

IMPLEMENTASI WEB-SERVICE UNTUK PENGEMBANGAN SISTEM LAYANAN PARIWISATA TERPADU

Tri Kuntoro Priyambodo

Fakultas MIPA - Universitas Gadjah Mada
E-mail: mastri@ugm.ac.id; mastri@jogja.go.id

Intisari

Telah dibuat sebuah sistem informasi kepariwisataan terpadu yang dikembangkan dengan menggunakan teknologi web-service. Untuk keperluan tersebut diuraikan contoh dua buah web-service yaitu web-service WisataBudayaJawa yang berfungsi untuk memberikan rekomendasi obyek-obyek wisata di propinsi Jawa Tengah dan Daerah Istimewa Yogyakarta, dan web-service Kurs yang berfungsi untuk memberikan informasi harian mengenai nilai tukar sejumlah mata uang asing terhadap mata uang rupiah. Kedua web-service tersebut saling diintegrasikan untuk memberikan gambaran mengenai sistem multi-tier serta dimanfaatkan untuk membangun sebuah program aplikasi web.

Dari pengembangan sistem, diperoleh hasil yang menunjukkan bahwa web-service sangat efektif untuk digunakan dalam pengembangan aplikasi karena programmer hanya memberikan input ke web-service dan mengambil hasil pengolahan data atau output dari web-service. Sedangkan dari percobaan penggunaan web-service pada komputer dengan sistem operasi Linux menunjukkan bahwa web-service dapat digunakan pada komputer dengan sistem operasi yang berbeda dengan sistem operasi dimana web-service tersebut dikembangkan.

Kata kunci: Web-service, pemrograman web, jaringan, sistem terdistribusi

1. Latar Belakang Masalah

Saat ini, sering terdengar istilah aplikasi berbasis web (*web-based application*). Sebagian besar orang berpikir bahwa *website* atau *homepage* yang dibuat dan dimiliki oleh suatu pihak itulah yang dimaksud dengan aplikasi berbasis web.

Aplikasi berbasis web sendiri sebenarnya tidak mutlak menggunakan *web browser* sebagai tampilan *front-end*nya. Menurut Gani (2001) pengertian web di sini adalah suatu jaringan luas atau keterhubungan antara beberapa aplikasi dan/atau komponen suatu aplikasi menjadi suatu aplikasi baru. Memang, aplikasi berbasis web yang sering kita jumpai saat ini masih menggunakan web browser seperti Microsoft Internet Explorer (IE) dan Netscape Navigator sebagai media untuk menampilkan gambar-gambar maupun tulisan yang membentuk aplikasi *front end*, namun dengan teknologi yang berkembang sekarang memungkinkan sebuah aplikasi untuk saling terhubung ke suatu database atau aplikasi lain secara *remote* menggunakan jaringan internet tanpa menggunakan program web browser sama sekali.

Saat ini telah berkembang sebuah teknologi yang memungkinkan bentuk akhir dari sebuah program atau aplikasi komputer adalah berupa sebuah *service* atau fungsi yang melakukan sebuah tugas atau proses yang spesifik dan dikenal dengan istilah *web-service*. Konsep ini sebenarnya bukanlah sebuah konsep yang baru, berbagai teknologi telah dikembangkan untuk mewujudkan kebutuhan akan koneksi atau hubungan antar aplikasi, seperti teknologi DCOM dan CORBA. Namun teknologi tersebut dianggap masih memiliki

keterbatasan, yaitu: adanya kesulitan untuk dilakukan silang teknologi antara sistem operasi yang satu dengan sistem operasi yang lain, maupun antara satu bahasa pemrograman dengan bahasa pemrograman yang lain (Manes, 2001).

Bertolak dari adanya teknologi web-service yang memungkinkan perpaduan fungsi-fungsi dalam membangun sebuah program aplikasi tanpa bergantung lagi pada sistem operasi maupun bahasa pemrograman yang digunakan, maka diambil topik pembahasan mengenai web-service dan mengimplementasikannya untuk membangun *service* yang akan memberikan rekomendasi mengenai sejumlah obyek wisata di Jawa Tengah dan Yogyakarta yang sebaiknya dikunjungi oleh wisatawan dalam rangka wisata budaya Jawa. *Service* tersebut dikembangkan dengan tujuan untuk melihat bagaimana web-service dapat mengakomodasi atau dimanfaatkan dalam membangun sebuah program aplikasi, memperlihatkan integrasi antar web-service serta untuk mengetahui apakah web-service dapat digunakan pada komputer dengan sistem operasi yang berbeda dengan sistem operasi *service provider*.

