

BAB 2

LANDASAN TEORI

2.1 Sistem Informasi Berbasis Web

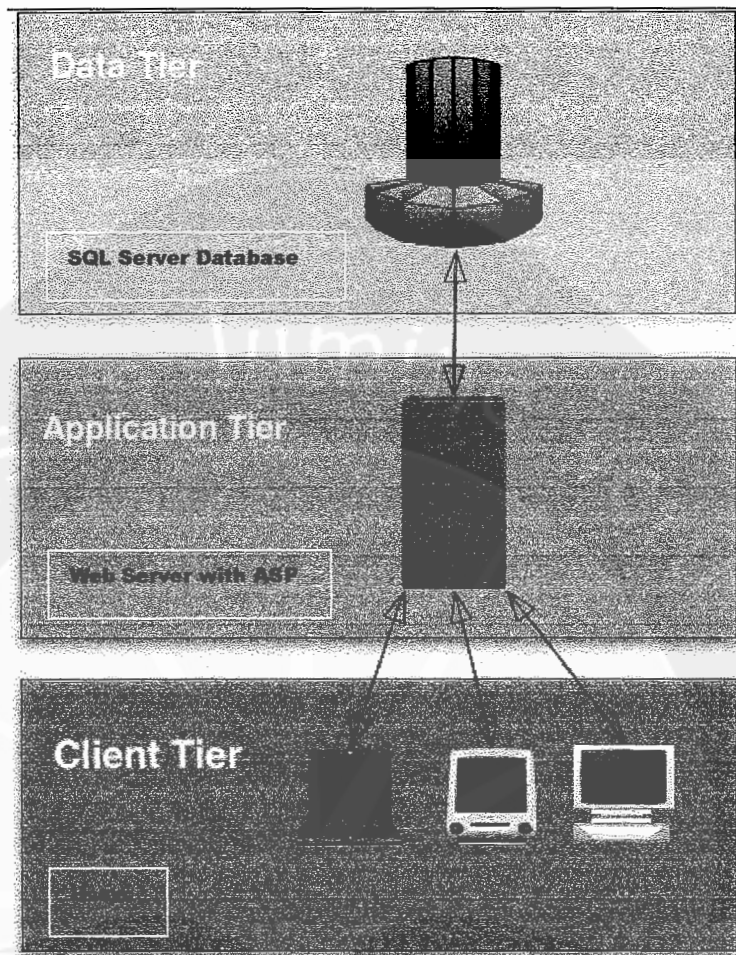
Saat ini komputer dan piranti pendukungnya telah masuk dalam setiap aspek kehidupan dan pekerjaan. Komputer tersebut dapat berdiri sendiri ataupun membentuk jaringan komputer dan internet yang menghubungkan berbagai tempat di dunia. *Interconnected Network* atau internet adalah sebuah sistem komunikasi global yang menghubungkan komputer - komputer dan jaringan - jaringan komputer di seluruh dunia, baik secara langsung maupun tidak langsung.

Web adalah fasilitas *hypertext* (sistem yang menghubungkan informasi satu ke informasi yang lain melalui suatu link) untuk menampilkan data berupa teks, gambar, bunyi, animasi dan data multimedia lainnya yang diantaranya data tersebut saling berhubungan satu sama lain. Untuk memudahkan membaca data dan informasi pada web dapat menggunakan web browser seperti *Microsoft Internet Explorer / Netscape* (Suryo, 2004).

2.1.1 Apa itu Aplikasi Berbasis Web?

Sebagian besar dari kita tentu sudah mengenal dan sering menggunakan web browser untuk mengakses internet. Secara sederhana, aplikasi yang berbasis web adalah aplikasi yang hanya membutuhkan web browser pada sisi client untuk mengakses informasi yang disimpan di server pusat. Di lingkungan akademis, sistem seperti

ini dikenal juga dengan aplikasi yang bersifat *Three Tier Application*.



