

Bab II

Landasan Teori

2.1 Internet

Internet (kependekan dari kata *inter-network*) adalah jaringan komputer yang saling terhubung ke beberapa jaringan komputer yang lain dalam ruang lingkup yang sangat besar, mencakup seluruh dunia. Internet adalah sistem komputer umum, yang terhubung secara global dan menggunakan *TCP/IP* sebagai protokol pertukaran paket (*Packet Switching Communication Protocol*) (Wikipedia, 2008). Cara menghubungkan rangkaian dengan kaedah ini dinamakan *internetworking*.

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 disumbang oleh *ARPANET* termasuk kaedah rangkaian tanpa-pusat (*decentralised network*), teori *queueing*, dan kaedah pertukaran paket (*packet switching*).

Pada 01 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 menghubungkan kebanyakan pengguna jaringan-jaringan komputer yang ada.

Intenet 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, adalah *email* atau surat elektronik, *Usenet, Newsgroup*, berbagi file (*File Sharing*), *WWW (World Wide Web)*, akses sesi (*Session Access*), *IRC, MUD, dan MUSH*. Di antara semua ini, *email* atau surat elektronik dan *World Wide Web* lebih sering digunakan, dan lebih banyak layanan yang dibangun yang berdasarkan *email* dan *World Wide Web*, seperti *milis (Mailing List)* dan *Weblog*. Internet memungkinkan adanya layanan terkini (*Real-time service*), seperti *web radio*, dan *webcast*, yang dapat diakses di seluruh dunia. Beberapa layanan internet populer yang didasarkan pada sistem tertutup (*Proprietary System*), adalah seperti *IRC, ICQ, AIM, CDDb, dan Gnutella*.

2.2 Web Service

Web service adalah sebuah program yang menerima dan merespon permintaan informasi (Wikipedia, 2008). Pada umumnya, sebuah *web service* menerima permintaan berdasarkan format *XML*. Format dari permintaan dan respon tergantung dari standar *XML* yang digunakan. Salah satu standar yang sering digunakan adalah *Simple*

Object Access Protokol (SOAP). Ada *public registry* dan bahasa seperti *UDDI*, *WSDL*, dimana menggunakan katalog untuk *web service* berbeda. Program panggilan yang dapat menayangkan *registry* seperti *Universal Description Discovery and Integration (UDDI)* untuk menemukan *web service* yang tepat, lalu menggunakan *WSDL* untuk menentukan parameter yang membutuhkan *service* dan akhirnya menggunakan protokol pemanggil dan *XML* standar seperti *SOAP* untuk secara aktual menghubungi *web service*.

Dilihat dari mekanisme kerja *web service* diatas, terdapat beberapa protokol standat yang digunakan yaitu:

a. *Extendible Markup Language (XML)*

XML adalah bahasa yang digunakan untuk menampilkan banyak tipe data (Wikipedia, 2008). *XML* hampir sama dengan *HTML* (keduanya keturunan dari *SGML*, sebuah pembangkit bahasa *markup*). Dimana *HTML* digunakan untuk menyatakan *web browser*, bagaimana menampilkan informasi pada end user. *XML* lebih umum digunakan untuk mengirimkan informasi diantara program. *File XML* seringkali tidak mempunyai informasi tentang *display* informasi program menerima data, seringkali menggunakan gaya *XSL sheet* dan *XSLT*, biasanya menangani secara terbatas. Struktur dari file *XML* biasanya didefinisikan oleh *Document Type Definition (DTD)* atau *XML Schema Definition (XSD)*.

b. *Web Service Description Language (WSDL)*

WSDL adalah format *XML* yang buat untuk menerangkan *web service* (Wikipedia, 2008). *WSDL* mendefinisikan :

- 1) Pesan-pesan (baik yang abstrak maupun yang konkret) yang dikirim ke dan menuju *web service*.
- 2) Koleksi-koleksi digital dari pesan-pesan (*port type*, antar muka).
- 3) Bagaimana *port type* yang ditentukan dijadwalkan *wire protokol*.
- 4) Dimana *service* ditempatkan.

