

BAB II KAJIAN PUSTAKA

A. *WEB SERVICE*

Selama ini kita pernah atau bahkan sering mendengar tentang *Web Service*. Apakah *Web Service* sama dengan *website*? Tentu saja tidak, *Web Service* dan *website* sangat berbeda. Perbedaan yang paling terlihat adalah *website* dibuat untuk memiliki tampilan atau *user interface* yang bagus, sedangkan *Web Service* tidak memiliki tampilan. *Web Service* tidak dibuat untuk berinteraksi langsung dengan *user*. Sesuai dengan kata *service* yang ada pada namanya, *Web Service* hanya menyediakan *service* atau layanan. Layanan inilah yang kemudian akan digunakan untuk dipanggil oleh aplikasi lainnya. Dengan demikian, yang akan menjadi *interface* adalah aplikasi yang memanggilmnya bukan *Web Service* itu sendiri. *Web Service* merupakan aplikasi yang dibuat agar dapat dipanggil atau diakses oleh aplikasi lain melalui *internet* maupun *intranet* dengan menggunakan XML sebagai format pengiriman pesan.

Web Service adalah kumpulan dari fungsi atau *method* yang terdapat pada sebuah *server* yang dapat dipanggil oleh *client* dari jarak jauh. Untuk memanggil *method-method* tersebut, kita bebas menggunakan aplikasi yang dibuat dengan bahasa pemrograman apa saja yang dijalankan diatas *platform* apa saja. Inilah keunggulan utama dari *Web Service* (Lucky, 2008).

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.

Web Service diciptakan untuk memecahkan masalah tersebut. Penggunaan *Web Service* memungkinkan perangkat-perangkat yang menggunakan sistem

operasi dan aplikasi yang berbeda satu sama lain dapat saling bertukar data dan informasi dengan mudah.

Web Service memiliki komponen-komponen pendukung dengan nama yang mungkin terdengar cukup aneh bagi yang pertama kali mendengarnya. Komponen-komponen tersebut merupakan dasar terbentuknya *Web Service*. Komponen tersebut antara lain:

1. *Extensible Markup Language (XML)*.

XML termasuk ke dalam keluarga *Markup Language* dan masih satu keluarga dengan HTML (*Hypertext Markup Language*) yang digunakan untuk membuat halaman *web*. Meskipun masih satu keluarga, kedua bahasa tersebut sebenarnya memiliki fungsi yang berbeda. Jika fungsi HTML lebih diutamakan untuk mengatur tampilan, maka fungsi XML lebih ditujukan untuk menyimpan data dan informasi. Selain itu ada satu perbedaan lagi antara HTML dan XML. Didalam XML kita bisa mendefinisikan tag milik kita sendiri. Contohnya adalah sebagai berikut:

```
<anggota>  
  <nama>Seseorang</nama>  
  <umur>25</umur>  
</anggota>
```

Seperti terlihat pada contoh diatas, kita bisa mendefinisikan tag kita sendiri untuk menjelaskan sebuah struktur data bernama anggota yang memiliki atribut nama dan umur.

XML merupakan dasar yang penting atas terbentuknya *Web Service*. *Web Service* dapat berkomunikasi dengan aplikasi-aplikasi yang memanggilnya dengan menggunakan XML. Kenapa menggunakan XML? Karena XML berbentuk *text* sehingga mudah untuk ditransportasikan menggunakan *protocol* HTTP. Selain itu XML juga bersifat *platform* independen sehingga informasi di

dalamnya bisa dibaca oleh aplikasi apapun pada *platform* apapun selama aplikasi tersebut mampu menerjemahkan tag-tag XML.

Kesimpulannya adalah apabila *Web Service* dan aplikasi dianggap sebagai manusia yang berbeda ras dan bahasa, maka XML adalah sebuah bahasa universal yang dapat mempersatukan mereka digunakan untuk saling berkomunikasi dan bertukar informasi (Lucky, 2008).

2. *Simple Object Access Protokol (SOAP)*.