Gambar 2.1 *Three Tier Application*

Pada contoh diatas, terlihat aplikasi yang dibangun menggunakan *SQL Server 2000 Database* pada sisi *Data Tier*, *Web server ASP* pada sisi *Application Tier* dan *Web Browser* pada sisi *Client Tier*. *Data Tier* dan *Application Tier* di-install pada mesin server, sedangkan pada *client* cukup di-install *Web Browser*. *Data Tier* program (sebaiknya menggunakan *My SQL*, *Oracle*, *SQL Server* maupun *Database Management System / DBMS* lainnya), bertugas mengelola semua data, sedangkan

Application Tier program bertugas memberikan layanan web bagi para *client* yang akan mengakses informasi menggunakan *web browser*.

2.1.2 Kemudahan Web Based Application

Terdapat beberapa kemudahan-kemudahan yang disediakan oleh sistem informasi yang dibangun berbasis web, antara lain :

1. *Data real time dan up to date.*

Semua *client* dapat meng-update data di sisi server (tentunya hanya data yang terkait dengan masing-masing pengguna), dan semua *client* juga dapat mengambil data yang sah (juga hanya data yang sudah diijinkan) setiap saat dari server selama terdapat koneksi *on line*.

2. *Integritas data tinggi.*

Dengan menggunakan DBMS yang tepat maka pengguna yang berbeda-beda akan dapat melihat data yang sama pada waktu yang bersamaan (karena bersumber dari 1 server yang sama), bukan data yang berlainan disetiap komputer pengguna.

3. *Pengaturan hak akses.*

Pengaturan hak akses ini juga akan meningkatkan keamanan data dari orang-orang yang tidak berhak terhadap bagian tertentu, walaupun orang tersebut sudah memiliki hak akses ke sistem. Disamping itu, dapat juga ditentukan siapa yang berhak melakukan perubahan data (*insert*, *update* dan *delete*) dan siapa yang boleh hanya melihat saja data tertentu.

2.2 Collaborative System

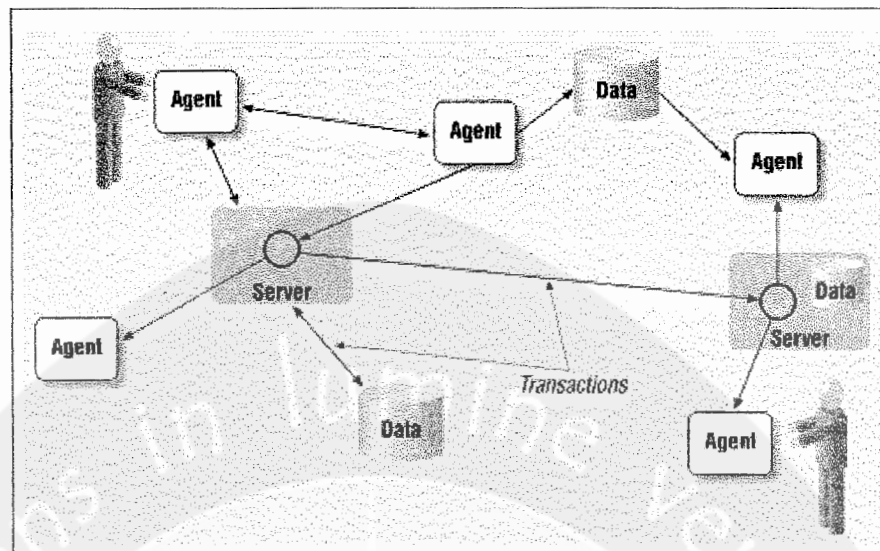
2.2.1 Pengertian Collaborative System

Collaborative System adalah sebuah sistem dimana berbagai user atau agen-agen bekerja dalam suatu aktivitas secara bersama. *Collaborative System* dapat dibedakan bahwa agen-agen di dalam sistem bekerja bersama dan memiliki kebutuhan untuk berinteraksi satu dengan lain, misalnya berbagi informasi, mengganti permintaan satu dengan lain.

