

Web Service

Asep Herman Suyanto
info@bambutechno.com
<http://www.bambutechno.com>

Web service adalah sistem perangkat lunak yang dirancang untuk mendukung interaksi yang bisa beroperasi *machine-to-machine* di atas jaringan. *Web service* mempunyai alat penghubung yang diuraikan di dalam format *machine-processable* (secara spesifik WSDL). Sistem lain saling berhubungan dengan *Web service* di dalam cara yang ditentukan oleh deskripsinya yang menggunakan pesan SOAP, secara khas disampaikan menggunakan HTTP dengan *XML serialization*, bersama dengan standar lain yang terkait dengan *web* (Booth et al., 2004).

Untuk menjalankan fungsinya, *web service* memerlukan agen. Agen adalah bagian perangkat lunak atau perangkat keras yang mengirimkan dan menerima pesan. Agen dapat ditulis dengan berbagai bahasa pemrograman. Dan dapat berganti-ganti bahasa pemrograman dengan fungsi yang sama.

Tujuan *web service* adalah untuk menyediakan beberapa fungsi atas nama pemilik nya – seseorang atau organisasi seperti bisnis atau perorangan. *Provider entity* adalah organisasi atau orang yang menyediakan agen yang sesuai untuk menerapkan *service* tertentu. *Requester entity* adalah seseorang atau organisasi yang berkeinginan untuk menggunakan *web service* *provider entity*. Itu akan menggunakan *requester agent* untuk menukar pesan dengan *provider agent* milik *provider entity*. Dalam pertukaran pesan agar berhasil, *requester entity* dan *provider entity* harus dulu sepakat menggunakan semantik dan mekanisme yang sama dalam pertukaran pesan. Semantik dalam *web service* adalah ekspektasi tentang perilaku (*behavior*) *service*. Ini semacam kontrak antara *requester entity* dan *provider entity* mengenai kegunaan dan konsekuensi dari interaksi tersebut.

Mekanisme pertukaran pesan didokumentasikan di *Web Service Description (WSD)*. WSD adalah spesifikasi mesin yang dapat berproses pada *interface web service*. Itu menggambarkan format pesan, tipe data, protokol transpor, dan format serialisasi yang digunakan antara *requester agent* dan *provider agent*. Itu juga menentukan satu atau lebih jaringan tempat di mana *provider agent* dapat dipanggil.

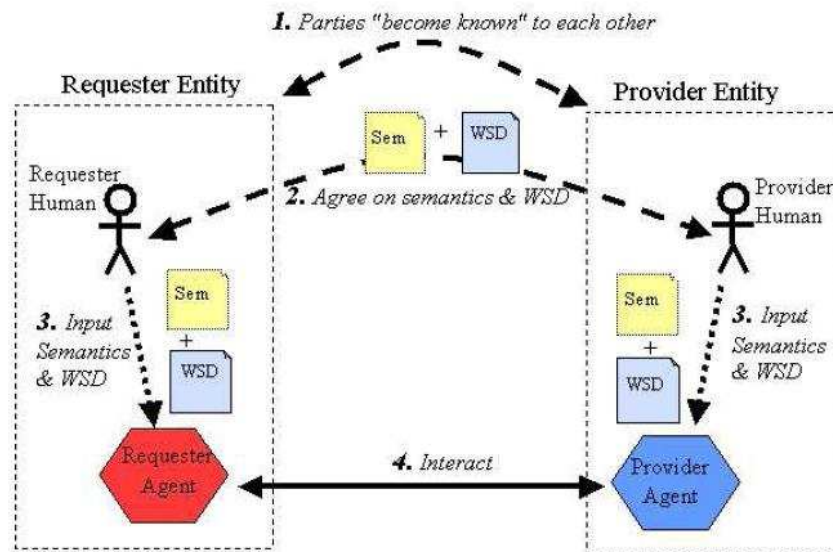
Kemungkinan banyak *requester entity* terlibat dan menggunakan *web service*. Gambar 2.6 menggambarkan secara riil dari langkah-langkah. Secara umum, penjelasan langkah – langkahnya sebagai berikut:

1. *Requester entity* dan *provider entity* saling mengetahui satu sama lain. Atau, setidaknya salah satu mengetahui yang lain.
2. *Requester entity* dan *provider entity* sama-sama setuju deskripsi layanan dan semantik yang akan mengatur interaksi antara *requester agent* dan *provider agent*.
3. Deskripsi layanan dan semantik direalisasikan/didapatkan oleh *requester agent* dan *provider agent*.
4. *Requester agent* dan *provider agent* bertukar pesan.

