

BAB II

LANDASAN TEORI

2.1 *Website*

Website adalah alamat atau lokasi di dalam *internet* suatu halaman *web*, umumnya membuat dokumen *HTML* dan dapat berisi sejumlah foto atau gambar grafis, musik, teks bahkan gambar yang bergerak. Dengan menggunakan teknologi tersebut, informasi dapat diakses selama 24 jam dalam satu hari dan dikelola oleh mesin. (Pardosi,2002:2).

Website adalah keseluruhan kumpulan halaman *web* dan informasi seperti gambar-gambar, suara, *file* video dan lain-lain yang disediakan bagi pengguna dalam sebuah *web server*. (Sudarmo, 2006:492).

Dari dua pendapat diatas *website* dapat disimpulkan sekumpulan halaman *web* milik seseorang atau suatu perusahaan dikumpulkan dan diletakan dalam sebuah situs *web*, yang umumnya bagian dari suatu nama domain (*domain name*) atau sub *domain di internet*.

2.2 *Web Service*

Web service diartikan sebagai sebuah antar muka (*interface*) yang menggambarkan sekumpulan operasi-operasi yang dapat diakses melalui jaringan, misalnya internet, dalam bentuk pesan XML. *Web service* diartikan sebagai sepotong atau sebagian informasi atau proses yang dapat diakses oleh siapa saja, kapan saja dengan menggunakan piranti apa saja, tidak terikat dengan sistem operasi atau bahasa pemrograman yang digunakan. (Kuntoro, 2005:1).

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. *Web service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu *web site* untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan *web service*.

Teknologi *web services* menawarkan kemudahan dalam menjembatani pulau-pulau informasi tanpa mempermasalahkan perbedaan teknologi yang digunakan masing-masing sumber. Misalkan sebuah situs informasi dibangun dengan menggunakan database *MySQL* sedangkan situs lainnya menggunakan *oracle* sedangkan anda sendiri menggunakan perangkat lunak *open source* dalam membangun situs web services akan mengatasi perbedaan ini.

Web services sebenarnya adalah kumpulan dari fungsi dan method yang terdapat pada sebuah *server* yang dapat dipanggil oleh klien dari jarak jauh, kemudian untuk memanggil *method-method* tersebut kita bebas menggunakan

aplikasi yang akan dibuat dengan bahasa pemrograman apa saja yang dijalankan pada *platform* apa saja. *Web Services* diperlukan karena pada masa sekarang ini perangkat keras, sistem operasi, aplikasi hingga bahasa pemrograman semakin beraneka ragam jenisnya. Keadaan tersebut dapat menimbulkan masalah dalam proses pertukaran data antar perangkat yang menggunakan aplikasi dan *platform* yang berbeda. (Febrian, 2007:34).

Dari dua pendapat diatas maka dapat disimpulkan bahwa *web service* diartikan sebagai sebuah antar muka (*interface*) yang menggambarkan sekumpulan operasi-operasi yang dapat diakses melalui jaringan, misalnya *internet*, dalam bentuk pesan XML.

2.2.1 Teknologi Protokol *Web Service*

Teknologi protokol *web service* merupakan protokol yang menghubungkan beberapa sistem yang saling berhubungan dan bisa bertukar informasi informasi dengan desentralisasi dan terdistribusi. SOAP dibangun dengan menggunakan protokol komunikasi HTTP. Karena HTTP didukung oleh semua *browser* dan *server*, maka protokol *web service* dapat berkomunikasi dengan berbagai aplikasi meskipun terdapat perbedaan sistem operasi, teknologi, dan bahasa pemrogramannya.