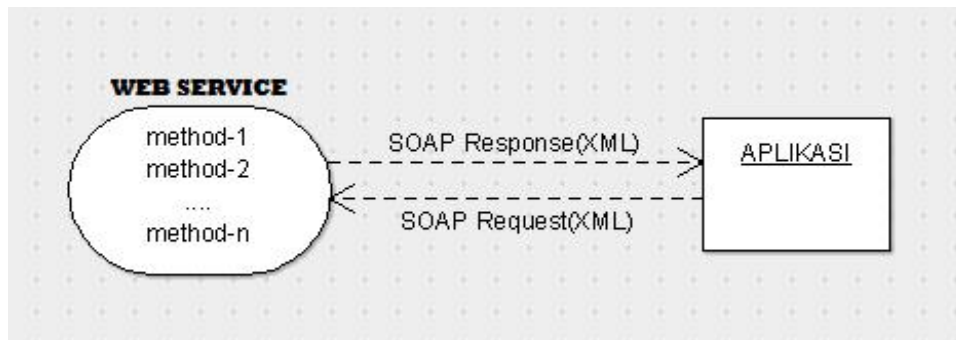
Bayangkan pada suatu hari Anda sedang berjalan-jalan dan bertemu dengan seorang turis. Turis tersebut kemudian mengajak Anda mengobrol dengannya menggunakan bahasa Indonesia, namun karena turis tersebut belum pandai berbahasa Indonesia maka bicaranya pun berantakan dan tidak memperhatikan tata bahasa yang benar. Setelah itu, apa yang mungkin terjadi? Bisa jadi Anda kesulitan untuk menangkap maksud pembicaraan si turis tersebut meskipun turis tersebut berbicara dengan bahasa Indonesia.

Nah, sama seperti kasus diatas, XML saja tidak cukup agar *Web Service* dapat berkomunikasi dengan aplikasi lainnya. XML yang digunakan untuk saling bertukar informasi antara *Web Service* dengan aplikasi lainnya harus menggunakan sebuah format standar yang dapat dimengerti oleh keduanya. Format itulah yang dikenal dengan nama SOAP.

SOAP merupakan format standar dokumen berbentuk XML yang digunakan untuk melakukan proses *request* dan *response* antara *Web Service* dengan aplikasi yang memanggilnya. Dokumen SOAP yang digunakan untuk melakukan *request* disebut *SOAP request* sedangkan dokumen SOAP yang diperoleh dari *Web Service* disebut *SOAP response*.

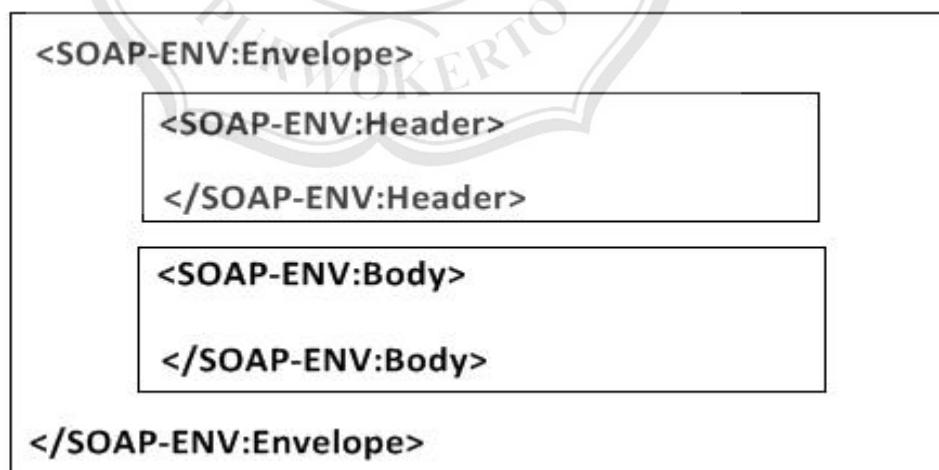
Jadi, jika XML adalah bahasa yang digunakan oleh *Web Service* dan aplikasi, maka SOAP adalah “tata bahasa” yang digunakan sehingga keduanya

bisa saling “memahami” saat sedang “berbicara”. Gambar 1 adalah proses request dan response menggunakan SOAP.



