

BAB 2

LANDASAN TEORI

2.1. Internet

Internet dapat diartikan sebagai jaringan komputer luas dan besar yang mendunia, yaitu menghubungkan pemakai komputer dari suatu negara ke negara lain di seluruh dunia, dimana di dalamnya terdapat berbagai sumber daya informasi dari mulai yang statis hingga yang dinamis dan interaktif.

2.1.1. Sejarah Internet

Sejarah internet dimulai pada 1969 ketika Departemen Pertahanan Amerika, *U.S. Defense Advanced Research Projects Agency* (DARPA) memutuskan untuk mengadakan riset tentang bagaimana caranya menghubungkan sejumlah komputer sehingga membentuk jaringan organik. Tahun 1972, Roy Tomlinson berhasil menyempurnakan program *email* yang ia ciptakan setahun yang lalu untuk ARPANET. Program *email* ini begitu mudah sehingga langsung menjadi populer. Tahun 1973, jaringan komputer ARPANET mulai dikembangkan ke luar Amerika Serikat. Pada 1979, Tom Truscott, Jim Ellis dan Steve Bellovin, menciptakan *newsgroup* pertama yang diberi nama USENET.

Karena komputer yang membentuk jaringan semakin hari semakin banyak, maka dibutuhkan sebuah protokol resmi yang diakui oleh semua jaringan. Pada tahun 1982 dibentuk *Transmission Control Protocol* atau TCP dan *Internet Protokol* atau IP yang kita kenal semua. Untuk menyeragamkan alamat di jaringan komputer yang ada, maka pada tahun 1984 diperkenalkan sistem nama domain, yang kini kita kenal dengan DNS atau *Domain Name System*. Tahun 1988, Jarko

Oikarinen dari Finlandia menemukan dan sekaligus memperkenalkan IRC atau *Internet Relay Chat*. Tahun 1990 adalah tahun yang paling bersejarah, ketika Tim Berners Lee menemukan program editor dan browser yang bisa menjelajah antara satu komputer dengan komputer yang lainnya, yang membentuk jaringan itu. Program inilah yang disebut *www*, atau *World Wide Web*.

2.1.2. Manfaat Internet

Secara umum ada banyak manfaat yang dapat diperoleh apabila seseorang mempunyai akses ke internet. Berikut ini sebagian dari apa yang tersedia di internet:

- a. Informasi untuk kehidupan pribadi: kesehatan, rekreasi, hobi, pengembangan pribadi, rohani, sosial.
- b. Informasi untuk kehidupan profesional/pekerja: sains, teknologi, perdagangan, saham, komoditas, berita bisnis, asosiasi profesi, asosiasi bisnis, berbagai forum komunikasi.

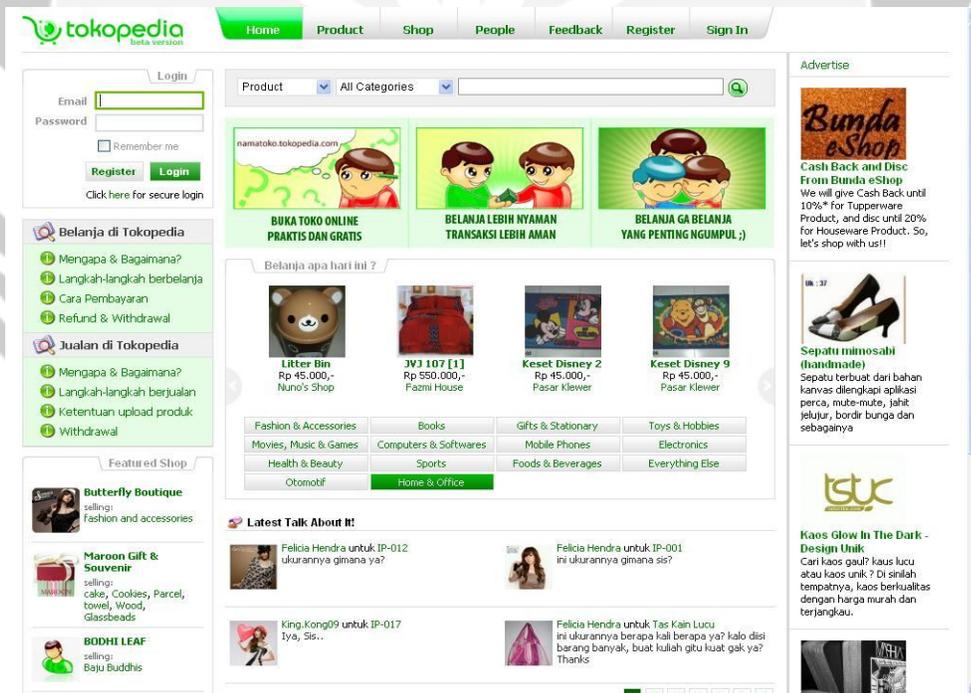
Satu hal yang paling menarik ialah keanggotaan internet tidak mengenal batas negara, ras, kelas ekonomi, ideologi atau faktor faktor lain yang biasanya dapat menghambat pertukaran pikiran. Internet adalah suatu komunitas dunia yang sifatnya sangat demokratis serta memiliki kode etik yang dihormati segenap anggotanya. Manfaat internet terutama diperoleh melalui kerjasama antar pribadi atau kelompok tanpa mengenal batas jarak dan waktu.

