2.0*. (*Web 2.0, www.pcmmedia.com, 2009*)

Meskipun sudah menjadi pembicaraan sejak tahun 2004, *Web 2.0* bukanlah kata yang umum bagi pengguna *internet*. Masih banyak pengguna yang mempertanyakan

maksud dan manfaat dari penggunaan *Web 2.0*, terutama jika dibandingkan dengan web yang telah mereka kenal selama ini. Ketika *Web 2.0* disebut sebagai tahap kedua dari perkembangan web yang telah ada saat ini, muncul kekhawatiran akan tidak kompatibelnya versi web tersebut dengan program *web browser* yang dimiliki pengguna. Padahal tidak ada satupun teknologi di sisi pengguna (*client*) yang perlu *upgrade* untuk dapat mengakses web tersebut. Perkembangan *Web 2.0* lebih menekankan pada perubahan cara berpikir dalam menyajikan konten dan tampilan di dalam sebuah *website*. Sebagian besar cara berpikir tersebut mengadaptasi gabungan dari teknologi web yang telah ada saat ini. Walaupun definisi mengenai *Web 2.0* masih belum secara utuh diformulasikan. Sampai hari ini, ada pihak yang mengatakan bahwa *Web 2.0* lebih menekankan pada *social network* atau jalinan sosial antara penggunanya seperti yang telah kita lihat selama ini dalam dunia *Blog*. Dengan adanya *RSS (Really Simple Syndication)* adalah teknologi penyampai berita) di dalam *Blog*, informasi-informasi di dalam sebuah *Blog* dimungkinkan dapat diadaptasi, dikoleksi, dan di *share* untuk menjadi bagian dari *Blog* lainnya.

*O'Reilly* dan *MediaLive International* menekankan bahwa *Web 2.0* merupakan sebuah *platform* bagi aplikasi. Mereka mendeskripsikan hal ini sebagai sebuah *software* yang berjalan melalui media *internet* dengan bantuan *web browser* dan tidak perlu diinstalasi terlebih dahulu seperti *software-software* yang umumnya kita gunakan sehari-hari. Bahkan konsep mengenai sistem operasi di

dalam web juga masuk dalam definisi tersebut di dalam konferensi *Web 2.0* pada tahun 2005.



Gambar 2.2 Network Software Model web 2.0

*Web 2.0* merupakan teknologi web yang menyatukan teknologi-teknologi yang dimiliki dalam membangun web. Penyatuan tersebut merupakan gabungan dari *HTML*, *CSS*, *JavaScript*, *XML*, dan tentunya *AJAX*. Perkembangan *web 2.0* lebih menekankan pada perubahan cara berpikir dalam menyajikan konten dan tampilan di dalam sebuah *website*. Dalam perkembangannya *Web 2.0* diaplikasikan sebagai bentuk penyajian halaman web yang bersifat sebagai program *desktop* pada umumnya seperti *Windows*. Fungsi-

fungsi pada penerapannya sudah bersifat seperti *desktop*, seperti *drag and drop*, *auto-complete*, serta fungsi lainnya. Aplikasi *Web 2.0* disajikan secara penuh dalam suatu *web browser* tanpa membutuhkan teknologi perangkat yang canggih dari sisi pengguna. Tidak mengherankan bila suatu aplikasi (*software*) dapat diakses secara *online* tanpa harus menginstalnya terlebih dahulu. Permasalahan manajemen *file* juga tidak merepotkan, bahkan *file* dapat disimpan dan juga dapat *sharing file* dengan pengguna lain. Kemudahan berinteraksi antara pengguna dengan sistem merupakan tujuan dibangunnya teknologi *Web 2.0*. Interaksi tersebut tentunya haruslah diimbangi dengan kecepatan untuk mengakses, oleh karena itu diperlukan suatu *bandwith* yang cukup untuk *loading* data. *Loading* data tersebut dilakukan saat pertama kali membuka situs, data-data tersebut antara lain *CSS*, *JavaScript*, dan *XML*.

Salah satu karakteristiknya adalah adanya dukungan pada pemrograman yang sederhana dan ide akan *web service* atau *RSS*. Ketersediaan *RSS* akan menciptakan kemudahan untuk dimodifikasi oleh *website* lain dengan menggunakan tampilannya masing-masing dan dukungan pemrograman yang sederhana. Adanya kemajuan inovasi pada antarmuka di sisi pengguna merupakan karakter dari *Web 2.0*. Dukungan *AJAX* yang menggabungkan *HTML*, *CSS*, *Javascript*, dan *XML* pada *Yahoo!Mail Beta* dan *Gmail* membuat pengguna merasakan nilai lebih dari sekedar situs penyedia *email*. Kombinasi media komunikasi seperti *Instant Messenger (IM)* dan *Voice over IP (VoIP)*

akan semakin memperkuat karakter *Web 2.0* di dalam situs tersebut.

