

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1. MM TexCraft**

MM Texcraft adalah sebuah UKM (usaha kecil menengah) yang usahanya bergerak dalam bidang kerajinan dengan menggunakan bahan dasar kain yaitu memproduksi bermacam-macam kerajinan tangan (penunjang dekorasi rumah, tas, souvenir) dan produk fashion (Baju, dll) berbahan dasar kain traditional dari berbagai daerah di Indonesia terutama tenun dan blaco. Usaha ini berdiri sejak tahun 2004.

UKM adalah sebuah istilah yang mengacu ke jenis usaha kecil yang memiliki kekayaan bersih paling banyak Rp 200.000.000 tidak termasuk tanah dan bangunan tempat usaha. Dan usaha yang berdiri sendiri. Menurut Keputusan Presiden RI no. 99 tahun 1998 pengertian Usaha Kecil adalah: "Kegiatan ekonomi rakyat yang berskala kecil dengan bidang usaha yang secara mayoritas merupakan kegiatan usaha kecil dan perlu dilindungi untuk mencegah dari persaingan usaha yang tidak sehat." (<http://id.wikipedia.org>, 2009)

#### **2.2. Kustomisasi**

Kustomisasi diambil dari kata *customize* yang artinya membuat atau memodifikasi berdasarkan spesifikasi individual atau personal. Dalam konteks pembuatan *software* ini kustomisasi adalah suatu fungsi dimana pengguna dapat memodifikasi suatu objek yang ada sesuai dengan spesifikasi yang diinginkan seperti

merubah ukuran, warna, dan bahan baku dari produk tertentu.

### 2.3. Internet

Secara harafiah, *internet* (kependekan dari '*inter-network*') ialah rangkaian komputer yang berhubung menerusi beberapa rangkaian. Sedangkan *Internet* (huruf 'I' besar) ialah sistem komputer umum, yang berhubung secara global dan menggunakan *TCP/IP* sebagai protokol pertukaran paket (*packet switching communication protocol*). Rangkaian *internet* yang terbesar dinamakan *Internet*. Cara menghubungkan rangkaian dengan kaedah ini dinamakan *internetworking*.

*Internet* dijaga oleh perjanjian bi- atau multilateral dan spesifikasi teknikal (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/surat\_elektronik, Usenet, Newsgroup, perkongsian file (File Sharing), WWW (World Wide Web), Gopher, akses sesi (Session Access), WAIS, finger, IRC, MUD,*

dan *MUSH*. Di antara semua ini, *email/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. (FerdI Irdianto,2007)

#### **2.4. Web Programming Scripting**

Dalam dunia web, terdapat dua kategori bahasa yang digunakan untuk membangunnya, *Client Side Scripting Language* dan *Server Side Scripting Language*. Perbedaan dari keduanya, terletak dari bagaimana bahasa tersebut diterjemahkan menjadi *HTML*. (FerdI Irdianto, 2007)

##### **2.4.1. Server Side**

Seperti namanya, *Server Side* melakukan kompilasi pada sisi server. Setelah melalui proses kompilasi, bahasa tersebut kemudian dikirim kembali kepada client berupa bahasa *HTML*. Itulah sebabnya, kode bahasa *Server Side* tidak akan terlihat pada *browser* anda. Contoh bahasa *Server Side* antara lain adalah *PHP*, *ASP*, *JSP*, *Cold Fusion*. (FerdI Irdianto,2007)

##### **2.4.2. Client Side**

*Client Side* melakukan proses kompilasinya pada sisi *Client*, tepatnya oleh *browser* yang dipakai oleh pengunjung. Itulah sebabnya tiap *browser* berbeda dalam menterjemahkan bahasa *Client Side*. Contoh bahasa

*Client Side* antara lain adalah Javascript dan VBScript.  
(Ferdinand Irdianto, 2007)

## 2.5. Online Shopping

*Online Shopping* adalah proses dimana pelanggan melakukan pembelian produk atau jasa melalui *internet*. Sebuah toko *online*, *e-shop*, *e-store*, *internet shop*, *webstore*, *online store* atau *virtual store* menyebabkan terbentuknya analogi fisik dari cara pembelian produk dan jasa dari pembelian produk dan jasa pada penjualan secara fisik seperti penjualan di mall-mall.