(Sumber: Pengantar World Wide Web, Edy Purwanto dan Tim Sub Bag jaringan informasi IPTEK, www.litbang.depkes.go.id/tik/media/Pengantar_WWW.doc, 30 September 2009, 14:50 WIB)

2.2. E-marketplace

E-marketplace adalah sebuah pasar virtual dimana di dalamnya berkumpul para penjual dan pembeli yang menjalankan bisnisnya secara online. Layanan yang ditawarkan oleh *E-marketplace* antara lain katalog elektronik yang menampilkan barang dan jasa, daftar direktori bisnis, dan lelang secara online.

Salah satu contoh e-marketplace adalah website tokopedia.com, website ini menyediakan dua jenis anggota di dalamnya, yaitu penjual dan pembeli. Transaksi pembayaran menggunakan transfer antar bank, dimana pihak tokopedia.com bertindak sebagai penengahnya, sehingga transaksi jauh lebih aman. Di bawah ini adalah gambar halaman website tokopedia.com:



Gambar 2.1. Halaman website www.tokopedia.com (www.tokopedia.com, 14 Oktober 2009, 20:58 WIB)

2.2.1. Keuntungan menggunakan E-marketplace

Keuntungan yang didapat jika bergabung dengan *E-marketplace* sangatlah bervariasi, baik untuk kalangan industri maupun pebisnis, atau untuk penjual dan pembeli. Di bawah ini adalah keuntungan-keuntungan yang kita dapatkan jika menggunakan *e-marketplace*:

1. Keuntungan secara umum
 - a. Lebih besarnya kesempatan bagi penjual dan pembeli untuk membentuk partnership baru.
 - b. Lebih transparan dalam proses penjualan, baik dalam ketersediaan barang maupun harga barang.
 - c. Dapat dihilangkannya masalah perbedaan jam kerja dalam perdagangan internasional.
2. Keuntungan bagi pembeli
 - a. Pembeli mendapatkan informasi harga dan ketersediaan barang yang terbaru sehingga memudahkan dalam melakukan transaksi pembelian.
 - b. Adanya kemudahan dalam membandingkan harga dan barang yang ditawarkan, dimana hal ini menghemat waktu dan biaya pencarian barang.
 - c. Pembeli merasa aman dalam melakukan transaksi karena melakukan transaksi dengan penjual yang juga anggota dari *e-marketplace*.
3. Keuntungan bagi penjual
 - a. Menambah media penyaluran penjualan barang ke pasar.
 - b. Mengurangi biaya pemasaran secara signifikan.

(Sumber: www.businesslink.gov.uk, 14 Oktober 2009, 20:44 WIB)

2.3. Website

World Wide Web (WWW) atau biasa disebut dengan Web, merupakan salah satu sumber daya Internet yang berkembang pesat (Kadir, 2003). Informasi Web didistribusikan melalui pendekatan *hypertext*, yang memungkinkan suatu teks pendek menjadi acuan untuk membuka dokumen yang lain. Dengan pendekatan *hypertext* ini, seseorang dapat memperoleh informasi dengan meloncat dari suatu dokumen ke dokumen yang lain. Dokumen-dokumen yang diakses pun dapat tersebar di pelbagai mesin dan bahkan di berbagai negara.