Majalah *Information Week* (09/2007) memberi definisi "*Web 2.0 is all the Web sites out there that get their value from the actions of users*". Jadi inti dari sebuah *Web 2.0* adalah keterlibatan pengguna dan kecanggihan web tersebut dalam kolaborasi, interaksi dan melayani pengguna. Contohnya adalah usaha pembaca dalam berkomentar dan *trackbacks* di *blog*. Dari sisi web adalah adanya *RSS* di dalam *blog* dimana semua informasi *blog* dimungkinkan dapat diadaptasi, dikoleksi, dan di *share* untuk menjadi bagian dari *blog* lainnya. Pengguna yang paling awam pun tidak hanya sekedar bisa membaca halaman web tetapi juga bisa buat dan merespon sebuah *content*. Mau komentar mudah, mau bikin *blog* mudah, mau jadi anggota dan menambah teman di *Friendster* mudah. Tidak perlu mengerti pemrograman atau web desain untuk melakukan itu semua. Jika *Web 1.0* adalah dunianya *web programmer* dan *web designer*, maka *Web2.0* adalah dunianya pengguna. Teknologi *Web 2.0* memiliki karakteristik-karakteristik sebagai berikut :

1. **Network as platform**, *Web 2.0* merupakan sebuah *platform* bagi aplikasi. Web sebagai *platform* dimana menjadikan web sebagai tempat bekerja di manapun berada. Cukup dengan membuka *web browser*, kita dapat mengerjakan tugas mengetik dokumen, perhitungan keuangan, atau merancang presentasi melalui aplikasi-aplikasi yang telah disediakan dan dapat dijalankan secara langsung melalui *internet*.

2. **Harnessing Collective Intelligence**, adanya partisipasi dari pengguna dalam berkolaborasi pengetahuan. Hal ini mengingatkan akan pemberian kepercayaan kepada pengguna *internet* untuk dapat berpartisipasi dalam berbagi pengetahuan di *Wikipedia*, sebuah *ensiklopedia* berbasis web yang disusun berdasarkan masukan-masukan pengguna *internet* di seluruh dunia.
3. **Data is the Next Intel Inside**, data menjadi *trademark* sebuah aplikasi, dimana penyuplai data akan memberikan *trademark* yang akan digunakan oleh pemilik *website* untuk memberikan garansi kepercayaan kepada pengunjunnya. Sebagai contoh *Portal Peta* dengan slogan "*Powered by Google*" memakai dukungan *Google Maps* pada peta dunia berbasis web.
4. **End of the Software Release Cycle**, *Web 2.0* sebagai akhir dari siklus peluncuran produk *software*, mengilustrasikan setiap produsen *software* tidak lagi meluncurkan produknya dalam bentuk fisik. Karena web menjadi *platform*, pengguna cukup datang ke *website* untuk menjalankan aplikasi yang ingin mereka gunakan. Hasil dari pengembangan fitur di dalam *software* dapat langsung dirasakan oleh pemakai. *Software* tidak lagi dijual sebagai produk namun berupa layanan (*service*). Produsen yang memberikan pelayanan yang cepat dan bagus, akan menjadi pilihan pengguna.
5. **RSS & XML Technology**, dukungan pada pemrograman yang sederhana dan ide akan *web service* atau

*RSS*. Ketersediaan *RSS* akan menciptakan kemudahan untuk dimodifikasi oleh *website* lain dengan menggunakan tampilan sendiri dan dukungan pemrograman yang sederhana.

**6. Software Tidak Terbatas pada Perangkat Tertentu**, hal ini mempertegas posisi web sebagai *platform* dimana setiap perangkat dapat mengaksesnya. Komputer tidak lagi menjadi satu-satunya perangkat yang dapat menjalankan berbagai aplikasi di *internet*. Setiap aplikasi harus didesain untuk dapat dinikmati komputer, *notebook*, *ponsel*, *PDA*, dan lain sebagainya.

**7. Rich User Experiences**, adanya kemajuan inovasi pada antarmuka (*interface*) di sisi pengguna. Dukungan *AJAX* yang menggabungkan *HTML*, *CSS*, *Javascript*, dan *XML* pada *Yahoo!Mail*, *Skype* dan *Gmail* membuat pengguna merasakan nilai lebih dari sekedar situs penyedia *email*.

## **2.1 Java Script**

*JavaScript* adalah nama implementasi *Netscape Communications Corporation* untuk *ECMA Script* standar, suatu bahasa skrip yang didasarkan pada konsep pemrograman berbasis *prototype*. Bahasa ini terutama terkenal karena penggunaannya di situs web (sebagai *JavaScript* sisi *client*) dan juga digunakan untuk menyediakan akses skrip untuk objek yang ditenamkan (*embedded*) di aplikasi lain. Walaupun memiliki nama serupa, *JavaScript* hanya sedikit berhubungan dengan bahasa pemrograman *Java*, dengan kesamaan utamanya adalah penggunaan *sintaks C*. Secara

semantik, *JavaScript* memiliki lebih banyak kesamaan dengan bahasa pemrograman *Self*.

*JavaScript* adalah merek dagang yang terdaftar dari *Sun Microsystems, Inc.* Bahasa ini digunakan di bawah lisensi untuk teknologi yang diciptakan dan diimplementasikan oleh *Netscape Communications* dan entitas barunya seperti *Mozilla Foundation*.

(*Javascript*, [id.wikipedia.org](http://id.wikipedia.org), 2007)

*Javascript* memungkinkan pengembang *Website* untuk mendesain situs yang interaktif. *Javascript* dapat berinteraksi dengan sumber kode *HTML*, yang memungkinkan pengarang web untuk memperindah *website* mereka dengan isi yang dinamis. *JavaScript* didukung oleh berbagai perusahaan *software* dan merupakan suatu bahasa yang terbuka dan dapat digunakan oleh semua orang tanpa perlu untuk membeli ijin untuk menggunakannya. *Javascript* juga didukung dengan *browser* terbaru dari *Netscape* dan *Microsoft*, meskipun *Internet Explorer* hanya mendukung *subset*, yang disebut *Microsoft* sebagai *Jscript*. (*Javascript*, [openwebmail.org](http://openwebmail.org), 2007)