WSDL sangat penting bagi portal karena portal pada umumnya mengumpulkan informasi dari berbagai macam *servis web* dalam layer tunggal dan oleh karena itu membutuhkan komunikasi satu sama lain dalam format yang tepat.

c. *Simple Object Access Protokol (SOAP)*

SOAP adalah standar untuk bertukar pesan berbasis *XML* melalui jaringan komputer atau sebuah jalan untuk program yang berjalan pada suatu *operating system (OS)* untuk berkomunikasi dengan program pada *OS* yang sama maupun *OS* yang berbeda dengan menggunakan *HHTP* atau file *XML* sehingga program pada suatu komputer dapat memanggil program pada komputer lain dan mengirimkan informasi serta bagaimana program yang dipanggil memberikan tanggapan (Wikipedia, 2008).

SOAP merupakan *platform independent*

karena memiliki basis *XML*, dengan demikian *SOAP* secara cepat menjadi protokol terdepan dalam mengirim dan menerima hasil *web service*.

d. *Universal Description Discovery and Integration (UDDI)*

UDDI berdasar pada spesifikasi untuk menemukan *web service* dan *public registry* dimana *web service* dapat mempublikasikan informasi tentang dirinya. *UDDI* telah secara luas mendukung semua elemen dari industri internet.

Keuntungan penggunaan *Web Service* :

- a. Format penggunaan terbuka untuk semua *platform*.
- b. Mudah di mengerti dan mudah dalam melakukan *debuging*.
- c. Dukungan *interface* yang stabil.
- d. Mudah untuk menengahi pesan-pesan proses dan menambahkan nilai.
- e. *Routing* dan *Transporting*.
- f. *Security*.
- g. *Management* dan *Monitoring*.
- h. *Schema* dan *Service design*.
- i. *Acceleration*.
- j. Mudah untuk mengembangkan dengan *semantic transport* tambahan.
- k. Terbuka, standard-standard berbasis teks.
- l. Pencapaian modular.
- m. Tidak mahal untuk diimplementasikan (relatif).
- n. Mengurangi biaya integrasi aplikasi *enterprise*.
- o. Implementasi yang incremental.

2.3 .Net Framework

Salah satu vendor yang mengusulkan konsep *web service* adalah *Microsoft*. *Microsoft* memperkenalkan nama *Web Service* pada juni 2000 (Wikipedia, 2008). Platform *Microsoft* untuk XML web service adalah *. Net*.

.NET Framework merupakan suatu komponen *Windows* yang terintegrasi yang dibuat dengan tujuan pengembangan berbagai macam aplikasi serta menjalankan aplikasi generasi mendatang termasuk pengembangan aplikasi *XML Web Services*.

Keuntungan Menggunakan *.NET Framework* adalah :

- a. Mudah. Yang dimaksud mudah di sini adalah kemudahan *developer* untuk membuat aplikasi yang dijalankan di *.NET Framework*. Mendukung lebih dari 20 bahasa pemrograman : *VB.NET, C#, J#, C++, Pascal, Phyton (IronPhyton), PHP (PhLager)*.
- b. Efisien. Kemudahan pada saat proses pembuatan aplikasi, akan berimplikasi terhadap efisiensi dari suatu proses produktivitas, baik efisien dalam hal waktu pembuatan aplikasi atau juga efisien dalam hal lain, seperti biaya.
- c. Konsisten. Kemudahan-kemudahan pada saat proses pembuatan aplikasi, juga bisa berimplikasi terhadap konsistensi pada aplikasi yang kita buat. Misalnya, dengan adanya *Base Class Library*, maka kita bisa menggunakan objek atau *Class* yang dibuat untuk aplikasi berbasis *windows* pada aplikasi berbasis *web*. Dengan

adanya kode yang bisa dintegrasikan ke dalam berbagai macam aplikasi ini, maka konsistensi kode-kode aplikasi kita dapat terjaga.