Peran teknologi *web service* adalah sebagai protokol pemaketan untuk pesan-pesan (*messages*) yang digunakan secara bersama oleh aplikasi-aplikasi penggunaannya. Spesifikasi yang digunakan tidak lebih seperti sebuah amplop biasa berbasis XML untuk informasi yang ditransfer, serta sekumpulan aturan bagi

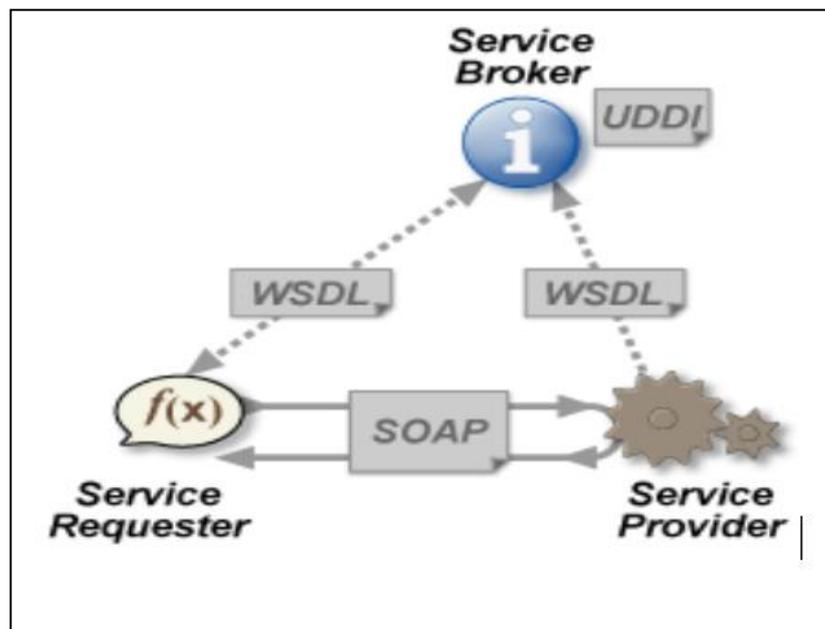
translasi aplikasi dan tipe-tipe data platform yang spesifik menjadi bentuk XML.

(Krisosa, 2001:1)

2.2.2 Cara Kerja *Web Service*

Web service memiliki cara kerja yang tergolong cukup mudah. Dengan memanfaatkan beberapa protokol standar yang sudah ada diantaranya : XML, WSDL, SOAP dan UDDI.

Simple Object Application Protocol (SOAP) dan beberapa teknologi yang didukung seperti *Web Service Description Language* (WSDL) dan *Universal Description Discovery and Integration* (UDDI) merupakan kombinasi dari XML yang dikirimkan melalui *HTTP*. SOAP menyediakan transmisi data yang sederhana dan *fleksibel*, pemanggilan fungsi, pengembalian nilai, *multiplatform*, dan dapat dikembangkan dengan menggunakan berbagai bahasa pemrograman.



Gambar 2.1 Cara Kerja *Web Service*

2.2.3 Kelebihan Web Service

Menurut Juliandri (2011:8), adapun kelebihan yang dimiliki oleh *web service* adalah:

1. *Web service* mempunyai sifat interoperability sehingga bisa diakses oleh aplikasi yang berjalan pada platform yang berbeda-beda.
2. *Web service* menggunakan standar dan protocol terbuka pada *internet*.
3. Dengan menggunakan HTTP atau SMTP, *web service* bisa menembus pengamanan *firewall* suatu organisasi tanpa mengubah konfigurasi *firewall*.
4. *Web service* memungkinkan fungsi-fungsi pada banyak perangkat lunak di internet untuk dipadukan menjadi satu web service baru.
5. *Web service* memungkinkan penggunaan ulang layanan dan komponen.
6. *Web service* bersifat loosely-coupled terhadap *client*.

2.3 SMS Gateway

Salah satu mode komunikasi yang handal saat ini adalah pesan pendek *short messaging system (SMS)*. Implikasinya, salah satu model komunikasi data yang bisa dipakai adalah SMS. Artinya, SMS tersebut harus bisa melakukan transaksi dengan database. Untuk itu perlu dibangun sebuah sistem yang disebut sebagai *SMS Gateway*. (Fikri, 2010:3).

Secara khusus, sistem ini akan memiliki fungsi-fungsi sebagai berikut:

1. *Message Management* dan *Delivery*
 - a. Pengaturan pesan yang meliputi manajemen prioritas pesan, manajemen pengiriman pesan, dan manajemen antrian.