*JavaScript* adalah sebuah bahasa *script* yang juga dipakai untuk membangun sebuah web dan dijalankan pada *browser web*. Banyak sekali penggunaan *tag Java Script* yang tidak memerlukan basis data dan dapat membuat lebih menarik tampilan halaman web. Misalnya, ucapan selamat datang, tulisan-tulisan di menu *window*, maupun di dalam isi situs web. Menurut *Anggawirya (2002)*, *tag Java Script* memiliki kemampuan mengakses semua tampilan *browser* dengan mudah dan aman. Selain itu, *Java Script* dapat menggabungkan dokumen-dokumen *HTML* yang ditulis dengan *Java* dan *ActiveX*, dan merupakan bahasa yang

digunakan oleh basis data *LiveWire Netscape* untuk membuat hubungan antar *server web* dan basis data *server* eksternal.

### 2.3 XML

*XML* adalah sebuah teknologi *cross platform*, dan merupakan *tool* untuk melakukan transmisi informasi. *XML* bukanlah program, atau pustaka, *XML* adalah sebuah teknologi atau sebuah bahasa *markup*, yaitu bahasa yang berisikan kode-kode berupa tanda-tanda tertentu dengan aturan tertentu yang digunakan untuk format dokumen *text* dengan *tag* sendiri agar dapat dimengerti. *XML* menyediakan format untuk mendeskripsikan data terstruktur. *XML* menyebabkan isi data mudah dimengerti dan memungkinkan pendefinisian kumpulan *tag* yang tidak terbatas. Dalam pengertian yang sederhana, sebuah dokumen *XML* hanyalah sebuah *file text* biasa yang berisikan berbagai *tag* yang didefinisikan sendiri oleh pembuat dokumen *XML* tersebut. Sesuai dengan namanya, *eXtensible Markup Language*, sebuah dokumen *XML* adalah sebuah dokumen dengan *markup*, sama seperti halnya dengan *HTML*. *XML* tidak didesain untuk menggantikan *HTML*, *XML* lebih dirancang untuk mendeskripsikan data dan memfokuskan diri pada data tersebut. Sementara, *HTML* didesain untuk menampilkan data dan memfokuskan diri pada bagaimana data ditampilkan. Secara desain, hal ini sudah jauh berbeda. Dengan demikian, *XML* bukanlah pengganti *HTML* karena memang dirancang berbeda. Hubungan antara *XML* dan *HTML* lebih ke arah pelengkap. Anda dapat menyimpan data dalam sebuah dokumen *XML* dan mempergunakan *HTML* untuk menampilkan

data tersebut. Dokumen *XML* juga terdiri dari berbagai *tag*. Hanya, bedanya, *tag-tag* tersebut tidak memiliki standar khusus. Kita sebagai pembuat dokumen lah yang harus menentukan *tag* dan artinya. Untuk menjaga agar *tag-tag* tersebut tetap berada di dalam lingkup jalan yang benar, maka keseluruhan aturan *tag* kita disimpan di dalam *Document Type Definition (DTD)* atau *XML Schema*. Dengan adanya aturan tersimpan di *DTD* atau *XML Schema*, maka sebuah dokumen *XML* diharapkan akan mampu mendeskripsikan diri sendiri (*self descriptive*). Boleh disamakan dengan *tag HTML* yang telah memiliki standar, walaupun, pada *XML*, artinya akan lebih luas lagi.

*XML* dikembangkan oleh Kelompok Kerja *XML* (sebelumnya dikenal dengan nama *SGML Editorial Review Board*) yang dibentuk di bawah dukungan *World Wide Web Consortium (W3C)* pada tahun 1996. Kelompok ini dipimpin oleh *Jon Bosak* dari *Sun Microsystems* dengan peran serta aktif dari *XML Special Interest Group* (sebelumnya dikenal dengan nama Kelompok Kerja *SGML*) yang juga dikelola oleh *W3C*. Tujuan desain *XML* adalah :

1. *XML* harus dapat digunakan secara langsung di Internet.
2. *XML* harus mendukung secara luas berbagai aplikasi.
3. *XML* harus kompatibel dengan *SGML*.
4. Program yang memproses dokumen *XML* harus mudah dibuat.
5. Jumlah fitur opsional pada *XML* harus dibuat seminimum mungkin, idealnya nol.
6. Dokumen *XML* harus jelas dan mudah dipahami

manusia.

7. Desain *XML* harus disiapkan secepatnya.
8. Desain *XML* harus formal dan singkat.
9. Dokumen *XML* harus mudah dibuat.
10. Keringkasan suatu *markup XML* tidak dipentingkan.

*Extensible Markup Language (XML)* adalah sebuah himpunan bagian dari *SGML* yang dideskripsikan secara keseluruhan dalam dokumen ini. *XML* bertujuan agar *SGML* generik dapat melayani, menerima, dan memproses dalam web dengan cara seperti yang dimungkinkan *HTML* saat ini. *XML* didesain untuk kemudahan implementasi dan interoperabilitas dengan *SGML* maupun *HTML*. *XML* merupakan sebuah profil aplikasi atau bentuk terbatas *SGML*, *Standard Generalized Markup Language [ISO 8879]*. Jika dilihat dari susunannya, dokumen *XML* mengikuti aturan dokumen *SGML*. (*XML*, [www.w3.org](http://www.w3.org), 2007)

Dokumen *XML* terdiri dari satuan penyimpanan yang disebut entitas, yang memuat baik data terurai maupun data tak terurai. Data terurai terdiri dari karakter-karakter, dimana sebagian membentuk data karakter, dan sebagian membentuk *markup*. *Markup* mengkodekan deskripsi tata letak penyimpanan dokumen dan struktur logika. *XML* menyediakan mekanisme untuk menentukan batasan-batasan suatu tata letak penyimpanan dan struktur logika.