- d. Produktivitas. Semua kemudahan-kemudahan di atas, pada akhirnya akan membuat produktivitas menjadi lebih baik. Produktivitas naik, terutama produktivitas para *developer*, akan berdampak pada meningkatnya produktivitas suatu perusahaan atau project.

.Net Framework memungkinkan *developer* untuk membuat aplikasi yang *reusable* dan *interpolable* dengan menyediakan kumpulan umum *Application Programing Interface (API)* untuk semua bahasa pemrograman. *.Net Framework* telah mendukung berbagai pengembangan aplikasi dan menjalankan berbagai aplikasi untuk generasi berikutnya.

.Net infrastruktur membentuk *framework* dimana *web service* dapat dibuat. Terdapat tiga komponen utama dari *.Net* yaitu :

- a. *Common Language Runtime (CLR)*

Semua bahasa pemrograman memiliki runtime, yaitu *service* yang berfungsi sebagai antarmuka bahasa pemrograman. *CLR* merupakan komponen inti dari *.Net* yang menyediakan dasar untuk membuat aplikasi berbasis *.Net*. *CLR* menyediakan *runtime* umum yang bisa digunakan di berbagai bahasa. Eksekusi aplikasi *.Net* dikelola oleh *CLR*.

Sebagai sebuah landasan agar program *.NET* berjalan dengan baik, *CLR* mempunyai beberapa tugas yaitu :

- 1) Mengatur dan mengelola eksekusi kode program.
- 2) Mengelola alokasi memori ketika program dieksekusi.
- 3) Memberikan layanan seperti *Just In Time (JIT)* kompilasi.
- 4) Mengelola penanganan kesalahan atau eksepsi.
- 5) Melakukan proses *debug* dan mengatur keamanan program.
- 6) Bersifat *assembly* saat proses *deployment*.
- 7) Memungkinkan adanya variasi dan integrasi berbagai bahasa pemrograman yang ada di lingkungan *.NET framework*.

b. Unified Programing Classes

Programing *class libraries* atau *Application Programing Interfaces (API)* digunakan oleh bahasa pemrograman yang berbeda. Untuk menggunakan bahasa pemrograman yang berbeda, developer harus mempelajari kumpulan *class libraries* yang berbeda juga. Hal ini akan memperlambat waktu pengembangan aplikasi. *.Net* menyediakan *unified programing class* dengan kumpulan umum *API* untuk semua bahasa pemrograman. Setiap bahasa pemrograman dapat berinteraksi satu sama lain, dan *unified programing class* memungkinkan developer untuk memilih bahasa pemrograman yang disukai dengan hanya menggunakan satu kumpulan *API*.

.Net Framework 3.5 adalah *framework* terbaru yang dikeluarkan oleh *Microsoft* yang memiliki fitur-fitur terbaru seperti :

- a. Integrasi dengan *Language Integrated Query (LINQ)*
- b. Integrasi *Windows Communication Foundation (WCF)* dan *Windows Workflow Foundation (WF)*, hal ini memungkinkan kita mengembangkan suatu *service* dengan mekanisme *long run* dan *workflow*.
- c. *WCF* kini tidak hanya berfokus pada dukungan skenario *entrepise* dan *web service*. *WCF* kini mendukung teknologi yang lebih luas seperti *AJAX*, *REST* dan *syndication*. Dengan demikian, kini *.Net Framework 3.5* dapat mengakses teknologi seperti *RSS*, *ATOM* dan lain-lain.
- d. *WCF* kini mendukung dan memberi kemudahan emplementasi *AJAX web 2.0* dan juga pengkodean ala *JSON*.
- e. Dukungan standar yang berkesinambungan bagi standar *web service* seperti *Atomic Transaction*, *Reliable Messaging*, *Secure Conversation*, dan *Coordination*.
- f. Dukungan partial trust pada aplikasi berbasis layanan *WCF* dan juga distribusi aplikasi berbasis *click once* dan *XBAD*.
- g. Penambahan beberapa fitur *class*, terutama fitur yang menfokuskan pada kebutuhan konsumen seperti tipe data *BigInteger*, *HashSet*, dan *DateTime2*, hingga *class NSA suite "B"* dan *FIPs compliant cryptography*.
- h. Perbaikan *Windows Presentation Foundation (WPF)* dari berbagai sisi, mulai dari performa, keamanan dan sebagainya.

