

## MINGGU KE II

### PENGENALAN WEB SERVICE

#### **Deskripsi**

Materi membahas contoh-contoh implementasi web service yang sudah digunakan oleh masyarakat. Untuk mengenal lebih jauh tentang layanan web service eksplorasi pembahasan tentang jenis web service dan manfaatnya, teknologi web service, arsitektur web servis, operasi web service, keuntungan web service.

#### **Tujuan Pembelajaran**

Dengan mengikuti pembelajaran ini, mahasiswa mampu:

1. Mengetahui berbagai layanan web service ditengah-tengah masyarakat
2. Memahami layanan webservice, mampu melatih pembuatan layanan web service
3. Melatih atau mencoba membuat layanan web service XML, SOAP, WSDL

#### **A. Pengenalan Web Service**

Selama ini mungkin Anda pernah atau bahkan sering mendengar mengenai Web Service. Sebenarnya apakah Web Service itu ? Samakah Web Service dengan Website ?

Web Service ternyata sangat berbeda dengan website. Perbedaan yang paling terlihat adalah Website dibuat untuk memiliki tampilan atau user interface yang bagus sedangkan Web Service tidak memiliki tampilan. Mengapa Web Service tidak memiliki tampilan (user interface) ?

Web Service adalah Sebuah aplikasi lintas platform yang dapat diakses melalui jaringan (intranet dan internet ) dimana dalam aplikasi tersebut menyediakan berbagai fungsi dengan tujuan digunakan untuk interaksi aplikasi satu dengan aplikasi yang lain. Web service dapat diartikan juga sebuah metode pertukaran data, tanpa memperhatikan dimana sebuah database ditanamkan, dibuat dalam bahasa apa, sebuah aplikasi yang mengkonsumsi data, dan di platform apa sebuah data itu dikonsumsi. Web service mampu menunjang interoperabilitas. Sehingga web service mampu menjadi sebuah jembatan penghubung antara berbagai sistem yang ada.

Web pada umumnya digunakan untuk melakukan respon dan request yang dilakukan antara client dan server.

**Sebagai Contoh :**

Implementasi Web Service adalah sistem login seperti yang ada di **Kaskus**, atau **Detik.com**.

Jika Anda mengunjungi **Kaskus**, maka Anda akan menjumpai **sistem login** yang dapat menggunakan **akun Facebook, Yahoo** maupun **Twitter**.

Dengan kata lain, Anda dapat bergabung dalam komunitas forum Kaskus atau istilah kerennya Kaskuser hanya dengan memiliki akun FB, Yahoo atau Twitter tanpa harus registrasi di dalam Kaskus nya sendiri.

Contoh lain implementasi Web Service adalah di bagian komentar Detik.com. Setiap kali kita akan mengisi komentar di Detik.com, kita bisa menggunakan akun FB kita.

Nah... yang menjadi pertanyaan adalah, kok bisa ya kita masuk ke dalam sistem Kaskus maupun Detik.com menggunakan akun lain seperti FB, Yahoo maupun Twitter? Ya... ini karena FB, Yahoo dan Twitter menyediakan service yang memungkinkan sistem lain menggunakan akun mereka untuk login. Dan... kebetulan Kaskus dan Detik.com ini memanfaatkan service tersebut. Dalam kasus ini, FB, Yahoo, dan Twitter dikatakan bertindak sebagai server sedangkan Kaskus dan Detik.com bertindak sebagai client. Lihat gambar di bawah ini.



**Contoh lain**, buat Anda yang memang sudah memiliki perangkat ponsel Android, mungkin sudah pernah menginstall aplikasi semacam detik.com, Kompas.com atau VivaNews.com, dimana Anda bisa melihat berita dari situs detik.com di ponsel Android Anda.

**Pertanyaannya** ? Apakah portal-portal tersebut dalam membuat aplikasi portal versi Android juga membuat database/data beritanya? Apakah mereka melakukan pengimputan data berita dua kali ? yaitu versi web dan versi android ?