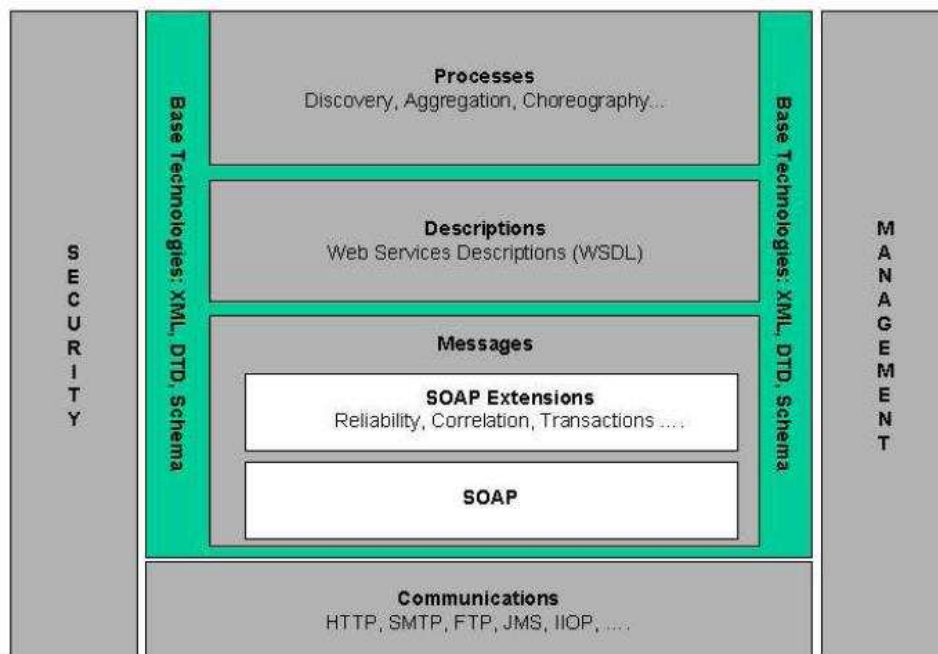
Arsitektur *web service* melibatkan banyak teknologi saling berhubungan dan di-layer. Ada banyak cara untuk visualisasi teknologi ini, sama halnya ada banyak cara untuk membangun dan menggunakan *web service*. Gambar 2.7 mengilustrasikan sebagian dari teknologi ini.

1. XML (Extensible Markup Language)

Extensible Markup Language (XML) adalah bagian dari SGML. Tujuannya untuk memungkinkan SGML umum untuk dilayani, diterima, dan diproses pada *Web* yang sekarang dengan HTML. XML telah dirancang untuk meringankan implementasi dan untuk interoperabilas dengan SGML dan HTML.



Gambar 2.6 Proses Umum dalam Melibatkan Web Service



Gambar 2.7 Tumpukan Arsitektur Web Service

XML dokumen terdiri dari unit penyimpanan disebut *entities*, yang berisi data yang tidak diuraikan atau diuraikan. Data yang diuraikan terdiri dari karakter, sebagian yang membentuk data karakter, dan sebagian dari yang membentuk *markup*. *Markup encode* deskripsi *layout* penyimpanan dan struktur logika dokumen. XML menyediakan mekanisme untuk memaksakan batasan pada *layout* penyimpanan dan struktur logika.

XML memecahkan kebutuhan teknologi di dalam banyak tempat. Dengan menawarkan standar, yang fleksibel dan format data yang dapat diperluas dengan sifatnya, XML secara signifikan mengurangi beban penyebaran banyak teknologi yang diperlukan untuk memastikan sukses *web service*. (Bray et.al., 2006).

2. XSD (XML Schema Definition Language)

XML Schema terdiri dari komponen seperti definisi jenis dan deklarasi elemen. Ini dapat digunakan untuk menilai kebenaran dari elemen *well-formed* dan item informasi atribut dan lagi pula boleh menetapkan tambahan untuk item dan turunan. Tambahan ini membuat informasi eksplisit yang mungkin telah tersembunyi dalam dokumen original, seperti dinormalisir dan/atau nilai *default* untuk atribut, elemen dan jenis dari elemen dan atribut item informasi. (Thompson et.al., 2004).