- i. Perangkat desain yang terintegrasi untuk *WPF*, *WCF*, dan *WF*.

2.4 XAML

XAML (*eXtensible Application Markup Language*) merupakan "*new Declarative Programming way*" yang dikeluarkan oleh *Microsoft* (Wikipedia, 2008). Untuk membuat sebuah aplikasi kita dapat memilih salah satu teknik atau mungkin kedua-duanya yang telah disediakan oleh *Microsoft*, yaitu *Imperative way* dan *Declarative way*. *Imperative way* digunakan untuk membuat sebuah aplikasi atau kode di mana aturan-aturan atau prosedur atau *logic flow* kita yang menentukan. Sedangkan *Declarative way* memberikan kebebasan pada komputer untuk menentukan sendiri proses atau *logic flow* dari kode yang kita tuliskan.

XAML ("*zamel*") digunakan untuk membuat *User Interface (UI)* sebuah aplikasi, dimana kodenya dibuat dengan mengikuti standar penulisan *XML*. *XAML* ini mirip dengan *HTML* dilihat dari penulisan elemen-elemen dan atribut-atributnya. Namun ada perbedaannya *HTML* tidak bersifat *strong-typed*, artinya kalau kita tuliskan sebuah elemen yang tidak dikenal atau *misspelling* maka *browser* akan mengabaikannya, sedangkan *XAML* bersifat *strong-typed*, artinya semua elemen-elemen atau atribut-atribut harus dituliskan secara benar dan kalau ada salah penulisan maka kode tersebut tidak akan pernah bisa di eksekusi.

XAML ini bersifat *interpreted*. Akan tetapi sebelum sebuah aplikasi yang *XAML*-based di *deploy* maka *XAML* tersebut akan di konversi terlebih dahulu ke *Binary*

Application Markup Language oleh *compiler*. Elemen-elemen yang dituliskan di *XAML* file berkorespondensi dengan *CLR class*, dan atribut-atributnya berhubungan dengan properti dari *class* yang digunakan.

Kalau kita sudah terbiasa dengan pembuatan *asp.net* maka *XAML* tersebut mirip dengan *file presentation (.aspx)* dan *file application logic (.vb atau .cs)*. *XAML* memang dibuat untuk memisahkan *presentation layer (.xaml)* dengan *application logicnya (.xaml.cs)* yang dijadikan sebagai *declarative way* untuk membuat aplikasi berbasis *Windows Presentation Foundation*. *Windows Presentation Foundation* ini bisa digunakan untuk membuat aplikasi berbasis *windows* dan aplikasi berbasis *web*. Untuk yang berbasis *web* dikenal dengan teknologi *Windows Presentation Foundation / Everywhere (WPF/F)*.

Untuk membuat *file XAML* kita dapat menggunakan *notepad* atau *tools* yang dikeluarkan oleh *Microsoft* yaitu *Microsoft Expression Blend* yang sampai saat ini telah masuk pada versi *beta 2.5*. Dengan *Microsoft Expression Blend* ini mudah sekali bagi *developer* untuk membuat *file XAML*. Seperti halnya kita *drag and drop* sebuah *control* pada *web form* di *ASP.Net* maka hasil *drag and drop* tersebut akan menghasilkan elemen-elemen *HTML*. Begitu juga dengan *Microsoft Expression Blend* ini, kita bisa menghasilkan *file XAML* hanya dengan *drag and drop control* dan atur properti-propertinya maka *file XAML* akan otomatis dihasilkan. Selain dengan ada juga *Microsoft Interactive Designer* yang digunakan kalau kita ingin membuat aplikasi dengan animasi yang lebih kompleks.