Sebuah modul perangkat lunak yang disebut *Prosesor XML* digunakan untuk membaca dokumen *XML* dan menyediakan akses terhadap isi dan struktur dokumen tersebut. (*XML*, [www.w3.org](http://www.w3.org), 2007)

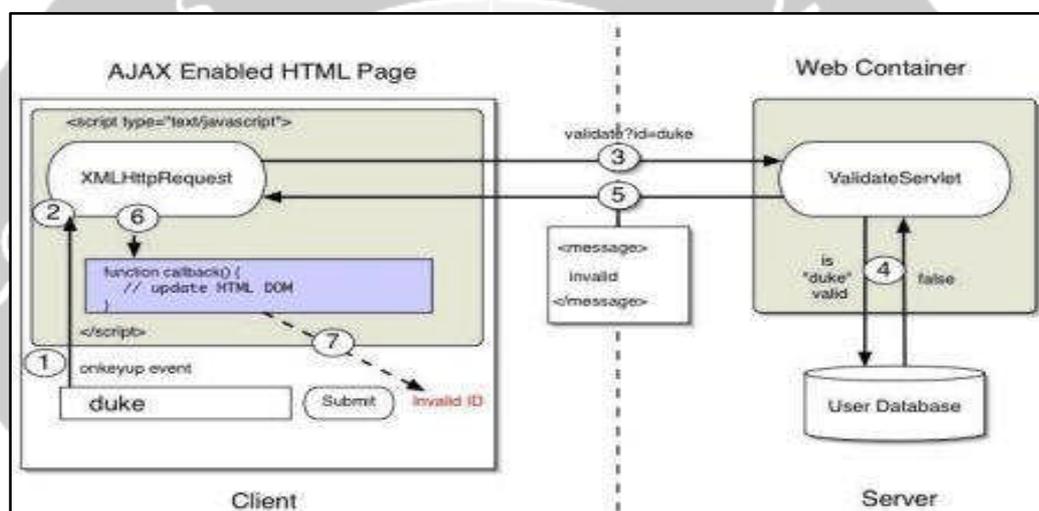
## 2.4 AJAX

AJAX diperkenalkan oleh seorang pakar pembuat program khususnya pemrograman berbasis web. Orang yang sangat berjasa tersebut tidak lain dan tidak bukan adalah *Rasmus Lerdorf*, dari tangan dingin *Rasmus Lerdorf* ini pulalah beberapa waktu yang lalu lahirlah *PHP*. Untuk mengaplikasikan AJAX dalam *website*, yang dibutuhkan adalah *browser* yang menyediakan layanan *Javascript*, dan komponen *XMLHTTP* bagi pengguna *Internet Explorer (IE)*, dan *XMLHttpRequest* untuk *Firefox*, *Safari*, *Opera* dan *browser* lainnya. Syarat-syarat tersebut hampir pasti sudah terpenuhi, mengingat pengguna *internet* banyak yang menggunakan *IE* dan *Firefox*.

Sebenarnya tidak ada hal baru dalam AJAX, karena yang digunakan adalah teknologi *Javascript*, yang sebenarnya sudah lama digunakan. Oleh karena itu, bagi yang sudah terbiasa dengan *Javascript*, akan mudah sekali mempelajari AJAX. Pembahasan ini tidak menekankan pada salah satu teknologi (*Javascript*, *AJAX*, *HTML*, *PHP*), akan tetapi keseluruhan dari hal tersebut. Hal ini karena penggunaan AJAX tidak hanya pada sisi *client (browser)*, tapi juga melibatkan respon dari *server*. Dengan menggunakan konsep ini, *page HTML* bisa membuat koneksi secara *asinkron* ke *server* dengan cara mengambil *XML* atau dokumen *text*. Selanjutnya *XML* atau dokumen *text* ini akan digunakan oleh *Javascript* untuk meng-*update* atau memodifikasi *Document Object Model (DOM)* pada *page HTML*.

Untuk selanjutnya teknologi ini juga dikenal *web remoting* atau *remote scripting*. *Developer Web* dapat

mengkombinasikan *plug in*, *java applets* atau *hidden frame* untuk mengemulasikan interaksi antara *page HTML* dengan dokumen *XML*. Dalam teknologi *Javascript* telah disediakan sebuah *object* yaitu *XMLHttpRequest*. Object ini sudah banyak disupport oleh banyak *browser* seperti *Internet Explorer*, *Firefox*, *Opera*, *Netscape* dan lain sebagainya. Ilustrasi interaksi *AJAX* dengan *server* ditunjukkan pada gambar 2.2 sebagai berikut.