Bagai jaring laba-laba, *website* telah membentang ke seluruh penjuru dunia. Tidak hanya terbatas pada lembaga-lembaga penelitian yang ingin mempublikasikan hasil riset, *website* juga banyak digunakan oleh perusahaan bisnis yang ingin mengiklankan produk atau melakukan transaksi bisnisnya.

2.3.1. Sejarah Website

Sejarah Web dimulai pada bulan Maret 1989 ketika Tim Berners-Lee yang bekerja di Laboratorium Fisika Partikel Eropa atau yang dikenal dengan nama CERN (*Conseil Européen pour la Recherche Nucléaire*) yang berada di Genewa, Swiss, mengajukan protokol (suatu tatacara untuk berkomunikasi) sistem distribusi informasi Internet yang digunakan untuk berbagai informasi di antara para fisikawan.

Protokol inilah yang selanjutnya dikenal sebagai protokol *World Wide Web* dan dikembangkan oleh *World Wide Web Consortium* (W3C). Sebagaimana diketahui, W3C adalah konsorsium dari sejumlah organisasi yang berkepentingan dalam pengembangan berbagai standar yang berkaitan dengan *Web*.

2.3.2. Aplikasi Website

Pada awalnya *website* dibangun hanya dengan menggunakan bahasa yang disebut dengan HTML (*HyperText Markup Language*) dan protokol yang digunakan dinamakan HTTP (*HyperText Transfer Protocol*). Pada perkembangan berikutnya, sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan HTML. Pada saat ini, banyak skrip seperti itu : antara lain yaitu PHP dan ASP, sedangkan contoh yang berupa objek antara lain adalah applet (Java).

Aplikasi *Web* sendiri dapat dibagi menjadi :

- a. *Web* Statis
- b. *Web* Dinamis

Web statis dibentuk dengan menggunakan HTML saja. Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus-menerus untuk mengikuti setiap perubahan yang terjadi. Kelemahan ini diatasi dengan model aplikasi *Web* dinamis.

Dengan memperluas kemampuan HTML, yakni dengan menggunakan perangkat lunak tambahan, perubahan informasi dalam halaman-halaman *Web* dapat ditangani melalui perubahan data, bukan melalui perubahan program. Sebagai implementasinya, aplikasi *Web* dapat dikoneksikan ke basisi data. Dengan demikian perubahan informasi dapat dilakukan oleh operator atau yang bertanggung jawab dengan kemutakhiran data, dan tidak menjadi tanggung jawab pemrogram.

Dengan menggunakan pendekatan *Web* dinamis dimungkinkan untuk membentuk aplikasi berbasis *Web* (*Web-based*

application). Sebagai contoh, sistem informasi akademis berbasis *Web* memungkinkan seorang mahasiswa melihat informasi nilai dari matakuliah-matakuliah yang sudah diambilnya dari mana saja. Selain itu, pada masa semester baru, mahasiswa dapat memasukkan data KRS (Kartu Rencana Studi) melalui internet.

2.3.3. Teknologi Website

Dari sisi teknologi yang digunakan untuk membentuk *Web* dinamis, terdapat dua macam pengelompokan, yaitu:

- a. Teknologi pada sisi *client* (*client-side technology*).
- b. Teknologi pada sisi *server* (*server-side technology*).

2.3.3.1. Teknologi Web pada sisi Client

Teknologi *Web* pada sisi *client* diimplementasikan dengan mengirimkan kode perluasan HTML atau program tersendiri dan HTML ke *client* (Kadir, 2003). *Client*-lah yang bertanggung jawab dalam melakukan proses terhadap seluruh kode yang diterima.

Kelemahan pendekatan seperti ini adalah terdapat kemungkinan bahwa browser pada *client* tidak mendukung fitur kode perluasan HTML. Sebagai contoh, kode VBScript yang dilekatkan pada kode HTML tidak akan berfungsi sekiranya browser yang digunakan *client* tidak mendukungnya. Kelebihan teknologi pada sisi *client* adalah memungkinkan penampilan *content* yang bersifat dinamis, misalnya menampilkan jam yang terus menerus berubah ataupun untuk membuat animasi gambar yang mengikuti gerakan penunjuk *mouse*.