Agen-agen dalam sistem saling berinteraksi dan bekerja dalam waktu yang bersamaan. Semua agen yang dilibatkan harus mengkoordinir satu dengan yang lain. Beberapa elemen yang dapat termasuk dalam *collaborative system* :

- Transaksi diantara *agents*, *servers*, dan data.
- Operational dan data *servers*.
- *Dynamic* dan *persistent data repositories* (penyimpanan data secara dinamis dan terus menerus).

2.2.2 Collaborative System Structure



Gambar 2.2 Collaborative System Structure

Agen, server, tempat penyimpanan data dan transaksi adalah semua elemen yang membangun *distributed system* secara umum, tetapi transaksi yang sesungguhnya adalah antara agen dan tujuan yang ingin dicapai oleh agen yang membuat sistem terkolaborasi. Beberapa contoh yang mengarah kepada *collaborative system* :

- *Interactive chat*, ini melibatkan *collaborative agents* dibawah kendali langsung manusia.
- *Coordinated data search*, melibatkan agen yang diprogramkan untuk bertindak secara otonomi.

Jadi, sebuah *collaborative system* dapat melibatkan transaksi antara agen, antara agen yang diprogramkan secara otonomi atau yang melibatkan keduanya.

2.3 Document Management System

Document Management System adalah sistem komputer untuk menyimpan, mengontrol dan mengakses informasi

secara elektronik dan juga digunakan untuk mengatur dokumen agar mudah dicari.

Atau dengan kata lain *document management* juga berarti mengelola / mengorganisasikan berbagai data dalam berbagai media, dari dokumen-dokumen kertas, email, *voice mail*, dan *electronic form*. Sehingga *Document Management* merupakan aktivitas untuk mengkonversi dokumen menjadi *file* elektronik. Kegiatan ini diawali dengan mengklasifikasikan (*classify*) kemudian disimpan (*storing*) untuk keperluan arsip.

Aplikasi *document management system* ini dibuat dengan menggunakan ASP .NET dengan bahasa pemrograman C# dan SQL Server sebagai *Database Management System (DBMS)*. Dalam sistem ini juga menggunakan Web Service untuk mendefinisikan fungsi-fungsi yang akan digunakan. Terdapat juga penggunaan CSS (*Cascading Style Sheet*) untuk membangun tampilan sebuah halaman web baik itu font, colour, background image, dll.

Diharapkan dengan pengembangan aplikasi *document management system* berbasis web ini, dapat mempermudah user dalam mengelola sebuah dokumen baik itu *upload*, *download*, *share document* maupun pengelolaan dokumen ke dalam sub-sub folder (folder dan sub folder dapat di ditambah, diubah namanya dan dihapus).

2.4 Web Service

Dalam pengertian sederhana, *XML Web Services* dapat di definisikan sebagai aplikasi yang diakses oleh aplikasi yang lain. *XML Web services* adalah *class* yang dikompilasi dan disimpan dalam sebuah *web server*. Sebuah *web services* tidak hanya dibuat dengan

menggunakan bahasa pemrograman Visual Basic .NET, tetapi juga bisa dibuat dengan menggunakan bahasa pemrograman lain yang mendukung penggunaan .NET Framework seperti Visual C#, Visual C++, Java dan yang lain (Suryo,2004).

XML Web Service menyimpan datanya dalam format XML dan karenanya menjadi multi platform dalam hal accessibilitynya. Web Service tersimpan di *Web Server* sehingga dapat diakses oleh berbagai bahasa pemrograman dengan lebih mudah baik dalam lingkungan LAN maupun Internet.

Sebuah *web service* sendiri diwakilkan dalam sebuah URI (*Universal Resource Identifier*) yang umumnya adalah sebuah alamat *web server* tersendiri. Dengan mengakses sebuah URI, maka *web services* menjadi sebuah *class* yang dapat diakses dari berbagai arah dan dengan berbagai cara baik untuk aplikasi berbasis Windows ataupun aplikasi berbasis web.