*Online shopping* adalah tipe dari *electronic commerce (e-commerce)* yang digunakan untuk transaksi *business-to business (B2B)* dan *business-to consumer (B2C)*.

Istilah-istilah "*webshop*" juga berarti suatu tempat dimana bisnis dilakukan *web development*, *web hosting* dan yang berhubungan dengan aktivitas kegiatan yang terjadi (*web* mengacu pada *world wide web* and "*shop*" memiliki arti atau istilah yang menggambarkan tempat dimana suatu pekerjaan dilakukan)

Pembelian secara *online* berarti bahwa ada cara-cara baru untuk menekan biaya dengan mengurangi jumlah pegawai yang diperlukan. Hal ini merupakan cara yang lebih efektif menyampaikan suatu barang ke orang-orang dan menyebar pada letak geografis yang berbeda

Kelebihan dan kekurangan *online shopping* :

1. *Bargaining power of consumer*, yaitu pembeli dapat menikmati pilihan yang lebih luas dalam berbelanja.

2. *Supplier Power* yaitu lebih sulit bagi pembeli untuk mengatur saluran atau koneksi non-*digital* sehingga *internet* dapat menambah nilai komersial suatu barang.
3. Ancaman yang dating dari para pesaing baru. *Online* artinya lebih mudah bagi penjual untuk mempromosikan layanan baru dengan biaya lebih rendah.
4. Ancaman dari barang pengganti
5. Persaingan diantara para kompetitor, lebih mudah untuk memperkenalkan layanan baru pada pasar yang berbeda. (<http://id.wikipedia.org> , 2009)

#### 2.6. CSS

Cascading Style Sheets (CSS) adalah suatu *bahasa stylesheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam *bahasa markup*. Penggunaan yang paling umum dari *CSS* adalah untuk memformat halaman web yang ditulis dengan *HTML* dan *XHTML*. Walaupun demikian, bahasanya sendiri dapat dipergunakan untuk semua jenis dokumen *XML* termasuk *SVG* dan *XUL*. Spesifikasi *CSS* diatur oleh *World Wide Web Consortium (W3C)*.

*CSS* digunakan oleh penulis maupun pembaca halaman web untuk menentukan warna, jenis huruf, tata letak, dan berbagai aspek tampilan dokumen. *CSS* digunakan terutama untuk memisahkan antara isi dokumen (yang ditulis dengan *HTML* atau *bahasa markup* lainnya) dengan presentasi dokumen (yang ditulis dengan *CSS*). Pemisahan ini dapat meningkatkan aksesibilitas isi, memberikan

lebih banyak keleluasaan dan kontrol terhadap tampilan, dan mengurangi kompleksitas serta pengulangan pada struktur isi.