- b. Pesan yang dilalukan harus sedapat mungkin *fail safe*. Artinya, jika terdapat gangguan pada jaringan telekomunikasi, maka system secara otomatis akan mengirim ulang pesan tersebut.

2. Korelasi

Berfungsi untuk melakukan korelasi data untuk menghasilkan data baru hasil korelasi. Pada sistem yang terpasang saat ini, arsitektur lalu lintas data melalui SMS sudah terjalin cukup baik. Hanya saja, keterbatasan akses data dan tujuan informasi SMS yang belum terfokus menyebabkan banyaknya jawaban standar (*default replies*), penggunaan SMS *gateway* sebagai berikut :

- a. Manajemen Inventori.
- b. CRM (*Customer Relationship Management*), misalnya Rumah Makan, Cafe, Executive Club, Stasiun Radio, Stasiun TV, Lembaga Pendidikan.
- c. *Call Center* dan SMS Pengaduan, misalnya Polisi, PLN, PAM, Instansi Pemerintah *SMS Gateway* memanfaatkan arsitektur teknologi komunikasi SMS untuk menerapkan aplikasi bernilai tambah dengan memanfaatkan komunikasi SMS untuk optimalisasi proses bisnis perusahaan dan peningkatan kualitas layanan dari institusi pelayanan publik.

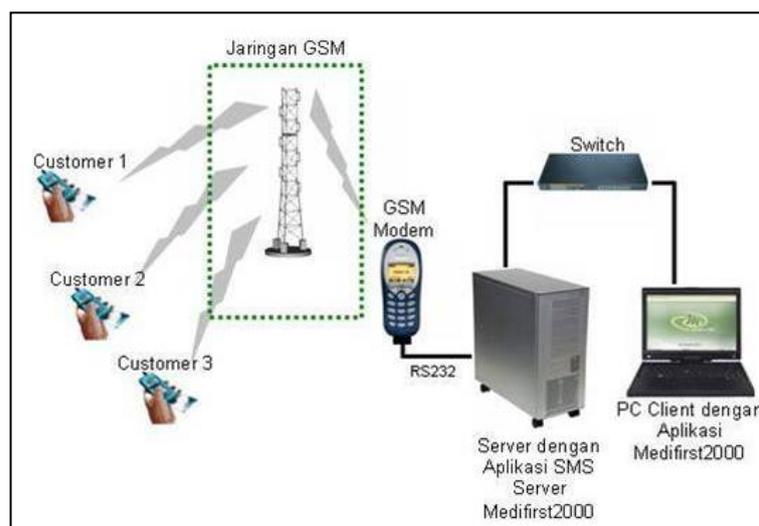
Beberapa kemampuan *SMS Gateway*, yaitu untuk:

- 1) Memperbesar skala aplikasi teknologi informasi dengan menggunakan komunikasi SMS interaktif.
- 2) Menyediakan aplikasi kolabo-rasi komunikasi SMS berbasis *web* untuk pengguna di institusi atau perusahaan.

- 3) Menjangkau konsumen maupun pengguna jasa layanan institusi atau perusahaan secara mudah menggunakan komunikasi SMS interaktif.

Cara kerja SMS *gateway* dapat dijelaskan sebagai berikut:

1. SMS dikirim oleh user ke telepon selular server.
2. SMS yang masuk ke telepon selular server diambil oleh Gammu dan dimasukkan ke dalam tabel *inbox*.
3. SMS pada telepon selular server akan dihapus saat Gammu mengambil pesan SMS tersebut.
4. SMS yang terdapat pada tabel *inbox* akan diproses oleh PHP. Pesan SMS yang telah diproses oleh PHP dinamakan *autoreply*.
5. *Autoreply* SMS akan di-*query* melalui PHP kemudian dimasukkan kedalam tabel *outbox*.
6. *Autoreply* pada tabel *outbox* di salin kedalam tabel *sentitems*.
7. *Autoreply* pada tabel *sentitems* di ambil oleh Gammu dan dikirim ke telepon selular server dan selanjutnya dikirim ke telepon selular user.