**Jawabannya** sudah pasti tidak, karena mereka pasti telah menggunakan layanan web service untuk pertukaran data pada dua versi aplikasi yang mereka buat.



Jadi apakah Web Service itu ? secara singkat **Web Service** adalah aplikasi yang dibuat agar dapat dipanggil atau diakses oleh aplikasi lain melalui internet dengan menggunakan format pertukaran data sebagai format pengiriman pesan. Adapun yang biasa dipakai adalah format pertukaran data dengan **XML** dan **JSON**.

Lalu, bagaimana caranya memanggil atau memanfaatkan sebuah Web Service ? sebuah web service dapat dipanggil oleh aplikasi lain dengan menggunakan bantuan HTTP (Hypertext Transfer Protocol). Web service juga memungkinkan untuk dipanggil dengan menggunakan protocol lain seperti SMTP (Simple Mail Transfer Protocol), namun yang paling umum digunakan HTTP.

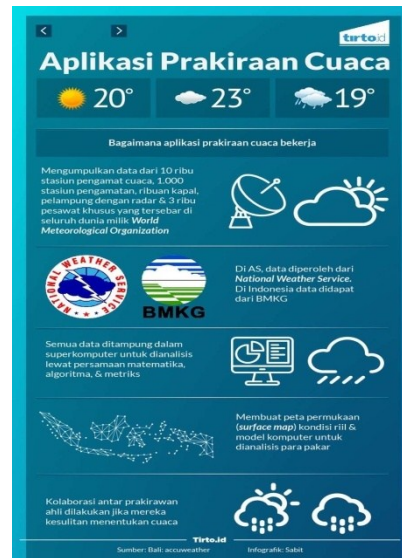
Karena web service menggunakan protokol **HTTP**, tentu **PHP** sebagai bahasa pemrograman web menjadi salah satu kekuatan dalam bahasa pemrograman yang mengelola web service. Meski banyak teknik dan metode untuk menghasilkan web service dengan PHP.

## B. Jenis-Jenis Web Service Berdasarkan Manfaat

- **Reusable application-components**

Jenis web service yang dimanfaatkan secara periodik atau berulang-ulang.

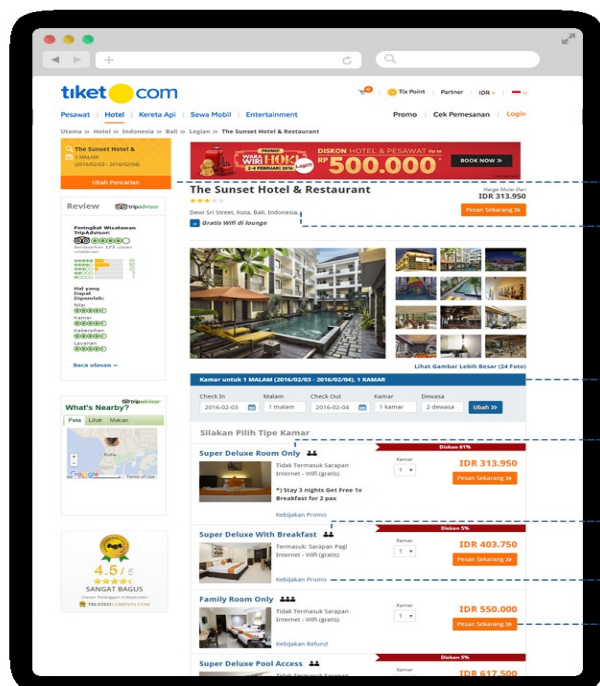
**Contoh :** layanan konversi mata uang, info cuaca, penterjemahan bahasa dll, seperti contoh gambar dibawah ini



- **Connect existing software**

web service dapat dimanfaatkan untuk mengintegrasikan antar berbagai aplikasi yang berbeda.