Yang termasuk dalam teknologi pada sisi *client*:

- a. Kontrol ActiveX

- b. Java Applet
- c. Skrip-skrip *client* (misalnya JavaScript)

2.3.3.2. Teknologi Web pada sisi Server

Teknologi Web pada sisi *Server* memungkinkan pemrosesan kode di dalam server sehingga kode yang sampai pada pemakai berbeda dengan kode asli pada *server* (Kadir, 2003).

Keuntungan penggunaan teknologi pada sisi *server* adalah:

- a. Mengurangi lalu lintas jaringan dengan cara menghindari percakapan bolak balik antara *client* dan *server*.
- b. Mengurangi waktu pemuatan kode, mengingat *client* hanya mengambil kode HTML saja.
- c. Mencegah masalah ketidak kompatibelan *browser*.
- d. *Client* dapat berinteraksi dengan data yang ada pada *server*.
- e. Mencegah *client* mengetahui rahasia kode (mengingat kode yang di berikan ke *client* berbeda dengan kode asli pada *server*).

Beberapa contoh teknologi yang berjalan di *server* adalah:

- a. Common Gateway Interface (CGI)
- b. Active Server Page (ASP)
- c. Java Servlets dan JavaServer Page (JSP)
- d. PHP

2.4. PHP

PHP singkatan dari *PHP Hypertext Preprocessor* yang digunakan sebagai bahasa *script* pada sisi *server* dalam pengembangan *website* yang disisipkan pada dokumen HTML.

Penggunaan PHP memungkinkan sebuah *website* dapat dibuat dinamis sehingga pemeliharaan *website* tersebut menjadi lebih mudah dan efisien.

PHP merupakan perangkat lunak *open source* yang disebar dan dilisensikan secara gratis serta dapat di-*download* secara bebas dari *website* resminya di <http://www.php.net>. Sebagai tambahan, PHP ditulis menggunakan bahasa pemrograman tingkat rendah, yaitu bahasa C.

2.4.1. Sejarah PHP

PHP diciptakan pertama kali oleh Rasmus Lerdorf pada tahun 1994. Awalnya, PHP digunakan untuk mencatat jumlah serta untuk mengetahui siapa saja pengunjung pada *homepage*-nya. Rasmus Lerdorf adalah salah seorang pendukung *open source*, oleh karena itu, ia mengeluarkan Personal Home Page Tools versi 1.0 secara gratis, kemudian menambahkan kemampuan PHP 1.0 dan meluncurkan PHP 2.0.

Pada tahun 1996, PHP telah banyak digunakan dalam *website* di dunia. Sebuah kelompok pengembang perangkat lunak yang terdiri dari Rasmus, Zeew Suraski, Andi Gutman, Stig Bakken, Shanne Caraveo, dan jim Winstead bekerja sama untuk menyempurnakan PHP 2.0. Akhirnya, pada tahun 1998, PHP 3.0 diluncurkan. Penyempurnaan terus dilakukan sehingga pada tahun 2000 dikeluarkan PHP 4.0.

2.4.2. Kelebihan-kelebihan PHP

PHP memiliki banyak kelebihan yang tidak dimiliki oleh bahasa *script* sejenis. PHP difokuskan pada pembuatan *script server-side*, yang bisa melakukan apa saja yang dapat dilakukan oleh CGI, seperti mengumpulkan data dari *form*,

menghasilkan isi halaman *website* menjadi dinamis, dan kemampuan mengirim serta menerima *cookies*, bahkan lebih daripada kemampuan CGI (Peranginangin 2006).