Metodologi yang digunakan dalam menyelesaikan penelitian ini adalah dokumentasi data, studi pustaka, desain sistem dan programming untuk mengimplementasikan sistem yang telah dirancang.

2. Web-service

Dalam Microsoft (2000) dinyatakan bahwa web-service merupakan tahapan ketiga dari tahapan

evolusi ASP (Application Service Provider) dimana pada tahapan pertama ditekankan pada penyediaan aplikasi desktop sedangkan pada tahapan kedua ditekankan pada penyediaan aplikasi berbasis *client-server*. Pada tahapan ketiga ini, komponen-komponen atau *building blocks* software disediakan sebagai service dan disebarkan lewat jaringan internet untuk diintegrasikan dengan aplikasi-aplikasi lain.

Menurut Kreger (2001) web-service diartikan sebagai sebuah antar muka (*interface*) yang menggambarkan sekumpulan operasi-operasi yang dapat diakses melalui jaringan, misalnya internet, dalam bentuk pesan XML. Sedangkan menurut Manes (2001), 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.

Web-service dapat dibangun dengan menggunakan bahasa pemrograman apa saja dan juga dapat diimplementasikan pada platform manapun. Kita dapat membangun web-service pada Windows 2000 dan menjalankannya melalui Windows, Linux, Unix, Mac, PalmOS dan WinCE (Hamids, 2000). Hal ini dimungkinkan karena web-service berkomunikasi menggunakan sebuah standar format data yang universal yaitu XML dan menggunakan protokol SOAP. Karena web-service menggunakan format data XML, maka web-service juga mewariskan sifat *multi-tier* dari XML sehingga memungkinkan terjadinya integrasi antar web-service atau aplikasi (Microsoft, 2001). Menurut Meiyanto (2001) pada sistem *multi-tier*, aplikasi maupun dokumen XML dapat dilewatkan ke pihak lain dan diolah oleh pihak tersebut. Dalam sistem ini dimungkinkan suatu aplikasi dapat mengambil data dari satu sumber tanpa harus tahu bahwa sebenarnya data tersebut dihasilkan melalui proses pengolahan oleh sistem lain sehingga dapat terjadi integrasi data maupun aplikasi yang sering disebut dengan A2A (*application to application*).

Dalam Kreger (2001) dikatakan bahwa model dari sebuah web-service didasarkan pada interaksi antara 3 komponen yang berperan dalam web-service, yaitu: service provider, service registry dan service requestor/consumer. Interaksi yang terjadi antara ketiga komponen tersebut juga melibatkan operasi *publish*, *find* dan *bind*. Service provider menyediakan service yang dapat diakses melalui jaringan komputer, misalnya internet. Kemudian, service provider mendeskripsikan service yang dibangun dan mem-*publish*-kan service description tersebut ke service registry atau secara langsung ke service consumer. Service requestor/consumer menggunakan operasi *find* untuk mendapatkan service description secara lokal maupun melalui service registry. Service description yang diperoleh itu kemudian digunakan

untuk men-*bind* service provider dan berinteraksi dengan implementasi web-service yang akan digunakan tersebut.

2.1 XML (*Extensible Markup Language*)

Menurut Walsh (1998), XML merupakan sebuah *Markup Language* untuk dokumentasi terstruktur. Dokumen-dokumen terstruktur adalah dokumen-dokumen yang mempunyai isi/*content* (kata, gambar) serta indikasi yang menyatakan makna dari *content* tersebut. XML mempunyai kelebihan sebagai berikut (Tidwell, 1999):

- a. XML tidak tergantung pada platform atau sistem operasi yang digunakan.
- b. Hasil pencarian data lebih akurat.
- c. Dokumen XML dapat diterjemahkan ke dalam beberapa format yang berbeda karena dalam XML data dan instruksi dipisahkan.