Gambar 1 Proses *Request* dan *Response* Menggunakan SOAP

Sebuah dokumen SOAP memiliki struktur standar. Struktur sebuah dokumen SOAP terdiri dari sebuah SOAP *Envelope*. Di dalam SOAP *Envelope* tersebut terdapat SOAP *Header* dan SOAP *Body*. Informasi yang dipertukarkan antara *Web Service* dan aplikasi akan diletakkan di dalam SOAP *Body*. Sedangkan isi SOAP *Header* adalah informasi-informasi yang sifatnya tambahan. Berikut ini adalah struktur dari sebuah dokumen SOAP (Lucky, 2008). Gambar 2 adalah skema dokumen SOAP (Lucky, 2008).



Gambar 2. Skema dokumen SOAP

3. *Web Service Description Language (WSDL).*

Sebelum mengakses sebuah *Web Service*, Anda tentu perlu mengetahui method-method apa saja yang disediakan oleh *Web Service* tersebut. Untuk mengetahuinya, Anda memerlukan sebuah dokumen yang bernama WSDL. WSDL adalah sebuah dokumen dalam format XML yang isinya menjelaskan informasi detail sebuah *Web Service*. Di dalam WSDL dijelaskan *method-method* apa saja yang tersedia dalam *Web Service*, parameter apa saja yang diperlukan untuk memanggil sebuah *method*, dan apa hasil atau tipe data yang dikembalikan oleh *method* yang dipanggil tersebut (Lucky, 2008).

4. *Universal Description, Discovery and Integration (UDDI).*

Ada satu lagi komponen pendukung *Web Service* yaitu UDDI. UDDI merupakan suatu *discovery service* yang digunakan untuk mendaftarkan dan mencari *Web Service*. Dengan menggunakan UDDI, Anda bisa mendaftarkan *Web Service* yang anda buat agar bisa dicari dan ditemukan oleh orang lain. Anda juga bisa mencari *Web Service* yang dibuat oleh orang lain berdasarkan kata kunci dan kategori tertentu (Lucky, 2008).

B. Pengembangan Penelitian Web Service Terdahulu

Sudah banyak *web service* yang telah dikembangkan dalam berbagai bidang pendidikan, perhubungan, teknologi, ekonomi dan bidang lainnya. Salah satu *web service* dalam bidang teknologi adalah pengembangan aplikasi portal tugas akhir di studio tugas akhir program studi s-1 ilmu komputer Fakultas MIPA Universitas Sumatra Utara (USU) Medan menggunakan *service oriented architecture* dengan *web service* (Alvin, 2010).

Android adalah sistem operasi bergerak (*mobile operating system*) yang sedang diminati khalayak umum. *Web service* merupakan aplikasi yang dapat diakses

dengan *android* salah satu implementasinya adalah *web service* di *system android* menggunakan *python* dan *sl4a (scripting language for android)*(setyadi, suadi dan mazharuddin, 2011).

Hikmawan (2011) memanfaatkan fitur sms yang sudah ada pada masing-masing telepon genggam untuk membuat suatu system SMS *Gateway* berbasis *web service* untuk mendukung proses penyebaran informasi antar anggota perusahaan dengan metode SMS *grouping*.

Secara umum aspek-aspek yang diperoleh dari penelitian di atas memberikan informasi yang diperlukan dalam pembuatan aplikasi implementasi *web service* kepada kendaraan bermotor.

C. SQL SERVER 2005

Semua DBMS (*Database Management Sistem*) modern saat ini menggunakan SQL (*Structured Query Language*) sebagai bahasa untuk memprogram *database*. SQL Server 2005 merupakan produk *Microsoft* adalah salah satu jenis *database* yang banyak digunakan di Indonesia. Produk ini mudah digunakan, mendukung aplikasi dengan arsitektur *client/server* (Radi, 2009).