Dengan teknologi-teknologi tersebut, untuk membuat sebuah aplikasi web yang penuh dengan berbagai macam animasi sangatlah mudah di *Windows Presentation Foundation / Everywhere* ini tanpa membutuhkan aplikasi *flash* dengan *action script* dan tanpa membutuhkan *java script* yang panjang. Teknologi presentation ini dapat berjalan di *Windows XP service pack 2*, *Windows Vista*, *Windows 2003 Server service pack 1* dan *Windows 2008 Server*.

2.5 Silverlight

Microsoft Silverlight adalah sebuah *plug-in cross-platform* dan *cross-browser* yang menyediakan dukungan terhadap *Rich Internet Applications* seperti animasi, *graphic vector*, dan permutaran audio dan video. *Silverlight* menawarkan model programming yang sangat fleksibel yang mendukung *AJAX*, *Visual Basic*, *Visual C#*, *Python*, dan *Ruby*, serta integrasi dengan aplikasi berbasis web yang sudah ada. *Silverlight* dibangun dibawah *Windows Presentation Foundation / Everywhere (WPF/F)*. Pada dasarnya *Silverlight* merupakan *sub-set* dari *WPF* yang sekaligus mendukung fitur-fitur *ASP.Net AJAX*.

Sampai dengan penulisan laporan ini, terdapat 2 versi *Silverlight* yang sudah tersedia, yaitu *Silverlight* versi 1.0 dan *Silverlight* versi 2.0. Versi 1.0 mendukung pengembangan *Rich Interactive Applications (RIAs)* yang menggunakan *Javascripts* sebagai bahasa pemrogramannya. Versi 2.0 memiliki semua fitur dari versi 1.0 serta mendukung *.Net Framework* dan *.Net Compliant programming languages* seperti *C#*, *VB*,

Python, Ruby serta *language integrated query (LINQ)*.

Silverlight sendiri merupakan aplikasi *client side* yang dibangun dengan menggunakan dasar *XAML* (*eXtensible Application Markup Language*). *XAML* terdiri atas baris-baris perintah yang mengikuti aturan penulisan *XML* yang dipakai untuk membentuk objek-objek yang diinginkan. *Silverlight* memakai *XAML* untuk membuat objek-objek grafis, animasi maupun multimedia.

Web merupakan sebuah lingkungan yang *stateless*. Skenario web konvensional adalah sebagai berikut, *request* untuk sebuah data dilakukan oleh *client* dengan mengirimkan pesan ke *web server*. Kemudian *web server* akan me-*respond* permintaan *client* dengan mengirimkan sebuah pesan *response*. Pesan yang dikirimkan antara *client* dan *server* mengandung teks sederhana. Aplikasi *Silverlight* dapat berkomunikasi dengan memanggil sebuah *web service*. Dalam hal ini dikarenakan *Silverlight* tidak mendukung *datasets*, maka dipakailah *arraylist* untuk menampung data yang diambil dari *SQL Server*. Walaupun *ASP.Net* dan *Silverlight* sama-sama menggunakan *.Net Framework* dan digunakan untuk pengembangan web, *Microsoft Silverlight* memiliki perilaku yang berbeda dengan *ASP.Net*. Kebalikan dari *ASP.Net*, semua proses pengekseskuan fungsi *Silverlight* terjadi di *client side* sehingga dapat mengurangi penggunaan sumber daya *server* secara berlebihan dan menambahkan performansi web di *client side*.

Silverlight mendukung pemanggilan fungsi secara *asynchronous*. Hal ini dapat membuat pengambilan data di *server* menjadi tidak kaku lagi, yaitu harus menunggu proses *request-response* sampai selesai sebelum dapat

melakukan proses yang lain.