PHP dapat digunakan pada semua sistem operasi, antara lain Linux, Unix (termasuk variannya HP-UX, Solaris, dan OpenBSD), Microsoft Windows, MAC OS X, RISC OS. PHP juga mendukung banyak Web Server, seperti Apache, Microsoft Internet Information Server (MIIS), Personal Web Server (PWS), Nestcape iPlanet servers, O'Reilly Website Pro server, audium, Xitami, OmniHTTPd, dan masih banyak lagi lainnya, bahkan PHP dapat bekerja sebagai CGI *processor*. PHP tidak terbatas pada hasil keluaran HTML (*HyperText Markup Language*). PHP juga memiliki kemampuan untuk mengolah keluaran gambar, *file* PDF, dan movie Flash. PHP dapat menghasilkan teks seperti XHTML dan *file* XML lainnya.

Salah satu fitur yang dapat diandalkan oleh PHP adalah dukungannya terhadap banyak *database*. Berikut *database* yang dapat didukung oleh PHP, antara lain Adabas D, *dbase*, Direct MS-SQL, Empress, Frontbase, Informix, MySQL ODBC, Oracle, Interbase, PostgreSQL, dan lain-lain.

2.5. Smarty

Smarty adalah mesin *template* untuk PHP. Lebih khusus, Smarty memfasilitasi cara yang bisa diatur untuk memisahkan logika aplikasi dan *content* dari tampilannya. Ini jauh lebih baik dijelaskan dalam situasi di mana pemrogram aplikasi dan desainer *template* memainkan aturan yang berbeda, atau secara umum bukan orang yang sama. Sebagai contoh, katakanlah kita sedang membuat halaman *website* yang menampilkan artikel koran.

- a. Artikel `$headline`, `$tagline`, `$author` dan `$body` adalah elemen konten, tidak berisi informasi mengenai bagaimana akan ditampilkan. Ia akan dioper ke dalam Smarty oleh aplikasi.
- b. Kemudian desainer mengubah *template* dan menggunakan kombinasi tag HTML dan tag *template* untuk membentuk presentasi terhadap variabel ini dengan elemen seperti tabel, div, warna latar belakang, ukuran font, stylesheets, svg, dan lain-lain.
- c. Suatu hari pemrogram perlu mengubah cara konten artikel diambil (perubahan dalam logika aplikasi). Perubahan ini tidak mempengaruhi desainer *template*, konten masih akan muncul dalam *template* persis sama.
- d. Demikian juga jika desainer *template* ingin mendesain ulang *template* seutuhnya, ini tidak memerlukan perubahan logika aplikasi.
- e. Oleh karena itu, pemrogram dapat membuat perubahan terhadap logika aplikasi tanpa perlu merestrukturisasi *template*, dan desainer *template* bisa membuat perubahan terhadap *template* tanpa membongkar logika aplikasi.

Satu tujuan desain Smarty adalah pemisahan logika bisnis (*business logic*) dan logika presentasi (*presentation logic*).

- a. Ini berarti *template* tentu saja dapat berisi logika di bawah kondisi yang hanya untuk presentasi saja. Hal seperti menyertakan *template* lain, memilih warna baris tabel, membesarkan huruf variabel, mengulang terus sebuah data array dan menampilkannya adalah contoh dari logika presentasi.

b. Ini tidak berarti bahwa Smarty memaksa pemisahan logika bisnis dan presentasi. Smarty tidak mengetahui yang mana adalah yang mana, maka menempatkan logika bisnis dalam template adalah kita sendiri yang melakukannya.

c. Juga, jika kita menginginkan tidak ada logika dalam *template*, kita tentunya dapat melakukannya dengan menetapkan konten cukup ke teks dan variabel saja. Salah satu aspek unik mengenai Smarty adalah kompilasi *template*. Ini berarti Smarty membaca *template file* dan membuat naskah PHP darinya.

Sekali dibuat, selanjutnya ia dieksekusi darinya. Oleh karenanya tidak ada beban menguraikan *template file* untuk setiap permintaan, dan setiap *template* dapat memanfaatkan solusi *cache compiler* PHP seperti eAccelerator, ionCube, mmCache atau Zend.

2.5.1. Penjelasan Singkat mengenai Smarty

2.5.1.1. Instalasi

Smarty membutuhkan *web server* yang menjalankan PHP 4.0.6 atau lebih tinggi. Instalasi *file* pustaka Smarty yang ada dalam sub direktori */libs/* dari distribusi. Ini adalah *file* berekstensi *.php* yang tidak boleh diubah. Ia berbagi diantara seluruh aplikasi dan hanya diubah ketika kita meningkatkannya ke versi Smarty baru.