1. Sekilas Tentang SQL Server 2005

SQL Server 2005 memperluas kinerja, keandalan, ketersediaan, programmabilitas dan mudah dalam penggunaannya. SQL Server 2005 meliputi beberapa fitur baru yang membuatnya menjadi suatu platform *database* berskala besar dan aplikasi *e-commerce*. SQL Server 2005 merupakan penyempurnaan dari SQL Server 2000 dan ditambah dengan beberapa fitur baru. Berikut ini fitur-fitur baru pada SQL Server 2005:

a. *Notification Services*

Notification Services adalah servis untuk mengirimkan dan menerima pemberitahuan (*notification*). *Notification Services* dapat mengirimkan pesan tepat waktu dari *database* kepada ribuan atau berjuta-juta para langganan di suatu perusahaan.

b. *Reporting Services*

Reporting Services adalah servis yang memberikan kesempatan dalam pembuatan laporan dari data SQL Server.

c. *Service Broker*

Service Broker adalah suatu teknologi terbaru pada *Microsoft SQL Server 2005* yang membantu *developer (programmer) database* untuk membangun keamanan dan dapat dipercaya. *Service Broker* menyediakan antrian dan pesan yang dapat dipercaya sebagai bagian dari *Database Engine*. Fitur ini menyediakan infrastruktur yang diperlukan untuk membangun aplikasi berkinerja tinggi.

2. *Arsitektur Client/Server*

SQL Server 2005 adalah salah satu jenis *database* relasional yang mendukung aplikasi dengan arsitektur *client/server (two tier)* yang mana sebagian proses dilakukan oleh *server* dan sebagian lagi dilakukan oleh aplikasi sehingga dapat mengurangi lalu lintas jaringan, karena SQL Server hanya memberikan data yang diperlukan oleh *client* saja (Radi, 2009).

D. MICROSOFT VISUAL STUDIO

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*,

aplikasi *Windows*, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, *Integrated Development Environment (IDE)*, dan dokumentasi (umumnya berupa *MSDN Library*). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain *Visual C++*, *Visual C#*, *Visual Basic*, *Visual Basic .NET*, *Visual InterDev*, *Visual J++*, *Visual J#*, *Visual FoxPro*, dan *Visual SourceSafe*.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas *Windows*) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas *.NET Framework*). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi *Silverlight*, aplikasi *Windows Mobile* (yang berjalan di atas *.NET Compact Framework*) (Wikipedia, 2012).

Dalam membangun aplikasi implementasi *Web service* ini menggunakan salah satu kompiler yang ada pada paket Visual Studio yaitu Visual C#. *Microsoft Visual C-Sharp* atau yang lebih dikenal dengan Visual C# adalah sebuah bahasa yang tidak diragukan lagi dalam proses pengembangan aplikasi berbasis *.NET Framework* dimana C# bebas dari masalah kompatibilitas dilengkapi dengan berbagai fitur yang sebagian besar merupakan fitur baru, menarik, dan tentu saja menjajikan.

Visual C# dibuat berdasarkan pemrograman C# yang merupakan bahasa pemrograman berorientasi objek dan mempunyai banyak kesamaan dengan C++, Java, dan VB. C# pada faktanya merupakan kombinasi antara efisiensi pemrograman C++, kesederhanaan pemrograman Java, dan penyederhanaan dari program Visual Basic.

Seperti pemrograman Java, C# juga tidak memperbolehkan *multipleinheritance* atau penggunaan *pointer* (pada *safe / managed code*), tetapi C# menyediakan *garbage memory collection* pada saat *runtime* dan pada saat pengecekan akses memori.

Meskipun bertentangan dengan pemrograman JAVA, C# tetap mempertahankan operasi unik yang terdapat pada bahasa pemrograman C++ seperti *overloading*, *enumerations*, *pre-processor directive*, *pointer* (pada *unmanaged/unsafe code*), dan fungsi *pointer*. Seperti halnya Visual Basic, bahasa pemrograman C# juga dilengkapi dengan *properties*.

Sebagai tambahan, bahasa pemrograman C# juga datang dengan beberapa fitur baru dan sangat menarik seperti *reflections*, *attributes*, *marshalling*, *remote*, *threads*, *streams data access* dengan ADO.NET, dan masih banyak lagi (Wahana, 2011).