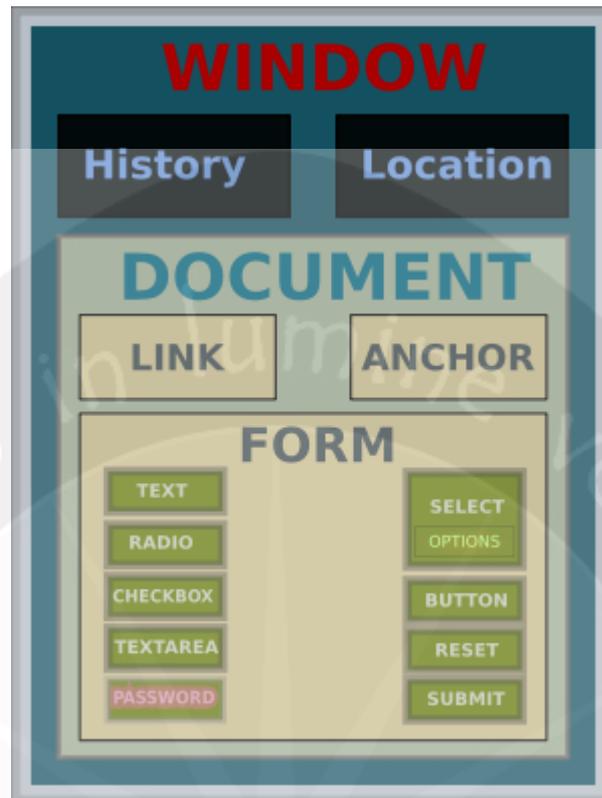
CSS memungkinkan halaman yang sama untuk ditampilkan dengan cara yang berbeda untuk metode presentasi yang berbeda, seperti melalui layar, cetak, suara (sewaktu dibacakan oleh *browser basis-suara* atau *pembaca layar*), dan juga alat pembaca *braille*. Halaman *HTML* atau *XML* yang sama juga dapat ditampilkan secara berbeda, baik dari segi gaya tampilan atau skema warna dengan menggunakan *CSS*. (<http://id.wikipedia.org>, 2009)

## 2.7. DOM

*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. (<http://www.sony-ak.com/index.php>).

Contoh lihat dibawah ini :

```
<BODY>
  <p>This is a <b>sample</b> paragraph</p>
  <p id='dua'>Hello world</p>
  <SCRIPT language="javascript">
    elemen = document.documentElement.lastChild;
    document.write(elemen.tagName + "<br>");
    //BODY
    elemenbaru = document.createElement("h1");
    elemen.appendChild(elemenbaru); //<h1></h1>
    teksbaru = document.createTextNode("Hello");
    elemenbaru.appendChild(teksbaru);
    //<h1>Hello</h1>
    par = document.getElementById("dua");
    elemen.removeChild(par);
    //par.outerHTML = "";
  </SCRIPT>
</BODY>
```



Gambar 2.1 Hirarki Objek-Document Object Model

## 2.8. Microsoft.NET

*Microsoft .NET* adalah *software* yang menghubungkan informasi, orang, sistem, dan *device/perangkat* yang menjangkau *client*, *server* dan tool pengembang.

*Microsoft .NET* disusun oleh :

- *Dot NET framework*, digunakan untuk membangun dan menjalankan semua macam *software* termasuk aplikasi berbasis web, aplikasi *smart client*, dan layanan web *XML (Extensible Markup Language)*. Menyediakan komponen untuk berbagi data melalui *Network* menggunakan *platform* protokol independen seperti *XML, SOAP, dan HTTP*.
- Tool-tool pengembang seperti *Microsoft Visual Studio.NET* yang menyediakan *IDE (Integrated*

*Development Environment*) untuk memaksimalkan produktivitas pengembangan menggunakan *.NET Framework*.

- Serangkaian server termasuk *Microsoft Window server 2003, Microsoft SQL Server* dan *Microsoft BizTalk Server* yang terintegrasi, untuk menjalankan, mengoperasikan dan mengelola services Web dan aplikasi berbasis Web.
- *Software client* seperti *Windows XP, Windows CE* dan *Microsoft Office XP* yang membantu pengembangan untuk menyebarkan dan mengelola aplikasinya.

(Ariyo Suryo Kusumo, Drs, *Visual Basic .NET, Elex Media Komputindo, Jakarta, 2004*)

### 2.8.1. **Dot NET Framework**

*Dot NET Framework* adalah lingkungan untuk membangun, deploying/menyebarkan, dan menjalankan services Web dan aplikasi lainnya.

*Dot NET Framework* disusun oleh dua komponen utama, yaitu *Common Language Runtime* (runtime bahasa umum) dan *.NET framework class* (pustaka kelas .NET Framework).