Ada 6 jenis markup yang bisa muncul dalam sebuah dokumen XML yaitu:

- a. Elemen dan atribut. Elemen menyatakan sifat dari *content* yang dilingkupinya sedangkan atribut merupakan pasangan dari nama-nilai yang muncul dalam tag setelah nama elemen.
- b. *Entity reference*, digunakan supaya tanda markup dapat dimasukkan ke dalam dokumen XML dan dianggap sebagai *content*.
- c. *Comment* atau komentar.
- d. *Processing Instruction* (PI), memungkinkan dokumen berisi suatu instruksi untuk suatu aplikasi.
- e. CDATA Section. Dalam sebuah dokumen, CDATA Section menginstruksikan *parser* untuk mengabaikan karakter-karakter tertentu yang mungkin akan dikenali sebagai karakter markup.
- f. *Document Type Declaration* (DTD). DTD berisi deklarasi markup yang memenuhi grammar untuk suatu kelas dokumen.

2.2 Simple Object Access Protocol (SOAP)

SOAP (*Simple Object Access Protocol*) merupakan protokol yang digunakan untuk mempertukarkan data atau informasi dalam format XML (Scheinblum, 2001). SOAP dapat dikatakan sebagai gabungan antara HTTP dengan XML karena SOAP umumnya menggunakan protokol HTTP sebagai sarana transport datanya dan data yang akan dipertukarkan ditulis dalam format XML. Karena SOAP menggunakan HTTP dan XML maka SOAP memungkinkan pihak-pihak yang mempunyai platform, sistem operasi dan perangkat lunak yang berbeda dapat saling mempertukarkan datanya.

Pada dasarnya SOAP mengikuti model transmisi pesan HTTP yang bersifat *request-response* dimana parameter SOAP request diletakkan dalam HTTP request dan parameter SOAP response diletakkan dalam HTTP response.

2.3 Web-services Description Language (WSDL)

Menurut Shohoud (2001) WSDL merupakan sebuah bahasa berbasis XML yang digunakan untuk mendefinisikan web-service dan menggambarkan bagaimana cara untuk mengakses web-service tersebut.

Deskripsi WSDL mendefinisikan sebuah service sebagai kumpulan dari port dimana tiap-tiap port didefinisikan secara abstrak sebagai portType yang mendukung sekumpulan operasi-operasi. Tiap-tiap operasi memproses sekumpulan pesan tertentu.

Dalam Manes (2001) disebutkan bahwa ada 5 elemen utama dalam sebuah dokumen WSDL yaitu:

- a. Elemen `<type>`, berfungsi untuk mendefinisikan tipe data-tipe data yang digunakan dalam pesan.
- b. Elemen `<message>`, berfungsi untuk mendefinisikan format dari sebuah pesan. Pesan digunakan sebagai struktur masukan (input) atau keluaran (output) bagi operasi.
- c. Elemen `<portType>`, berfungsi untuk mendefinisikan sekumpulan operasi-operasi. Tiap-tiap elemen `<operation>` mendefinisikan sebuah operasi dan pesan masukan atau keluaran yang berkaitan dengan operasi tersebut.
- d. Elemen `<binding>`, berfungsi untuk memetakan operasi-operasi dan pesan yang terdefiniskan pada port type ke protokol tertentu.
- e. Elemen `<service>`, berfungsi untuk mendefinisikan sekumpulan port-port yang saling berhubungan. Elemen `<port>` memetakan binding ke lokasi dari sebuah web-service.

2.4 Universal Description, Discovery and Integration (UDDI)

Menurut Ariba, IBM dan Microsoft (2000) UDDI merupakan suatu cara untuk mempublikasikan dan menemukan informasi tentang web-service. Dalam dunia internet sekarang ini, UDDI bisa dianalogikan dengan search engine. Perbedaan antara UDDI dengan search engine adalah bahwa search engine hanya berisi URL dari suatu website, search engine tidak dapat memberikan hasil misalnya berupa alamat e-mail. Karena search engine hanya berisi URL maka hanya ada satu format data yang dapat ditampilkan yaitu HTML. Search engine tidak dapat menampilkan dokumen SOAP maupun dokumen XML. Di samping itu, search engine tidak dapat digunakan untuk memanggil aplikasi lain yang terletak di tempat lain (*remote call*). Sedangkan UDDI mampu melakukan hal-hal yang tidak dapat dilakukan oleh search engine, misalnya memanggil aplikasi lain yang terletak secara *remote* (Accenture dkk, 2001).