E. **ANDROID**

Android adalah system operasi bergerak (*mobile operating system*) yang mengadopsi sistem operasi Linux, namun telah dimodifikasi. Android diambil alih oleh Google pada tahun 2005 dari Android, Inc sebagai bagian dari strategi untuk mengisi pasar sistem operasi bergerak. Google mengambil alih seluruh hasil kerja Android termasuk tim yang mengembangkan Android.

Google menginginkan agar Android bersifat terbuka dan gratis, oleh karena itu hampir setiap kode program Android diluncurkan berdasarkan lisensi *open-source* Apache yang berarti bahwa semua orang yang ingin menggunakan Android dapat *men-download* penuh *source code*-nya.

Di samping itu produsen perangkat keras juga dapat menambahkan *extension*-nya sendiri ke dalam Android sesuai kebutuhan produk mereka. Model pengembangannya yang sederhana membuat Android menarik bagi *vendor-vendor* perangkat keras (contoh: Samsung).

Keuntungan utama dari Android adalah adanya pendekatan aplikasi secara terpadu. Pengembangan hanya berkonsentrasi pada aplikasi saja, aplikasi tersebut

bisa berjalan pada beberapa perangkat yang berbeda selama masih ditenagai oleh Android (pengembang tidak perlu mempertimbangkan kebutuhan jenis perangkatnya) (Suprianto, 2012).

F. INTERNET INFORMATION SERVICES (IIS)

IIS atau *Internet Information Services* adalah sebuah HTTP *web server* seperti Apache yang digunakan dalam sistem operasi *server Windows*, mulai dari *Windows NT 4.0 Server*, *Windows 2000 Server* atau *Windows Server 2003*. Layanan ini merupakan layanan terintegrasi dalam *Windows 2000 Server*, *Windows Server 2003* atau sebagai *add-on* dalam *Windows NT 4.0*. Layanan ini berfungsi sebagai pendukung protokol TCP/IP yang berjalan dalam lapisan aplikasi (*application layer*). IIS juga menjadi *fondasi* dari *platform Internet* dan *Intranet Microsoft*, yang mencakup *Microsoft Site Server*, *Microsoft Commercial Internet System* dan produk-produk *Microsoft Back Office* lainnya.

IIS telah berevolusi semenjak diperkenalkan pertama kali pada *Windows NT 3.51* (meski kurang banyak digunakan) hingga IIS versi 6.0 yang terdapat dalam *Windows Server 2003*. Versi 5.0 diintegrasikan dalam *Windows 2000*, sedangkan *Windows XP Professional* memiliki IIS versi 5.1. *Windows NT 4.0* memiliki versi 4.01 yang termasuk ke dalam *add-on Windows NT Option Pack*. Dalam *Windows NT 4.0 Workstation* atau *Windows 95/98*, IIS juga dapat diinstalasikan sebagai *Microsoft Personal Web Server (PWS)*(Iqbal, 2011).

1. Versi

IIS tersedia dalam beberapa versi dan sistem operasi sebagai berikut:

- a. IIS 1.0 untuk *Windows NT 3.51*, yang tersedia sebagai tambahan yang gratis.
- b. IIS 2.0 untuk *Windows NT 4.0*
- c. IIS 3.0 untuk *Windows NT 4.0 Service Pack 3*
- d. IIS 4.0 untuk *Windows NT 4.0 Option Pack*