Gambar 2.3 ilustrasi interaksi *AJAX* dengan *Server*

Konsep *AJAX* ini ide awalnya dikemukakan oleh *Rasmus Lerdorf*. *Rasmus Lerdorf* mengemukakan konsep *AJAX* ini dalam sebuah *mailing list* (*milist*). Berawal dari *milist* tersebut konsep *AJAX* kemudian dikenal dan dikembangkan oleh banyak orang. Berikut ini isi *milist* yang dikirimkan oleh *Rasmus Lerdorf*.

```
List:      php-general
Subject:   [PHP] Rasmus' 30 second AJAX Tutorial - [was Re: [PHP]
AJAX & PHP]
From:      Rasmus Lerdorf <rasmus () lerdorf ! com>
Date:      2005-07-21 22:50:56
Message-ID: 42E026D0.3090601 () lerdorf ! com
I find a lot of this AJAX stuff a bit of a hype. Lots of people
have
been using similar things long before it became "AJAX". And it
really
isn't as complicated as a lot of people make it out to be. Here
```

is a simple example from one of my apps. First the Javascript:

```
function createRequestObject() {
  var ro;
  var browser = navigator.appName;
  if(browser == "Microsoft Internet Explorer"){
    ro = new ActiveXObject("Microsoft.XMLHTTP");
  }else{
    ro = new XMLHttpRequest();
  }
  return ro;
}
var http = createRequestObject();
function sndReq(action) {
  http.open('get', 'rpc.php?action='+action);
  http.onreadystatechange = handleResponse;
  http.send(null);
}
function handleResponse() {
  if(http.readyState == 4){
    var response = http.responseText;
    var update = new Array();
    if(response.indexOf('|' != -1)) {
      update = response.split('|');
      document.getElementById(update[0]).innerHTML =
update[1];
    }
  }
}
```

This creates a request object along with a send request and handle response function. So to actually use it, you could include this js in your page. Then to make one of these backend requests you would tie it to something. Like an onclick event or a straight href like this:

```
<a href="javascript:sndReq('foo')">[foo]</a>
```

That means that when someone clicks on that link what actually happens is that a backend request to `rpc.php?action=foo` will be sent. In `rpc.php` you might have something like this:

```
switch($_REQUEST['action']) {
  case 'foo':
    / do something /
    echo "foo|foo done";
    break;
  ...
}
```

Now, look at `handleResponse`. It parses the `"foo|foo done"` string and splits it on the `'|'` and uses whatever is before the `'|'` as the element id in your page and the part after as the new `innerHTML` of that element. That means if you have a div tag like this in your

```

page:
  <div id="foo">
  </div>
Once you click on that link, that will dynamically be changed to:
  <div id="foo">
  foo done
  </div>
That's all there is to it. Everything else is just building on
top of
this. Replacing my simple response "id|text" syntax with a
richer XML
format and make the request much more complicated as well.
Before you
blindly install large "AJAX" libraries, have a go at rolling your
own
functionality so you know exactly how it works and you only make
it as
complicated as you need. Often you don't need much more than
what I
have shown here.
Expanding this approach a bit to send multiple parameters in the
request, for example, would be really simple. Something like:
function sndReqArg(action, arg) {
  http.open('get', 'rpc.php?action='+action+'&arg='+arg);
  http.onreadystatechange = handleResponse;
  http.send(null);
}
And your handleResponse can easily be expanded to do much more
interesting things than just replacing the contents of a div.

-Rasmus

```

Atas dasar konsep yang dikemukakan oleh Rasmus Lerdorf tersebut, penulis mencoba untuk mengembangkan lebih lanjut bagaimana memanfaatkan konsep AJAX untuk membangun aplikasi web yang fleksibel dan interaktif. AJAX bukan hanya menjadi solusi akan permasalahan pentingnya *update* halaman web. Beberapa keuntungan lain yang didapatkan dari penggunaan AJAX adalah :

1. **User Interface**, adanya proses *refresh* pada halaman web tradisional mengindikasikan sebuah proses yang berat. Berbeda jika Anda menggunakan AJAX, di mana halaman dapat di-*update* secara otomatis, tanpa melakukan *reload/refresh*. Hal ini tentu saja

mempercepat respon halaman terhadap interaksi dari pengguna.

2. **Always Visible**, sebuah *control* yang mengizinkan Anda meletakkan sebuah informasi (misalnya pengumuman, tanggal dan jam, dan seterusnya) pada suatu posisi yang tetap di halaman web, walaupun halaman tersebut di-*scroll* atau di-*resize*.
3. **Animation**, untuk membuat animasi pada suatu event tertentu, seperti *OnClick*, *OnLoad*, *OnMouseOver*, dan lain-lain.
4. **CascadingDropDown**, teknik yang cukup umum digunakan dalam *AJAX*, yaitu memuat data *dinamisProperty* status memiliki lebih banyak kondisi, beberapa yang terutama antara lain adalah di dalam sebuah *dropdown*.
5. **NumericUpDown**, banyak cara untuk memilih sebuah angka. Pada aplikasi *desktop*, Anda tentu telah terbiasa memilih angka dengan cara mengklik tombol *up* untuk menaikkan nilai pada sebuah *textbox*, atau klik tombol *down* untuk menurunkan nilai. Contohnya saat menentukan *margin* pada aplikasi *Microsoft Word*. Pada halaman web tradisional, Anda dapat menentukan angka dengan sebuah *dropdown*, ataupun mengetikkan secara manual angka tersebut. Jika ingin berkreasi, *control NumericUpDown* ini dapat menciptakan tombol *up* dan *down* yang serupa dengan aplikasi *desktop*.
6. **UpdateProgress**, salah satu *control* untuk mengatasi permasalahan waktu tunggu pada *AJAX*. Saat proses *behind the scene* terjadi, mungkin Anda perlu menampilkan sebuah status *progress* yang terjadi,

sehingga pengguna menyadari bahwa halaman web sedang bekerja dan memanggil data.