Smarty menggunakan konstan PHP bernama `SMARTY_DIR` yang merupakan *path file* sistem lengkap ke direktori *libs/Smarty*. Pada dasarnya, jika aplikasi kita dapat menemukan file `Smarty.class.php`, kita tidak perlu membuat `SMARTY_DIR` karena Smarty akan mengetahui dirinya sendiri. Oleh karena itu, jika `Smarty.class.php` tidak dalam `include_path` kita,

atau kita tidak menyertakan *path* absolut kepadanya dalam aplikasi, maka kita harus mendefinisikan `SMARTY_DIR` secara manual. `SMARTY_DIR` harus menyertakan akhiran garis miring(`/`). Setelah itu, kita menyiapkan direktori Smarty untuk aplikasi yang akan kita buat, dimana kita memerlukan empat direktori standar yang bernama `templates`, `templates_c`, `configs`, dan `cache`. Keempat direktori tersebut didefinisikan dalam properti kelas Smarty yang masing-masing adalah `$template_dir`, `$compile_dir`, `$config_dir`, dan `$cache_dir`. Di bawah ini adalah contoh struktur direktori dan *file* aplikasi yang menggunakan Smarty:

```

/usr/local/lib/Smarty-v.e.r/libs/
    Smarty.class.php
    Smarty_Compiler.class.php
    Config_File.class.php
    debug.tpl
    internals/*.php
    plugins/*.php

/web/www.example.com/
    guestbook/
        templates/
            index.tpl
        templates_c/
        configs/
        cache/
        htdocs/
            index.php

```

Gambar 2.2. Struktur file dan direktori(Manual Smarty-2.6.14-docs.pdf)

2.5.1.2. Sintaks Dasar

a. Komentar

Komentar template dikelilingi oleh bintang, dan ditutup oleh tag pembatas seperti:

```
{* ini adalah sebuah komentar *}
```

Komentar Smarty tidak ditampilkan dalam output *template* akhir, tidak seperti `<!-- HTML comments -->`. Ini berguna untuk membuat catatan internal dalam *template* yang tak seorangpun akan melihatnya.

b. Variabel

Variabel *template* dimulai dengan tanda dollar(\$). Variabel ini dapat berupa angka, huruf dan garis bawah, lebih mirip variabel PHP. Kita bisa mereferensi array dengan indeks secara numerik atau non-numerik. Juga referensi properti dan metode obyek.

Variabel *file config* adalah perkecualian untuk sintaks \$ dan sebaliknya direferensi dengan sekeliling #tanda kris#, atau melalui variabel `$smarty.config`. Di bawah ini adalah contoh kode variabel:

```
{ $foo }      --> menampilkan nilai dalam variabel
{ #foo# }    --> menampilkan nilai dalam file config foo
{ $smarty.config.foo } --> sama seperti { #foo# }
```

c. Fungsi

Setiap tag Smarty bisa mencetak variabel ataupun memanggil beberapa fungsi. Ini diproses dan ditampilkan dengan menutupi fungsi dan atributnya di dalam pembatas seperti: `{funcname attr1='val1' attr2='val2'}`.

```

{config_load file='colors.conf'}

{include file='header.tpl'}
{insert file='banner_ads.tpl' title='Smarty is cool'}

{if $logged_in}
    Welcome, <font color="{#fontColor#}">{$name}!</font>
{else}
    hi, {$name}
{/if}

{include file='footer.tpl' ad=$random_id}

```

Gambar 2.3. Penggunaan fungsi dalam Smarty (Manual
Smarty-2.6.14-docs.pdf)

2.5.2. Kelebihan Smarty

Karena menggunakan cache dalam implementasinya, maka Smarty sangat cepat dalam menampilkan halaman *website*. Namun, selain itu, ada beberapa kelebihan Smarty, yaitu:

- a. Efisien karena pengurai PHP yang mengerjakan pekerjaan beratnya.
- b. Tidak ada kelebihan penguraian *template*, hanya sekali mengompilasi.
- c. Pintar mengenai rekompilasi hanya *file template* yang telah diubah.
- d. Kita dapat membuat dengan mudah fungsi kustom dan pengubah variabel, agar bahasa *template* bisa diperluas secara ekstrim.
- e. Template bisa mengkonfigurasi sintaks tag {pemisah}, agar kita dapat menggunakan {\$foo}, {{ \$foo }}, <!--{\$foo}-->, dan lain lain.
- f. Konstruksi {if}..{elseif}..{else}..{/if} dioper ke pengurai PHP, maka sintaks ekspresi {if...} bisa berupa evaluasi sesederhana atau serumit yang kita inginkan.