Gambar 2.2 Cara Kerja *SMS Gateway*

2.4 HTTP

HyperText Transfer Protocol (HTTP) merupakan protokol yang berfungsi untuk mendesain dan menjelaskan bagaimana *server* dan *client* berinteraksi dalam mengirim dan menerima dokumen *web*, protokol ini didisain untuk mentrasfer berkas yang berisi *hypertext* seperti berkas yang berisi HTML yang digunakan di *word wide web*. (Febrian, 2007:230).

HyperText Transfer Protocol (HTTP) adalah protokol untuk memindahkan *file hypertext* di *internet*. Hal ini memerlukan *HTTP client* program di suatu ujung dan *HTTP server* program di ujung yang lain.(Sudarmo, 2006:194)

Kesimpulan dari dua pendapat diatas adalah *HyperText Transfer Protocol (HTTP)* adalah *protocol* yang digunakan untuk mentrasfer dokumen dalam *word wide word (www)*. *Protocol* ini adalah protokol jaringan. Fungsi utamanya adalah untuk menciptakan hubungan antara *web server* dan mentransmisikan halaman *HTML* kepada pengguna *internet* yang sedang *browsing*. protokol ini didisain untuk mentrasfer berkas yang berisi *hypertext* seperti berkas yang berisi HTML yang digunakan di *word wide web*.

2.5 Apache

Apache adalah *server web* yang tersedia secara gratis dan disebarakan dengan lisensi (*open source*). *Apache* tersedia bagai bermacam-macam sistem operasi, seperti *UNIX (FreeBSD, Linux, Solaris* dan lainnya) dan *Windows*. *Apache* mengikuti standar protokol *hypertext transport protokol hypertext* transportasi protokol yaitu HTTP 1.1 (Febrian, 2007:32).

Apache adalah suatu program *server web* yang paling umum (*server HTTP*) dalam internet. *Apache* merupakan aplikasi terbuka yang awalnya diciptakan dari serangkaian perubahan yang dilakukan terhadap *server web*, dibuat. (Sudarmo, 2006:19).

Dari pendapat di atas *apache* dapat disimpulkan bahwa *server web* yang tersedia secara gratis dan disebar dengan lisensi "open source". *Apache* tersedia bagi bermacam-macam sistem operasi, seperti *UNIX (FreeBSD, Linux, Solaris* dan lainnya) dan *Windows*.

2.6 PHP

Hypertext Preprocessor (PHP) adalah bahasa scripting yang dapat menyatu (*embedded*) dengan kode-kode *HTML* dan dieksekusi di sisi *server (Server Side Scripting)*. Semua perintah yang ditulis akan dieksekusi oleh *server* dan hasil jadinya kode *HTML* dapat dilihat di *browser web*. (Sanjaya, 2004 : 36).

PHP adalah pemrograman yang digunakan untuk membuat *software* yang merupakan bagian dari sebuah situs *web*. *PHP* dirancang untuk berbaur dengan *HTML* yang digunakan untuk membuat halaman *web*. (Sudarmo, 2006:323)

Kesimpulan dari pendapat di atas adalah *Hypertext Preprocessor (PHP)* adalah bahasa scripting *open source* yang ditulis menggunakan sintaks bahasa *C, java* dan *perl*. *Script PHP* menyatu dengan file *HTML*.

2.7 MySQL

Mysql adalah *software* sistem manajemen *database*. *Database* adalah suatu koleksi data yang terstruktur. *Database* ini bisa berupa daftar belanja sederhana sampai informasi yang sangat besar dari suatu perusahaan internasional. Untuk menambahkan, mengakses dan memproses data disimpan di komputer. (Rickyanto, 2002: 32).

MySQL adalah suatu *database* populer dengan pengembangan *web developers*. Kecepatan dan ukuran yang kecil membuatnya ideal untuk *website*. Ditambah lagi dengan fakta bahwa *MySQL* adalah *open source* yang artinya gratis (Simarmata, 2006 : 29).

Dari dua pendapat di atas bahwa *MySQL* adalah *software* sistem manajemen *database*. *Database* ini bisa berupa daftar belanja sederhana sampai informasi yang sangat besar dari suatu perusahaan internasional. Untuk menambahkan, mengakses dan memproses data disimpan di komputer.