**Contoh:** aplikasi traveling diintegrasikan dengan aplikasi pemesanan hotel, wisata, rental kendaraan dll meskipun dengan platform, lokasi dan pemrograman yang berbeda. seperti contoh gambar dibawah ini



### C. Teknologi Pendukung Web Service

Dalam pengoperasiannya, Web Service menerapkan empat komponen / teknologi yang mendukung kinerjanya. 5 (Lima) komponen tersebut adalah :

- a) XML (*Extensible Markup Language*) merupakan standar untuk markup dokumen yang disahkan oleh W3C (World Wide Web Consortium). XML mendefinisikan sintaks yang umum digunakan untuk markup data secara sederhana. XML menyediakan format standar untuk dokumen komputer. Format ini cukup fleksibel untuk disesuaikan dengan domain yang beragam seperti website, pertukaran data elektronik, grafik vektor, hierarki, serialisasi objek, Remote Procedure Call (RPC), dan sistem voice mail (Harold dan Means, 2002). XML merupakan bagian penting bagi pengembang yang ingin membangun web service. XML berbasis teks, sehingga bersifat platform independent. Sifat ini membuat pertukaran data antar platform menjadi mungkin.

#### Contoh Dokumen XML :

```
<?xml version="1.0" encoding="utf-8" ?>
<karyawan instansi="UBSI">
  <nama>Rachmat Hidayat, M.Kom</nama>
  <nik> </nik>
  <jabatan>Staf Akademi</jabatan>
  <status>Tetap</status>
  <alamat>Tangerang</alamat>
</karyawan>
```

- b) SOAP (*Simple Object Access Protocol*) adalah sebuah protokol berbasis XML dengan komunikasi bergaya RPC (*Remote Procedure Call*) yang menyediakan cara standar untuk memaketkan pesan (Chappell dan Jewell, 2002). SOAP mendefinisikan cara dalam pembungkusan (encapsulation) dan pertukaran (exchange) pesan.

Dokumen XML yang terdiri dari komponen:

1. **SOAP Envelope**, mengidentifikasi dokumen XML sebagai SOAP message
2. **Elemen Header** (opsional)
3. **Elemen Body**, berisi informasi pemanggilan dan responsnya

4. **Elemen Fault** (opsional), berisi error yang terjadi saat pemanggilan diproses

**Contoh Dokumen SOAP Message Dokumen XML yang terdiri dari komponen:**

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap=" envelope" soap:encodingStyle="
encoding">
<soap:Header>....</soap:Header>
<soap:Body>
<soap:Fault>.....</soap:Fault>
</soap:Body>
</soap:Envelope>
```

- c) *Web Service Description Language* (WSDL) adalah sebuah dokumen dalam format XML yang isinya menjelaskan informasi detail sebuah Web Service. Di dalam WSDL dijelaskan method-method yang tersedia dalam web service, parameter apa saja yang diperlukan untuk memanggil sebuah method, dan hasil atau tipe data yang dikembalikan oleh method yang dipanggil.

## FORMAT WSDL

### Elemen <portType>

Mendefinisikan nama web service, operasi- operasi yang dilakukan, dan message yang terlibat

### Elemen <message>

Mendefinisikan message yang digunakan oleh web service

### Elemen <binding>

Mendefinisikan protokol komunikasi yang digunakan

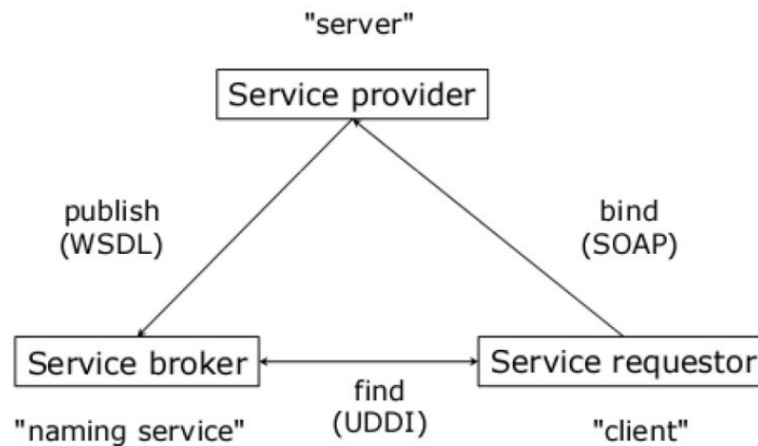
### Elemen <type>

Mendefinisikan tipe-tipe data yang