- g. Membolehkan pengulangan tidak terbatas dari sections, if's dan lain lain.
 - h. Dimungkinkan untuk menyertakan kode PHP langsung dalam file *template* kita, meskipun ini mungkin tidak diperlukan (ataupun direkomendasikan) karena mesin dapat dikustomisasi.
 - i. Adanya dukungan *built-in caching*.
 - j. Adanya fungsi tambahan dalam penanganan *cache*.
- (Sumber: Manual Smarty-2.6.14-docs.pdf, 15 Oktober 2008, 22:45 WIB)

2.6. AJAX

Ajax sebenarnya akronim dari *Asynchronous JavaScript and XML* dimana merupakan sebuah teknik komunikasi (dalam aplikasi *website*) antara *client* dan *server* secara Asinkron(tanpa harus memperbarui halaman *website*).

2.6.1. Sejarah AJAX

Ajax sebenarnya bukanlah sesuatu yang baru, karena teknologi yang dipakai Ajax, yaitu XMLHttpRequest (XHR) telah digunakan di *browser* Internet Explorer 5 yang digunakan sebagai kontrol terhadap Active X. Teknik ini terlihat baru karena keluar dari paradigma mengenai *request/response* dalam dunia web(Asleson, Ryan, & Schutta, Nathaniel T. 2006. *Foundations of Ajax*. USA: Apress.).

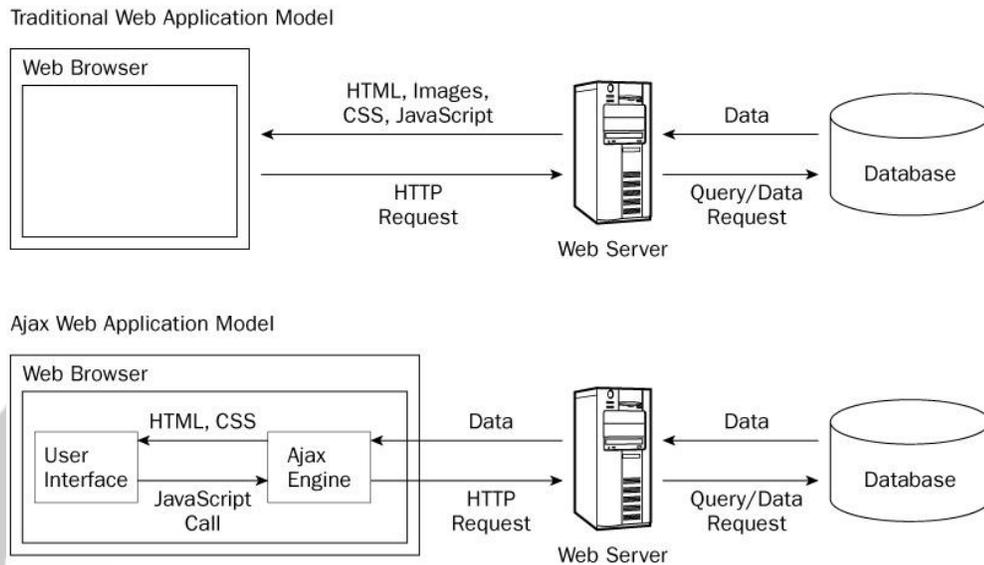
Istilah Ajax sendiri muncul saat Jesse James Garrett dari Adaptive Path, pada bulan Februari 2005 membuat esai yang berjudul "Ajax: A New Approach to Web Applications", dimana Ajax merupakan akronim dari *Asynchronous JavaScript and XML*. Teknik ini memungkinkan browser dapat berkomunikasi dengan *server* tanpa harus memperbarui halaman. *Website*

yang pertama kali mengadopsi teknik ini adalah Google (digunakan dalam Google Maps, Google Suggest, Gmail).