Secara sederhana *.NET Framework* adalah platform tunggal dimana semua orang dapat mengembangkan aplikasi menggunakan suatu sisten yang mirip dengan *JVM (Java Virtual Machine)*. Hanya berbeda dengan *Java*, tidak ada penghalang bahasa dengan *.NET* sehingga aplikasi dapat dikembangkan menggunakan bahasa : *VB, C++, C#, J#* dan 20 bahasa pemrograman lainnya yang kompatibel dengan *.NET Framework*.

Tujuan dari *.NET Framework* adalah :

- Menyediakan lingkungan pemrograman berorientasi objek, apakah kode objek disimpan dan dijalankan secara lokal, dijalankan secara loka tetapi disebarkan melalui *internet* atau dijalankan secara remote (dijalankan melalui suatu tempat).
- Menyediakan lingkungan untuk menjalankan suatu kode yang menjamin keamanan saat kode dijalankan, termasuk kode yang dibuat oleh pihak yang tidak diketahui atau pihak ketiga yang setengah dipercaya.
- Menyediakan lingkungan untuk menjalankan suatu kode yang dapat mengeliminasi masalah performa dari lingkungan *scripted* dan *interpreted*.
- Membuat pengembang memiliki pengalaman yang konsisten dalam berbagai aplikasi seperti aplikasi berbasis windows dan aplikasi berbasis web.

Definisi lain dari *.NET framework* bisa juga diartikan sebagai serangkaian teknologi yang :

- mempersatukan aplikasi web yang sekarang masih terisolasi.
- membuat informasi yang tersedia disetiap saat dan disetiap tempat.
- Menyederhanakan pengembangan dan penyebaran aplikasi

### 2.8.2. ASP.NET

*ASP.NET* adalah teknologi baru dalam pemrograman web yang merupakan kelanjutan dari teknologi *ASP 3.0*. Perbedaan utama disbanding *ASP* klasik adalah penggunaan *.NET Framework* sebagai fondasi pemrograman. *ASP.NET*

memiliki beberapa kelebihan dibandingkan teknologi terdahulu, antara lain:

1. Kemudahan mengakses berbagai *library .NET Framework* secara konsisten dan *powerful*, yang mempercepat pengembangan aplikasi.
2. Penggunaan berbagai bahasa pemrograman secara penuh, misalnya *VB.NET*, *C#*, *J#*, dan *C++*. Dalam *ASP.NET* bahasa-bahasa ini dapat digunakan secara penuh sebagaimana layaknya bekerja di *Windows Application*. Hal ini jauh berbeda dibandingkan *ASP* klasik yang menggunakan *scripting language*. Misalnya *VBScript* agak berbeda syntaxnya dengan *VB*, dalam *ASP.NET* kita dapat menggunakan *VB.NET* dengan *syntax* yang 100% sama dengan ketika membangun *Windows Application*. Demikian juga dengan *C#*, tidak ada perbedaan *syntax* antara *C#* untuk membuat kode *ASP.NET* dengan *Windows Form*. Ini sangat memudahkan memprogram aplikasi *Windows* dan *Web* secara konsisten.
3. Tersedia berbagai *Web Control* yang dapat digunakan membangun aplikasi secara cepat. Programmer dapat dengan mudah mengkaitkan data ke *web control* sebagaimana layaknya memprogram *windows application*. Hal ini sangat mempercepat pembuatan aplikasi dibandingkan harus menyusun kode-kode *HTML* secara manual.