Perbedaan antara eksekusi secara *synchronous* dan *asynchronous* memang tipis. Pada eksekusi secara *synchronous*, eksekusi program dijalankan dari baris-baris pertama program pada *source code* dimana setiap baris code dieksekusi secara berurutan (*sekuens*). Program akan dieksekusi satu baris tiap satu waktu. Setiap kali sebuah fungsi dipanggil, maka eksekusi program akan menunggu sampai fungsi tersebut selesai dijalankan, sebelum melanjutkan ke baris code selanjutnya. Cara eksekusi ini akan memiliki permasalahan yang tidak terduga. Misalnya ketika sebuah fungsi dipanggil, bagaimana jika user ingin menghentikan proses tersebut sebelum fungsi tersebut selesai dijalankan? Dengan eksekusi program secara *synchronous*, program akan menjadi "*stuck*" mengunggu proses selesai dijalankan, tanpa ada jalan keluar tengah. Eksekusi secara *asynchronous* menghindari kemacetan ini.

2.6 LINQ

LINQ merupakan singkatan dari *Language Integrated Query*, yaitu *query* yang terintegrasi dengan bahasa (Wikipedia, 2008). Artinya, kini bahasa pemrograman *C#* dan *VB* memiliki fitur baru (dan tentunya *syntax* baru) untuk mendukung *query* ke suatu sumber data. Sebelum adanya *LINQ*, kita menuliskan *query* di dalam suatu argument dari suatu *method*.

2.7 Wikipedia

Wikipedia merupakan ensiklopedia isi bebas yang dimulai pada tahun 2001 (versi bahasa Indonesia dimulai pada tahun 2003). Artikel-artikel di *Wikipedia* merupakan hasil kolaborasi oleh para penyumbang dari seluruh dunia. Situs ini merupakan situs *wiki*, yang berarti siapapun dapat menyunting artikel, memperbaiki dan menambahkan informasi, hanya dengan mengklik pranala sunting yang berada di atas setiap halaman.

Wikipedia merupakan merek dagang dari *Wikimedia Foundation Inc.* yang juga telah membuat keseluruhan keluarga *Wikipedia*, antara lain *Wikiquote*, *Wiktionary*, *Wikisource*, *Wikibooks* dan *Wikinews*. Di *Wikipedia* dan proyeknya yang lain, Anda diharapkan berani menyunting, dan turut menyumbangkan pengetahuan di bidang yang Anda minati.

Wikipedia dimulai pada 15 Januari 2001 oleh Jimmy Wales, Larry Sanger, dan beberapa kolaborator antusias bahasa Inggris. Kini ada 13.000 kontributor aktif yang mengerjakan 7.000.000 artikel dalam lebih dari 250 bahasa. Saat ini terdapat 91.774 artikel dalam bahasa Indonesia; setiap harinya ratusan ribu pengunjung dari seluruh dunia membuat puluhan ribu suntingan dan membuat ribuan artikel baru di *Wikipedia* berbahasa Inggris

2.8 Web 2.0

Inovasi dalam dunia web semakin hari kian mengalami perkembangan yang berarti, ini dibuktikan dengan adanya Teknologi Web 2.0 yang dikembangkan sekitar tahun 2004. Walaupun sudah termasuk lama kedengarannya oleh para praktisi web, namun sebagian

besar mereka masih bertanya-tanya tentang fungsi dan kegunaannya. 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 user. Tidak mengherankan bila suatu aplikasi (*software*) dapat diakses secara *online* tanpa harus menginstalnya terlebih dahulu. *Software* tersebut misalnya *software* pengolah kata (seperti *MS Word*) atau *software* pengolah angka (seperti *MS Excel*). Teknologi ke depan suatu *software* berbasisi web tidak lagi dijual melainkan suatu fasilitas gratis yang dapat digunakan setiap waktu. Permasalahan manajemen file juga tidak merepotkan, bahkan file dapat disimpan dan juga dapat di-sharing dengan user lain.

Kemudahan berinteraksi antara user dengan sistem merupakan tujuan dibangunnya teknologi Web 2.0. Interaksi tersebut tentunya haruslah diimbangi dengan kecepatan untuk mengakses, oleh karena itu diperlukan suatu bandwidth 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 di-*remix* oleh website lain dengan menggunakan tampilannya masing-masing dan dukungan pemrograman yang sederhana. Adanya kemajuan inovasi pada antar muka di sisi pengguna merupakan karakter dari Web 2.0.