3. WSDL (Web Services Description Language)

WSDL adalah sebagai protokol komunikasi dan format pesan distandardisasi di komunitas *web*, WSDL mungkin terus meningkat dan penting untuk bisa menguraikan komunikasi dalam beberapa cara terstruktur. WSDL menunjukkan kebutuhan ini dengan penjelasan *XML grammar* untuk menguraikan *service* jaringan seperti koleksi komunikasi *endpoints* untuk mampu bertukar pesan. Definisi *service* WSDL menyediakan dokumentasi untuk sistem terdistribusi dan bertindak sebagai petunjuk untuk mengotomatiskan detil yang dilibatkan di dalam komunikasi aplikasi (Christensen et.al., 2001).

WSDL adalah tatabahasa XML untuk penetapan *public interface* untuk *web service* (Cerami, 2002). *public interface* ini dapat meliputi:

1. Informasi pada semua fungsi yang tersedia secara publik.
2. Tipe Data informasi untuk semua pesan XML.
3. Membungkus informasi tentang *specific transport protocol* untuk digunakan.
4. Menunjuk informasi untuk menempatkan *service* yang ditetapkan.

4. SOAP (Simple Object Access Protocol)

SOAP adalah protokol untuk pertukaran informasi yang tersusun di dalam lingkungan didesentralisasi, terdistribusi. Menggunakan teknologi XML untuk mendefinisikan kerangka pesan yang dapat dikembangkan yang menyediakan pembentukan pesan yang dapat ditukar di atas berbagai dasar protokol. Kerangka telah dirancang untuk tidak terikat pada *model programming* tertentu dan semantik spesifik implementasi lain (Gudgin et.al., 2003).

Walaupun SOAP dapat digunakan dalam berbagai sistem yang *messaging* dan dapat dikirimkan via berbagai protokol pengangkutan, fokus SOAP yang utama adalah *Remote Procedure Calls (RPC)* yang dijalankan via HTTP. Seperti XML-RPC, SOAP adalah *platform* mandiri, dan oleh karena itu memungkinkan aplikasi berbeda untuk berkomunikasi dengan satu sama lain.

2.3.5 REST (Representational State Transfer)

Representational State Transfer (REST) adalah gaya arsitektur perangkat lunak untuk sistem hypermedia terdistribusi seperti *world wide web*. Merupakan salah satu alternatif dari SOAP. *Representational State Transfer* dimaksudkan untuk menimbulkan gambaran bagaimana aplikasi *web* dirancang dengan baik yang menunjukkan reaksi: jaringan dari halaman *web* (keadaan mesin sebetulnya), di mana *user* maju sampai aplikasi dengan pemilihan *link* (keadaan transisi), menghasilkan halaman berikut (mewakili status berikutnya dari aplikasi) yang sedang ditransfer ke *user* dan menyumbangkan untuk penggunaan mereka.

REST sering digunakan dalam pengertian lepas untuk menguraikan penghubung sederhana yang mengirimkan data spesifik domain di atas HTTP tanpa lapisan *messaging* tambahan seperti SOAP atau *session tracking* via *HTTP cookies*. Pemisahan *Client-Server REST* menyangkut menyederhanakan implementasi komponen, mengurangi kompleksitas *connector* semantik, meningkatkan efektivitas perbaikan *performance*, dan meningkatkan

scalability dari komponen server asli. Dilapisan batasan sistem mengizinkan perantara – *proxies*, *gateways*, dan *firewalls* – untuk diperkenalkan pada berbagai *point* di dalam komunikasi tanpa mengubah alat penghubung antara komponen, dengan begitu membiarkannya untuk membantu dalam menterjemahan komunikasi atau meningkatkan *performance* via besaran skala, membagi *cache*.

REST memungkinkan perantara memproses dengan menghambat pesan untuk deskriptif sendiri: interaksi tanpa status antara permintaan, metoda standar dan jenis media digunakan untuk menandai adanya semantik dan informasi pertukaran, dan tanggapan menandai adanya *cacheability*. (Fielding, 2000).