```
<definitions>
<types> ... </types>
<message> ... </message>
<portType> ... </portType>
<binding> ... </binding>
</definitions>
```



merupakan tempat dimana Service provider mempublikasikan layanannya. Pada arsitektur Web Service, Service registry bersifat optional. Teknologi web service memungkinkan kita dapat menghubungkan berbagai jenis software yang memiliki platform dan sistem operasi yang berbeda.



### Bagaimana Web Service Beroperasi?

Sisi Server:	Sisi Client:
<ul style="list-style-type: none"><li>✓ Membuat fungsi utama/core function</li><li>✓ Membuat service wrapper berupa XML-RPC atau SOAP</li><li>✓ Membuat deskripsi service berupa WSDL atau instruksi integrasi XML-RPC (memuat semua method public, argumen dan return valuenya); plus dokumentasi yang human readable</li><li>✓ Deploy (rilis) service</li><li>✓ Daftarkan service tersebut melalui UDDI agar discoverable</li></ul>	<ul style="list-style-type: none"><li>✓ Mencari service melalui UDDI</li><li>✓ Mengambil service description file berupa WSDL atau instruksi XML-RPC</li><li>✓ Membuat klien XML-RPC atau SOAP (dapat berupa fungsi lokal atau pesan XML untuk dikirim → berdasarkan WSDLnya)</li><li>✓ Memanggil remote service tersebut</li></ul>

### E. Operasi-Operasi Web Service

Secara umum, web service memiliki 3(tiga) operasi yang terlibat di dalamnya, yaitu:

1. **Publish/Unpublish:** Menerbitkan/menghapus layanan ke dalam atau dari registry.
2. **Find:** Service requestor mencari dan menemukan layanan yang dibutuhkan.



3. **Bind:** Service requestor setelah menemukan layanan yang dicarinya, kemudian melakukan binding ke service provider untuk melakukan interaksi dan mengakses layanan/service yang disediakan oleh service provider.

#### **F. Keuntungan Menggunakan Web Service**

- a. *Web Service* menyediakan interoperabilitas antar berbagai aplikasi perangkat lunak yang berjalan pada platform yang berbeda.
- b. *Web Service* menggunakan standard dan protocol yang bersifat terbuka.
- c. *Web service* mengijinkan penggunaan kembali service dan komponen didalam suatu infrastruktur.
- d. *Web service* dapat secara bebas digunakan (loosely coupled) dengan demikian memudahkan suatu pendekatan terdistribusi ke peintegrasian aplikasi

#### **G. Keuntungan Menggunakan Web Service**

- a. Karakteristik standard Web Service saat ini masih dalam tahap perkembangan awal dibandingkan open standard komputer terdistribusi yang lebih matang seperti CORBA. Ini nampaknya akan merupakan suatu kerugian yang temporer ketika kebanyakan vendor sudah merasa terikat dengan standard OASIS untuk menerapkan Mutu dari aspek service dari produk mereka.
- b. Web Service dapat saja memiliki performance/kinerja yang lemah dibandingkan dengan pendekatan komputasi terdistribusi lain seperti RMI, CORBA, atau DCOM. Ini merupakan suatu trade-off yang umum ketika memilih format yang text-based. XML dengan tegas tidak menghitung antar tujuan disain-nya baik singkatan dari penyandian maupun efisiensi dari uraian. Ini bisa berubah dengan standard XML Infoset, yang menguraikan bahasa yang XML-based dalam kaitan dengan hal-hal yang abstrak (unsur-unsur, atribut, logika bersarang). Penyajian angle-bracket (< >) secara tradisional kini dilihat sebagai suatu serialisasi ASCII (atau Unicode) dari XML, bukan XML itu sendiri. Pada model ini, serialisasi biner adalah suatu alternatif yang sama yang sah. Penyajian biner seperti SOAP MTOM menjanjikan untuk meningkatkan efisiensi wire dari XML messaging.