*Code Behind*, artinya kode-kode pemrograman yang menjadi logic aplikasi ditempatkan terpisah dengan kode *user interface* yang berbentuk *HTML*. Ini sangat memudahkan dalam *debugging*, karena kode untuk

*presentation layer* tidak tercampur dengan kode *application logic*. (<http://ilmukomputer.com/>)

### **2.8.3. VB.NET**

*Visual Basic.NET (VB.NET)* adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak diatas sistem *.NET Framework*, dengan menggunakan bahasa *BASIC*. Dengan menggunakan alat ini, para programmer dapat membangun aplikasi *Windows Form*, dan juga aplikasi web berbasis *ASP.NET*, dan juga aplikasi *comand line*. Bahasa *Visual Basic* sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai revolusi dari *Microsoft Visual Basic* versi sebelumnya yang diimplementasikan diatas *.NET Framework*. (Ariyo Suryo Kusumo, Drs, *Visual Basic .NET, Elex Media Komputindo, Jakarta, 2004*)

### **2.9. Web server**

*Web server* adalah sebuah perangkat lunak server yang berfungsi menerima permintaan *HTTP* atau *HTTPS* dari klien yang dikenal dengan *browser* web dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk *document HTML*.

*Server* web yang terkenal diantaranya *Apache* dan *Microsoft Internet information Service (IIS)*. *Apache* merupakan *server* web antar *platform*, sedangkan *IIS* hanya dapat beroperasi di Sistem Operasi *Windows*. (<http://id.wikipedia.org>)

### **2.10. XML, Javascript, dan AJAX**

### 2.10.1. XML

*XML* kependekan dari *eXtensible Markup Language*, dikembangkan mulai tahun 1996 dan mendapatkan pengakuan dari *W3C* pada bulan Februari 1998. Teknologi yang digunakan pada *XML* sebenarnya bukan teknologi baru, tapi merupakan turunan dari *SGML* yang telah dikembangkan pada awal 80-an dan telah banyak digunakan pada dokumentasi teknis proyek-proyek berskala besar. Ketika *HTML* dikembangkan pada tahun 1990, para pengagagas *XML* mengadopsi bagian paling penting pada *SGML* dan dengan berpedoman pada pengembangan *HTML* menghasilkan markup language yang tidak kalah hebatnya dengan *SGML*.

Seperti halnya *HTML*, *XML* juga menggunakan elemen yang ditandai dengan tag pembuka (diawali dengan '<' dan diakhiri dengan '>'), tag penutup (diawali dengan '</' dan diakhiri '>') dan atribut elemen (parameter yang dinyatakan dalam tag pembuka misal <form name="isidata">). Hanya bedanya, *HTML* mendefinisikan dari awal tag dan atribut yang dipakai didalamnya, sedangkan pada *XML* kita bisa menggunakan tag dan atribut sesuai kehendak kita. Untuk lebih jelasnya lihat contoh dibawah:

```
<pesan>
<dari>MIS Manager</dari>
<buat>HRD Manager</buat>
<buat>Bagian rekrut</buat>
<buat>Computer Suport team</buat>
<subyek>Permohonan Tenaga kerja baru</subyek>
```

```
<isi>Mohon diberikan tenaga kerja baru untuk mengisi  
lowongan di Departemen MIS</isi>  
</pesan>
```

Pada contoh diatas <pesan>, <dari> <buat>, dan <isi> bukanlah tag standard yang telah ditetapkan dalam XML. Tag-tag itu kita buat sendiri sesuai keinginan kita. Sampai di sini XML tidak melakukan apapun. Yang ada hanyalah informasi yang dikemas dengan tag-tag XML. Kita harus membuat software lagi untuk mengirim, menerima atau menampilkan informasi di dalamnya. ([www.ilmukomputer.com](http://www.ilmukomputer.com), 2007)

#### 2.10.2. Javascript

JavaScript adalah nama implementasi Netscape Communications Corporation untuk ECMAScript standar, suatu bahasa skrip yang didasarkan pada konsep pemrograman berbasis *prototipe*. Bahasa ini terutama terkenal karena penggunaannya di situs web (sebagai JavaScript sisi klien) dan juga digunakan untuk menyediakan akses skrip untuk objek yang dibenamkan (*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 terdaftar dari Sun Microsystems, Inc. Bahasa ini digunakan di bawah lisensi untuk teknologi yang diciptakan dan diimplementasikan oleh