Di dalam .NET, sebuah *web services* lebih dikhususkan untuk aplikasi berbasis web sehingga akan sangat menguntungkan, baik bagi *programmer* maupun pihak pengguna. Bagi *programmer*, penggunaan *web services* akan mempermudah dalam pendefinisian dan penggunaan *class*. Selain itu, dengan adanya *web services*, *programmer* dapat dengan mudah melakukan revisi terhadap fungsi yang ada didalam *web services* tanpa harus melakukan *upgrade* terhadap program yang telah terinstalasi di komputer *client* (Suryo,2004).

Dengan ekstensi file default *asmx*, sebuah *web services* secara umum membutuhkan sebuah *web server* yang mendukung penggunaan ASP .NET dengan versi 1.0 hingga

2.0 yang telah terinstalasi dengan baik pada sistem operasi tertentu.

Sistem *Web Service* ini diharapkan meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam *Web Service* dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya.

2.5 CSS (*Cascading Style Sheet*)

2.5.1 Pengertian CSS (*Cascading Style Sheet*)

Teknologi dibalik *style sheets* disebut CSS. CSS adalah singkatan dari *Cascading Style Sheets*. CSS adalah sebuah bahasa yang mendefinisikan konstruksi sebuah style seperti *fonts*, warna dan posisi yang mana biasa digunakan untuk menggambarkan bagaimana informasi pada halaman web di format dan ditampikan. CSS digunakan dalam kode HTML untuk menciptakan kumpulan style yang akan memperluas kemampuan HTML. Dapat diartikan pula bahwa CSS adalah "gaya" dalam pemrograman HTML yang meminimalkan penulisan tag yang berulang-ulang.

CSS style dapat disimpan di halaman web HTML atau pada file style sheet terpisah. *Style sheet* mengandung perintah-perintah tentang style, dimana perintah-perintah tersebut akan mengaplikasikan style pada elemen-elemen tipe yang diiberikan. Ketika digunakan secara terpisah, maka perintah-perintah dalam style sheet ditempatkan pada external dokumen style sheet dengan extention file *.css*.

Style rule adalah sebuah instruksi tentang formatting yang dapat diaplikasikan kepada elemen dalam halaman web, seperti paragraf text atau sebuah *link*. *Style rule* terdiri dari satu atau lebih properti dan properti tersebut mengasosiasikan nilai - nilai. *Internal stlye sheet* ditempatkan langsung di dalam sebuah halaman web, sebaliknya *external style sheet* terdapat pada dokumen yang terpisah dan membutuhkan link yang sederhana untuk menuju pada halaman web melalui "tag" khusus.

"*Cascading*" adalah sebuah bagian dari singkatan CSS yang menunjuk pada cara dimana perintah-perintah dalam style sheet diaplikasikan pada elemen-elemen di dokumen HTML. Dengan *Style Sheet* dapat mengontrol seluruh *layout* dari web site anda, dan jika anda mengubah tampilan dari web site anda cukup dengan memodifikasi *style sheet*.

2.5.2 Kategori Style Sheets

Terdapat 4 kategori style sheets :

- a. *Inline Style Sheet* (di dalam elemen HTML).

Inline Style Sheet dilakukan dengan menambahkan atribut style kedalam tag HTML-nya. Berlaku hanya pada tag dimana style ditambahkan.

Contoh :

```
<p style="font-size: 25pt;
fontweight: bold; font-style:italic;
color:red;">
</p>
```

- b. *Emmbedded Style Sheet* (di dalam dokumen HTML).

Emmbedded Style Sheet diletakan pada tag <head>. Berlaku hanya untuk halaman dimana style didefinisikan.

Contoh : <head>

```
<title> Example Page </title>
```

```
<style type="text/css">
```

```
h1 { font-size: 16pt; font-weight:bold;}
```

```
p {color:blue;}
```

```
</style>
```

```
</head>
```

c. *External Style Sheet* (di file external).

External Style Sheet diletakan diluar dokumen HTML-nya. Menggunakan properti <link> untuk menghubungkan halaman HTML-nya dengan file .css nya.

Contoh : <link href="myfreetemplates.css"

```
rel="stylesheet" type="text/css" />
```

d. *Default Style Sheet* (style default dari browser).