- e. IIS 5.0 untuk Windows 2000 (Professional dan Server)
 - f. IIS 5.1 untuk Windows XP Professional dan Windows XP Media Center Edition
 - g. IIS 6.0 untuk Windows Server 2003 dan Windows XP Professional x64 Edition
 - h. IIS 7.0 untuk Windows Server 2008 dan Windows Vista (Edisi Business, Edisi Enterprise, dan Ultimate)
 - i. IIS 7.5 untuk Windows Server 2008 R2 dan Windows 7
2. Fitur
- a. Terintegrasi dengan Windows NT secara penuh (sistem keamanan, auditing, dan izin akses NTFS)
 - b. Mendukung penuh protokol HTTP versi 1.1
 - c. Sudah mencakup protokol FTP
 - d. Dukungan terbatas untuk protokol SMTP
 - e. Dukungan untuk protokol NNTP
 - f. Dukungan untuk protokol keamanan SSL
 - g. Dapat digunakan sebagai platform di mana aplikasi web berjalan, yakni dengan menggunakan *Active Server Pages (ASP)*, *ASP.NET*, *Internet Server API (ISAPI)*, *Common Gateway Interface (CGI)*, *Microsoft .NET Framework*, *Microsoft Visual Basic Scripting (VBScript)*, *JScript*, dan beberapa bahasa skrip yang dapat diinstalasikan seperti Perl atau PHP.
 - h. Mengizinkan aplikasi web untuk dijalankan sebagai proses yang terisolasi dalam ruangan memori yang terpisah untuk mencegah satu aplikasi membuat *crash* aplikasi lainnya.
 - i. Dapat diatur dengan beberapa cara: *Microsoft Management Console*, *via web browser*, atau menggunakan skrip *Windows Scripting Host*.

- j. *Bandwidth throttling* yang dapat mencegah sebuah situs web memonopoli *bandwidth* yang tersedia.

G. ECLIPSE

Eclipse *foundation* membagi pekerjaannya ke dalam proyek dan sub-subproyek. Proyek yang mungkin sering terdengar adalah proyek Eclipse, proyek Eclipse *Tools*, serta proyek Eclipse *Tecno*logy. Eclipse project adalah kerangka utama Eclipse Foundation. Proyek Eclipse ini memiliki 3 subproyek di dalamnya, subproyek *Platform*, subproyek *Java Development Tools*, serta subproyek *Plug-In Development*. Subproyek *Platform* mengerjakan segala sesuatu tentang Eclipse, seperti fasilitas penyuntingan teks (*editing*), pencarian (*searching*), pembuatan halaman-halaman bantuan (*help pages*), pencarian kesalahan (*debugging*), dan pembentukan versi (*versioning*). Pada bagian paling inti subproyek *Platform* ini ada yang dinamakan *core platform* yang intinya adalah kode-kode untuk memulai dan menjalankan Eclipse, penciptaan dan pengelolaan *plug-in* serta pengelolaan sumber daya program yang lain. Sebagai tambahan, subproyek *platform* mendefinisikan tampilan utama antarmuka pengguna (*user interface*) seperti yang dimiliki Eclipse saat ini, dimana antarmuka pengguna ini berbasis 2 teknologi, yaitu teknologi kontroversial SWT (*Standard Widget Toolkit*) dan teknologi non-kontroversial *JFace*. SWT merupakan himpunan kelas dan metode antarmuka grafis dasar, di antaranya tombol, menu, label, dan event, sementara *JFace* merupakan kaskas antarmuka grafis yang peringkatnya lebih tinggi, seperti *wizzard*, *viewer*, serta pemformatan teks. *JFace* merupakan perkembangan lebih lanjut dari kaskas dasar SWT.

Banyak orang menganggap bahwa Eclipse tidak lebih dari lingkungan pengembangan terintegrasi (*IDE-Integrated Development Environment*) untuk Java. Jika kita menjalankan Eclipse maka kita akan melihat beberapa *view* (sudut pandang) Java di dalamnya, yaitu proyek Java, kakas pencarian Java, serta sejumlah hal yang bersifat spesifik Java. Tetapi sesungguhnya Java hanya merupakan suatu bagian dari Eclipse. Eclipse lebih besar dari sekedar IDE Java. Pada kenyataannya Eclipse merupakan bahasa yang bersifat mandiri terhadap *platform* yang di dalamnya memuat lingkungan terintegrasi untuk pengembangan aplikasi Java. Lingkungan pengembangan Java merupakan subproyek yang terpisah, yang dinamakan subproyek JDT (*Java Development Tools*). Subproyek ini memuat kompailer Java, alat penyunting Java yang canggih, pemeriksaan kesalahan (*debugging*) yang terintegrasi, dan sebagainya (Nugroho, 2007).