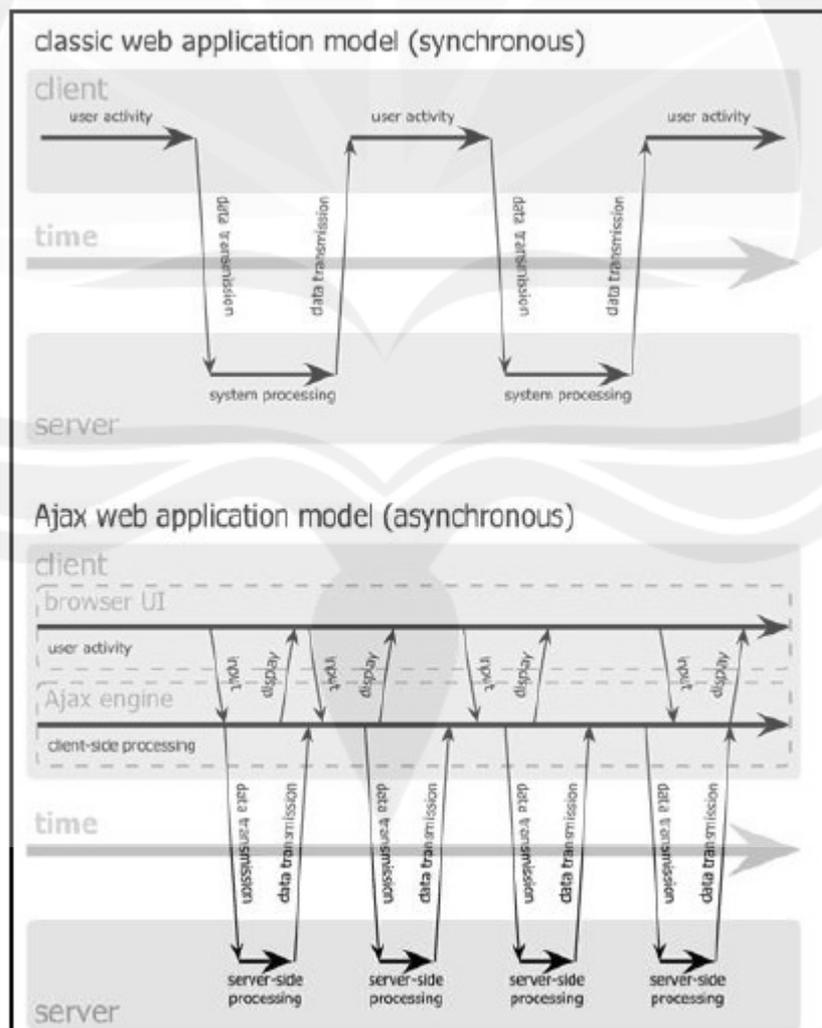
*Netscape Communications* dan entitas barunya seperti *Mozilla Foundation*. (<http://id.wikipedia.org>, 2007)

### 2.10.3. AJAX

*Asynchronous JavaScript And XML*, atau disingkat *Ajax*, adalah suatu teknik pemrograman berbasis web untuk menciptakan aplikasi web interaktif. Istilah *AJAX* digunakan pada website yang berinteraksi dengan server melalui *JavaScript* secara asinkron (*background*), sehingga pengguna tidak perlu untuk meload keseluruhan halaman. Hal ini menyebabkan pengiritan waktu dan *bandwidth*, dan juga menghasilkan website yang makin interaktif. Website yang dijadikan contoh *AJAX* sejati adalah *Gmail*. Bagi yang pernah menggunakan *Gmail*, tentu merasakan kehandalan *interface* webmailnya dengan *AJAX*.