Disamping terdapat keuntungan dalam teknologi *AJAX*, terdapat juga kerugian dengan menggunakan *AJAX*. Tepatnya adalah terdapat beberapa konsekuensi jika menggunakan *AJAX*. Antara lain adalah :

1. **Integrasi Browser**, *browser* memiliki *engine* tersendiri untuk mengolah *history*, sehingga dapat kembali ke halaman sebelumnya dengan mudah, dengan mengeksekusi tombol "*Back*". Dengan menggunakan *AJAX*, halaman yang diciptakan secara dinamis tidak *ter-register* di dalam *engine browser*, sehingga mungkin navigasi (seperti contoh tombol "*Back*") akan tidak berjalan sesuai keinginan. Walaupun demikian, bukannya tidak ada jalan atau trik pemrograman yang dapat mengatasi hal ini.
2. **Waktu Tunggu**, dengan *AJAX*, memang tidak terdapat lagi *refresh* keseluruhan halaman. Tetapi menimbulkan masalah baru, yaitu waktu tunggu yang terjadi karena pertukaran data antara *request* dari pengguna dengan *response server*, menjadi lebih terasa dan terkadang menyesatkan pengguna yang masih awam. Untuk itu, memberikan *progress* selama waktu tunggu, terlebih untuk data yang relatif cukup besar, akan dapat membantu permasalahan tersebut.
3. **Optimasi Search Engine**, sebuah *website* yang statik mudah untuk di-*index* di dalam *search engine*, tidak demikian dengan halaman yang dibuat dengan *AJAX*. Masalah serupa dihadapi oleh semua *website* yang

dinamis, di mana *search engine* tidak dapat mengenali *content website*.

4. **Kompatibilitas Browser**, *javaScript* merupakan bahasa pemrograman utama di dalam *AJAX*. Permasalahannya adalah masing-masing *browser* (juga *browser* yang sama tetapi dengan versi yang berbeda), mungkin mengimplementasikan *JavaScript* berbeda satu dengan yang lainnya. Untuk menghindarinya, *JavaScript* yang Anda gunakan perlu dites dengan berbagai *browser*. Sebaiknya menghindari solusi menuliskan *JavaScript* yang berbeda untuk *browser* yang berbeda, untuk mendukung kompatibilitas *browser*. Cara ini dapat memperumit kode program anda dan menjadi tidak efisien

## 2.5 **AJAX Framework**

*AJAX* terus berkembang seiring dengan kemampuannya yang untuk menciptakan *website* yang interaktif, dan dinamis. Untuk lebih memudahkan proses pemrograman bagi *developer*, maka dibuat *AJAX Framework*. *AJAX Framework* terdiri dari kumpulankumpulan fungsi dan *library* yang siap digunakan. Dengan adanya *framework*, *developer* yang menggunakan *AJAX* terbantu dari dua sisi, yaitu sisi *client* dan sisi *server*. Pada sisi *client*, *framework* menyediakan *function-function JavaScript* untuk mengirimkan *request* ke *server*. Pada sisi *server*, *framework* melakukan proses terhadap *request* yang diterima, mencari data yang diinginkan, dan mengirimkannya kembali pada *web browser*. *AjaxLib* menyediakan *function-function* untuk keperluan sisi

*client*. *AJAX Framework* dapat juga ditemui pada *platform* dan bahasa lainnya, misalnya pada *platform Windows .NET* yang dikenal dengan *ASP.NET AJAX* (sebelumnya dikenal dengan nama *Microsoft Atlas*), yang menyediakan komponen-komponen yang dapat Anda gunakan dengan mudah.

## 2.6 Internet

*Interconnected Network* atau yang lebih populer dengan sebutan *Internet* adalah sebuah sistem komunikasi global yang menghubungkan komputer-komputer dan jaringan-jaringan komputer di seluruh dunia. Setiap komputer dan jaringan terhubung secara langsung maupun tidak langsung ke beberapa jalur utama yang disebut *internet backbone* dan dibedakan satu dengan yang lainnya menggunakan *unique name* yang biasa disebut dengan alamat IP 32 bit. Contoh: 202.155.4.230 . Komputer dan jaringan dengan berbagai *platform* yang mempunyai perbedaan dan ciri khas masing-masing (*Unix, Linux, Windows, Mac, dll*) bertukar informasi dengan sebuah *protokol* standar yang dikenal dengan nama *TCP/IP (Transmission Control Protocol/Internet Protocol)*.

Rangkaian pusat yang membentuk *internet* diawali pada tahun 1969 sebagai *ARPANET*, yang dibangun oleh *ARPA (United States Department of Defense Advanced Research Projects Agency)*. Beberapa penyelidikan awal yang disumbangkan oleh *ARPANET* termasuk kaedah rangkaian tanpa pusat (*decentralised network*), teori *queueing*, dan kaedah pertukaran paket (*packet switching*). Pada 1 Januari 1983, *ARPANET* menukar *protokol* rangkaian pusatnya, dari *NCP* ke *TCP/IP*. Ini

merupakan awal dari *internet* yang kita kenal hari ini. Pada sekitar 1990-an, *internet* telah berkembang dan menyambungkan kebanyakan pengguna jaringan-jaringan komputer yang ada.

Saat ini *internet* dijaga oleh perjanjian *bilateral* atau *multilateral* dan spesifikasi teknis (*protokol* yang menerangkan tentang perpindahan data antara rangkaian). *Protokol-protokol* ini dibentuk berdasarkan perbincangan *Internet Engineering Task Force (IETF)*, yang terbuka kepada umum. Badan ini mengeluarkan dokumen yang dikenali sebagai *RFC (Request for Comments)*. Sebagian dari *RFC* dijadikan *Standar Internet (Internet Standard)*, oleh *Badan Arsitektur Internet (Internet Architecture Board - IAB)*.