2.8 Unified Modeling Language

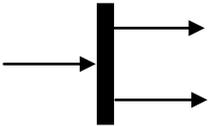
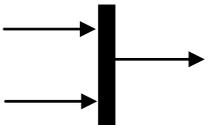
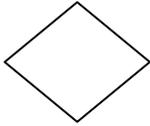
Unified Modeling Language (UML) merupakan salah satu bentuk language atau bahasa, menurut pencetusnya UML di definisikan sebagai bahasa visual untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek-aspek dari sebuah sistem. (Nugroho,2004:16).

Komponen *Unified Modeling Language* (UML) adalah :

1. **Activity Diagram**, *activity* diagram bersifat dinamis, diagram state ini memperlihatkan state-state pada sistem, diagram ini terutama penting untuk

memperlihatkan sifat dinamis dari antarmuka, kalaborasi dan pemodelan sistem-sistem yang reaktif.

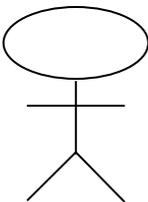
Tabel 2.1 Simbol *Activity Diagram*

No.	Simbol	Keterangan
1	<i>Start State</i> 	<i>Start state</i> adalah sebuah kondisi awal sebuah <i>object</i> sebelum ada perubahan keadaan.
2.	<i>End State</i> 	<i>End state</i> adalah menggambarkan ketika objek berhenti memberi respon terhadap sebuah event.
3.	<i>State/Activities</i> 	<i>State</i> atau <i>activities</i> menggambarkan kondisi sebuah entitas, dan digambarkan dengan segiempat yang pinggirnya.
4.	<i>Fork (Percabangan)</i> 	<i>Fork</i> atau percabangan merupakan pemisalah beberapa aliran konkuren dari suatu aliran tunggal.
5.	<i>Join (Penggabungan)</i> 	<i>Join</i> atau penggabungan merupakan penggabungan beberapa aliran konkuren dalam aliran tunggal.
6.	<i>Decision</i> 	<i>Decision</i> merupakan suatu logika aliran konkuren yang mempunyai dua cabang aliran konkuren.

Sumber : Nugroho, *Rational Rose Untuk Pemodelan Berorientasi Objek*, 2004.

2. **Use Case Diagram**, *use case* digram bersifat statis, diagram ini memperlihatkan himpunan *use case* dan aktor-aktor suatu jenis khusus dari kelas. Terutama sangat penting untuk mengorganisasikan dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.

Tabel 2.2 Simbol *Use Case*

No.	Simbol	Keterangan
1	Aktor 	Merupakan kesatuan <i>eksternal</i> yang berinteraksi dengan sistem.
2.	<i>Use Case</i> 	Rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem.
3.	<i>Generelation</i> 	Menggambarkan hubungan khusus atau interaksi dalam objek.

Sumber : Nugroho, *Rational Rose Untuk Pemodelan Berorientasi Objek*, 2004.

2.9 Penelitian Sebelumnya

1. Kuntoro, 2005. ”implementasi *web service* untuk pengembangan layanan pariwisata terpadu”. Sistem yang telah dibangun dimaksudkan untuk menunjukkan implementasi *web-service* dalam pengembangan aplikasi client dan integrasi antar *web-service* yang menggambarkan suatu sistem

multi-tier. *Web service* adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. *Web service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu web site untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan *web service*.

2. Berlilana, 2008. "**Web Services Sebagai Penyedia Layanan Administrasi Pada Penjualan Tiket Bus**". Saat ini perusahaan bus yang menyediakan jasa akomodasi angkutan dalam menangani reservasi dan penjualan tiket masih banyak yang menggunakan sistem tradisional dalam pelayanannya dimana agen yang ditunjuk akan mencatat pemesanan tiket dengan quota yang telah disesuaikan oleh kantor pusat, jika ada penambahan quota untuk jurusan tertentu pada agen tertentu, maka agen itu harus melakukan pengecekan ke agen lain atau kantor pusat untuk konfirmasi..