Untuk mengaplikasikan *AJAX* (*Asynchronous JavaScript and XML*) dalam website, yang dibutuhkan adalah browser dengan kemampuan *JavaScript* dan komponen *XMLHTTP* bagi pengguna *Internet Explorer* dan *XMLHttpRequest* untuk *Firefox* dan browser lainnya. Syarat-syarat tersebut hampir pasti sudah terpenuhi, mengingat pengguna internet kita banyak yang menggunakan *Internet Explorer* dan *Firefox*. Sebenarnya tidak ada hal baru dalam *AJAX*, karena yang digunakan adalah teknologi *JavaScript*, yang notabene sudah lama digunakan. Karena itu, bagi yang sudah terbiasa dengan *JavaScript*, maka akan mudah sekali mempelajari *AJAX*.  
([www.sonyak.com](http://www.sonyak.com))

*ASP.NET AJAX*, merupakan salah satu bahasa pemrograman yang mendukung *AJAX*. *AJAX ASP.Net*, memungkinkan pengguna untuk membuat suatu halaman web yang cepat dengan memadukan fitur yang kaya dan elemen *user interface* yang umum. *AJAX ASP.Net* menyediakan pustaka *client-script* yang bekerja pada bahasa antar-brower *ECMAScript (JavaScript)* dan teknologi *dynamicHTML (DHTML)* digabungkan dengan *ASP.Net 2.0 Server-based development platform*. Dengan *AJAX ASP.Net*, pengguna dapat mengembangkan kemampuan *user* dengan aplikasi web yang efisien. Gambar 2.1 adalah perbandingan pemrosesan web tanpa Ajax dengan web Ajax.



### Gambar 2.2 Model Pemrosesan Ajax

Dari gambar tersebut dapat dilihat bahwa interaksi dengan *user* dapat terus berlangsung pada halaman berbasis Ajax. Adapun cara kerja dari Ajax adalah sebagai berikut :

1. Inisial *request* yang dilakukan oleh *browser*, dimana *client* me-*request* suatu URL ke *server*.
2. Halaman *Web* yang lengkap dikirimkan oleh *server* ke *Web browser* yang terdapat disisi *client*
3. Semua *request* yang akan dikirim ke *server* kemudian di inisialisasikan sebagai suatu *function call* ke *AJAX engine*
4. *AJAX engine* kemudian akan membuat sebuah *XmlHttpRequest* yang kemudian dikirim ke *server*
5. *Server* memproses *request* tersebut dan mengirimkan responnya ke *client* dalam format XML. Dokumen XML yang dikirimkan oleh *server* hanya terdiri atas data-data yang diperlukan untuk mengubah elemen-elemen tertentu dari suatu halaman *Web*.
6. *AJAX engine* memproses *response* dari *server*, kemudian mengupdate bagian tertentu dari suatu halaman *Web* yang perlu diupdate atau melakukan operasi terhadap data yang diterima dari *server*.

Ajax merupakan kombinasi dari:

1. *XHTML* (atau *HTML*) dan *CSS* untuk bahasa *mark up* dan tampilan.
2. *DOM* yang diakses dengan *client side scripting language*, khususnya implementasi *ECMAScript* seperti *JavaScript* dan *JScript*, untuk

menampilkan secara dinamis dan berinteraksi dengan informasi yang ditampilkan

3. Objek *XMLHttpRequest* untuk melakukan pertukaran data asinkronus dengan web server. Pada beberapa *framework* dan kasus *Ajax*, objek *IFrame* lebih dipilih daripada *XMLHttpRequest* untuk melakukan pertukaran data dengan web server.
4. *XML* umumnya digunakan sebagai format untuk pengiriman data, walaupun format lain juga memungkinkan, seperti *HTML*, *plain text*, *JSON* dan *EBML*.

Seperti halnya *DHTML*, *LAMP*, atau *SPA*, *Ajax* bukanlah teknologi spesifik, melainkan merupakan gabungan dari teknologi yang dipakai bersamaan. Bahkan, teknologi turunan/komposit yang berdasarkan *Ajax*, seperti *AFLAX* sudah mulai bermunculan. (Ferdinand, 2007)