*Protokol-protokol internet* yang sering digunakan adalah seperti *IP, TCP, UDP, DNS, PPP, SLIP, ICMP, POP3, IMAP, SMTP, HTTP, HTTPS, SSH, Telnet, FTP, LDAP,* dan *SSL*. Beberapa layanan populer di *internet* yang menggunakan *protokol* di atas, ialah *email* atau surat elektronik, *Usenet, Newsgroup, File Sharing, WWW (World Wide Web), Gopher, Session Access, WAIS, finger, IRC, MUD,* dan *MUSH*. Di antara semua ini, *email* atau surat elektronik dan *World Wide Web* lebih kerap digunakan, dan lebih banyak *servis* yang dibangun berdasarkannya, seperti *milis (Mailing List)* dan *Weblog*. *Internet* memungkinkan adanya *servis* terkini (*Real-time service*), seperti *web radio, dan webcast, yang dapat diakses di seluruh dunia.*

Jumlah pengguna *internet* yang besar dan semakin berkembang, telah mewujudkan budaya *internet*. *Internet* juga mempunyai pengaruh yang besar atas ilmu, dan

pandangan dunia. Dengan hanya berpandukan mesin pencari seperti *Google*, pengguna di seluruh dunia mempunyai akses yang mudah atas bermacam-macam informasi. Dibanding dengan buku dan perpustakaan, *Internet* melambangkan penyebaran (*decentralization*) informasi dan data secara ekstrim.

*Internet* juga semakin banyak digunakan di tempat umum. Beberapa tempat umum yang menyediakan layanan internet termasuk perpustakaan, dan *internet cafe* atau warnet (juga disebut *Cyber Cafe*). Terdapat juga tempat awam yang menyediakan pusat akses *internet*, seperti *Internet Kios*, *Public access Terminal*, dan Telepon web. Dengan tersedianya koneksi *internet* di tempat-tempat tersebut, maka orang awam dapat menikmati layanan-layanan yang disediakan oleh *internet*. Dan hal ini memicu munculnya komunitas dunia maya yang terdiri dari pengguna awam. Komunitas dunia maya bisa berupa *mailing list*, *newsgroup*, *bulletin board*, maupun *portal* komunitas seperti *friendster* dan *facebook*.

## **2.7 Web Server**

*Web server* atau adalah *server* internet yang mampu melayani koneksi transfer data dalam protokol HTTP. *Web server* dirancang untuk melayani bahasa jenis data, mulai dari *text*, *hypertext*, gambar, suara, *plug in* dan sebagainya. *Web server* pada umumnya melayani data dalam bentuk *file* HTML maupun *file* kolaborasi dengan HTML seperti PHP, Javascript dan sebagainya. Salah satu *web server* yang banyak dipakai yaitu Apache *web server*. Apache *web server* merupakan *web server* yang bersifat

*open source*, fleksibel dan mendukung berbagai macam *platform* sistem operasi.

## **2.8 PHP (*Hypertext Preprocessor*)**

Menurut Henky Prihatna (2005), *PHP (Hypertext Preprocessor)* merupakan bahasa berbentuk skrip yang ditempatkan dalam *server* dan diproses di *server*. Hasilnya dikirimkan ke *client*, tempat pemakai menggunakan *browser*.

*PHP* adalah bahasa *server-side scripting* yang menyatu dengan *HTML* untuk membuat aplikasi halaman *web* yang dinamis. *Server-side scripting* maksudnya adalah sintak dan perintah-perintah yang diberikan, sepenuhnya akan dijalankan di *server* dan skrip disertakan pada dokumen *HTML*. Ketika user membuka halaman suatu *web*, *server* akan memproses perintah *PHP* dan mengirimkan hasilnya dalam format *HTML* ke *browser* pengguna tersebut, pengguna tidak dapat melihat kode program yang ditulis ke dalam *PHP* sehingga keamanan dari halaman *web* lebih terjamin.

*PHP* merupakan perangkat lunak yang *open source* dan mampu lintas *platform*. *PHP* menawarkan *konektivitas* yang baik dengan beberapa *database* dan tidak terkecuali semua *database* yang berinterface *ODBC* sehingga *konektivitas database* dengan *web* dapat diakses dengan cepat, mudah, dan stabil.

*Hypertext PreProcessor (PHP)* merupakan salah satu bahasa pemrograman yang banyak dipakai untuk membangun *web* yang dinamis. *PHP* dapat disisipkan diantara bahasa *HTML* dan karena bahasa *server side*, maka bahasa *PHP* akan dieksekusi di *server*, sehingga yang dikirimkan ke

*browser* adalah "hasil jadi" dalam bentuk *HTML*, dan kode *PHP* tidak akan terlihat. Penggunaan bahasa *PHP* ini sangat mendukung untuk pembuatan buku tamu, *polling*, *counter*, forum, dengan menggunakan *DBMS MySQL*. Adapun kelebihan-kelebihan dari *PHP* menurut *Sutarman (2003)* yaitu :

1. Dipakai karena memiliki kecepatan akses yang tinggi.
2. Dapat dijalankan pada beberapa *web server* yang berbeda dan sistem operasi yang berbeda pula. *PHP* dapat berjalan di sistem operasi *UNIX*, *Windows98*, *Windows NT* dan *Macintosh*.
3. Diterbitkan secara gratis.
4. Dijalankan pada *web server Microsoft Personal Web Server*, *Apache*, *IIS*, *Xitami*, dan sebagainya.
5. Disebut sebagai bahasa yang *embedded* (bisa ditempel atau diletakkan dalam tag *HTML*).