2.6.2. Konsep AJAX

Agar dapat menggunakan teknik Ajax, maka terlebih dahulu, kita membuat *instance* dari objek XMLHttpRequest. Objek XMLHttpRequest bukanlah standar dari W3C, oleh karena itu terdapat perbedaan implementasinya antara browser IE dengan browser yang berbasis Gecko (Safari, Firefox, dan Opera). Di browser IE, XMLHttpRequest dianggap sebagai objek ActiveX, sedangkan di browser berbasis Gecko, XMLHttpRequest dianggap sebagai objek dari JavaScript. Namun, permasalahan bukanlah masalah yang serius, karena dapat diatasi dengan kode JavaScript yang melakukan pengecekan browser.

Melalui instance XMLHttpRequest ini, dilakukan request ke server dengan menggunakan *method* open() dan send(). Setelah itu, dengan menggunakan fungsi untuk menerima hasil dari server (fungsi *callback*), maka apa yang diberikan oleh server, ditampilkan ke halaman *website*. Di bawah ini adalah gambar perbandingan antara penggunaan teknik Request/Response (tradisional) dan teknik Ajax :



Gambar 2.4. Model aplikasi web tradisional dan AJAX(Zakas, Nicholas C., McPeak, Jeremy, & Fawcett, Joe. 2006. *Professional Ajax*. USA: Wiley Publishing, Inc.)

2.7. jQuery

Sekarang ini, banyak *website* yang dibangun dengan lingkungan dinamis, dimana tampilan yang menarik dan interaktif sangat ditonjolkan. Oleh karena itu, para developer menggunakan pustaka-pustaka JavaScript seperti jQuery untuk melakukan hal tersebut, salah satu alasan mengapa menggunakan jQuery karena kemampuan dalam melakukan pekerjaan yg sangat luas.

2.7.1. Sejarah jQuery

Pada awalnya, John Resig (pembuat jQuery) melakukan penyempurnaan pada pustaka Prototype pada bulan Agustus 2005. kemudian pada tanggal 14 Januari 2006, jQuery secara resmi dirilis. Bulan Agustus 2006, dirilis versi stabilnya, yaitu jQuery 1.0, dimana fitur-fiturnya antara

lain mendukung CSS selector, *event-handling*, dan interaksi dengan AJAX.

Pada januari 2007, dirilis jQuery 1.1, dimana method-method yang jarang digunakan pada akhirnya dikombinasikan, selain itu, dilakukan pengurangan beberapa *method*. Bulan Juli 2007, dirilis jQuery 1.1.3, dimana dilakukan perbaikan kecepatan pada *engine* jQuery. Pada versi ini, jQuery menjadi salah satu library favorit layaknya Prototype, Mootools, dan Dojo. Pada bulan September 2007, dirilis jQuery 1.2, pada versi ini, XPath dihapuskan dari pustaka, selain itu, dilakukan penambahan *namespaced events* sehingga pembuatan *plugin* untuk jQuery menjadi lebih mudah.

Bulan Mei 2008, dirilis jQuery 1.2.6, pada rilis ini, *plugin* Dimensions milik Brandon Aaron dimasukkan kedalam pustaka utama. Akhirnya, pada bulan Januari 2009, dirilislah jQuery 1.3, dimana dilakukan perbaikan pada *selector engine* (*Sizzle*) sehingga meningkatkan performa jQuery secara signifikan (Chaffer, 2009).

2.7.2. Kelebihan jQuery

Saat ini, jQuery merupakan salah satu pustaka JavaScript yang populer, hal ini dikarenakan banyaknya pengembang *website* yang menggunakan jQuery. Mereka menggunakan jQuery karena banyaknya kelebihan yang ditawarkan oleh jQuery, antara lain:

- a. mengakses elemen di dalam dokumen HTML dan dapat merespon interaksi user.
- b. fungsi -fungsi animasi yang dapat disisipkan ke dalam sebuah dokumen.

- c. Menyederhanakan pembuatan kode yang menggunakan teknik AJAX.
- d. menyederhanakan pekerjaan JavaScript yang kompleks.