Sedangkan menurut *Henky Prihatna (2005)*, kelebihan *PHP* adalah sebagai berikut :

1. *Class platform*, *PHP* dapat dipakai di hampir semua *web server* yang ada di pasaran.
2. *PHP* mempunyai kemampuan untuk melakukan koneksi dengan berbagai macam database yaitu *Interbase*, *mSQL*, *MySQL*, *Unixdbm*, *Oracle*, dan lain sebagainya.
3. *PHP* dapat melakukan semua aplikasi program *CGI* seperti mengambil nilai form, menghasilkan halaman web yang dinamis.
4. Bersifat *open source*. Sebagai produk *open source* biaya operasional *PHP* selalu mengalami kemajuan ke arah produk yang lebih handal setiap saat.

Dalam penulisan perintah-perintah PHP di dalam editor teks dan fungsi-fungsi yang telah ada dalam PHP bersifat *case sensitive* yang artinya penulisan huruf besar (kapital) dan huruf kecil dalam fungsi-fungsi PHP dibedakan. Kode PHP diawali dengan tag `<?>` untuk *short open tag* atau `<?php` dan diakhiri dengan tag `?>`. Adapun struktur untuk menuliskan skrip PHP adalah sebagai berikut :

```
<html>
<head>
<title>Document Title</title>
</head>
<body> <?php
    echo "This is my first Script";
    // One row comment
    /* Multipple
       row comment */?>
</body>
</html>
```

## 2.9 HTML (Hypertext Markup Language)

*HTML* (*hypertext markup language*) adalah suatu bahasa yang digunakan untuk menulis menulis halaman *web*. *Html* dirancang untuk digunakan tanpa tergantung pada suatu *platform* tertentu. Dokumen *html* adalah dokumen teks biasa, dan disebut sebagai *markup language* karena mengandung *tag-tag* tertentu yang digunakan untuk menentukan tampilan suatu teks dan tingkat kepentingan dari teks tersebut dalam suatu dokumen. Format umum HTML adalah :

```
<html>
<head>
<title>Untitled Document</title>
</head>
<body>
</body>
</html>
```

## 2.10 WWW (World Wide Web)

*World Wide Web (WWW)* merupakan salah satu fasilitas yang disediakan di internet. Web merupakan dunia maya di internet yang terdapat ribuan info tentang segala hal dan dikemas dalam bentuk dokumen *hypertext* (Nugroho, 2005). *Hypertext* merupakan konsep dasar dari *world wide web*. Dokumen *hypertext* adalah salah satu dokumen yang memungkinkan untuk menjelajahi dari satu halaman web ke halaman web yang lainnya dengan menggunakan suatu *link* (penghubung).

Web bekerja dalam jaringan komputer yang memanfaatkan teknologi *Hypertext Transfer Protocol (HTTP)*, *Uniform Resource Locator (URL)*, *Protocol Transfer* dan *Domain Name Sistem (DNS)*.

*HTTP (HyperText Transfer Protocol)* adalah suatu protokol yang digunakan oleh *World Wide Web*. *HTTP* mendefinisikan bagaimana suatu pesan bisa diformat dan dikirimkan dari *server* ke *client*. *Uniform Resource Locator (URL)* adalah rangkaian karakter menurut suatu format standar tertentu, yang digunakan untuk menunjukkan alamat suatu sumber seperti dokumen dan gambar di internet (Prihatna, 2005). *DNS (Domain Name Sistem)* adalah sebuah sistem yang menyimpan informasi tentang nama *host* maupun nama *domain* dalam bentuk basis data tersebar (*distributed database*) di dalam jaringan komputer (Nugroho, 2006).

## 2.11 MySQL

*MySQL* adalah sebuah program (*software*) yang membolehkan pengguna berhubungan dengan *server MySQL* dengan menjalankan *query* untuk menampilkan hasilnya

(Prihatna, 2005). Dalam bahasa *SQL* pada umumnya informasi tersimpan dalam tabel-tabel yang secara *logic* merupakan struktur dua dimensi terdiri dari baris (*row* atau *record*) dan kolom (*column* atau *field*). Sedangkan dalam sebuah *database* dapat terdiri dari beberapa tabel. Sebagai basisdata *server* yang memiliki konsep basisdata modern, *MySQL* memiliki banyak keistimewaan, diantaranya yaitu *portability*, *open source*, *multiuser*, *performance tuning*, dan lain sebagainya.

*MySQL* menggunakan perintah dalam bahasa *SQL* antara lain :

1. **SELECT**, Perintah ini digunakan untuk mengambil data dari suatu tabel. Sintak penulisannya adalah: `SELECT{*|namafield}FROM namatabel [WHERE kondisi]`.
2. **INSERT**, Perintah ini digunakan untuk menyisipkan data ke dalam tabel. Sintak penulisannya adalah: `INSERT INTO nama tabel [(field1[,fiels2,...])] VALUE (ekspresi1[,ekspresi2,...])`.
3. **DELETE**, Perintah ini digunakan untuk menghapus sebuah *record* dari suatu tabel. Sintak penulisannya adalah : `DELETE FROM namatabel WHERE kondisi`.
4. **UPDATE**, Perintah ini digunakan untuk memperbaharui nilai suatu data pada tabel. Sintak penulisannya adalah : `UPDATE namatabel SET kriteria WHERE kondisi`.