Untuk mencari sebuah web-service, service consumer terlebih dahulu mengirimkan *query* ke UDDI registry untuk mencari service yang diinginkan. Dari tModel, service consumer akan mendapatkan deskripsi WSDL yang menyatakan antarmuka service (*service interface*). Dengan menggunakan deskripsi WSDL yang telah diperoleh tersebut, service consumer dapat membuat sebuah pesan SOAP untuk berkomunikasi dengan web-service yang diinginkan.

2.5 Vendor Utama dalam Web-service

Saat ini, web-service sudah didukung oleh ratusan vendor perangkat lunak di dunia. Diantara vendor-vendor tersebut, terdapat 3 vendor besar yang sangat aktif dalam pengembangan web-service yaitu: Microsoft, IBM dan Oracle.

Microsoft mendukung pengembangan web-service melalui produknya Microsoft Visual Studio .NET. Bahasa pemrograman baru yang ditambahkan pada Visual Studio .NET adalah Visual C# .NET. Karakteristik yang dimiliki oleh C# adalah sebagai berikut (Trupin, 2001): *simple*, *consistency*, *object-orientation* dan *Type safety*. Selain itu, C# juga sudah mengimplementasikan *automatic garbage collection* dimana ruang memori yang tidak terpakai lagi akan dibersihkan secara otomatis. C# mempunyai 2 tipe yaitu: *value type* dan *reference type* (Active State dkk, 2001). *Value type* meliputi tipe data sederhana (float, int, char, dll), tipe data enum dan tipe data struct. Sedangkan *reference type* mencakup tipe class, interface, delegate dan array.

Sebagai sebuah bahasa pemrograman, C# juga mempunyai mekanisme untuk melakukan pemanggilan fungsi. Jenis pemanggilan fungsi dalam C# terbagi menjadi 2 yaitu: *call by value* dan *call by reference* (Tien, 2001).

Microsoft .NET framework merupakan platform bagi Microsoft Visual Studio .NET. Ada 3 komponen utama pada Microsoft .NET framework yaitu (Books, 2001):

- a. *Common Language Runtime (CLR)*. CLR merupakan lingkungan dimana aplikasi .NET yang telah dikompilasi menjadi intermediate language (IL) dijalankan. IL dikompilasi lagi dengan cara *Just In Time* ke bahasa native mesin. Hal yang menarik dari CLR adalah memungkinkan terjadinya integrasi antar bahasa (*Multilanguage integration*) karena pada CLR terdapat common type system yang berisikan tipe data-tipe data standar.
- b. *.NET Framework Base Classes*. Layer ini menyediakan banyak class dan model obyek untuk data, input/output, security, dan lain sebagainya. Class-class yang terdapat pada layer ini dapat dimanfaatkan dari berbagai macam bahasa pemrograman yang berbasis .NET Framework sehingga memberikan efek

ketersediaan (*availability*) dan konsistensi yang tinggi antar bahasa pemrograman.

- c. User dan Program Interface. Pada layer ini terdapat Windows forms (WinForms) untuk membuat aplikasi desktop. Selain itu, juga terdapat web forms dan web-services yang diimplementasikan pada ASP.NET.

Selain Microsoft, IBM juga memberikan sumbangan bagi pengembangan web-service. Dalam Kraft (2002) dikatakan bahwa IBM Web Sphere Studio Application Developer merupakan lingkungan pengembangan yang bersifat visual untuk membangun web-service yang tidak lagi bergantung pada platform yang digunakan (*platform-neutral*). Walaupun tujuan utama dari Web Sphere Studio adalah untuk mendukung pengembangan web-service, namun Web Sphere Studio juga dapat digunakan untuk membangun web site dan aplikasi desktop. Sedangkan Oracle mendukung implementasi web-service melalui produknya Oracle9i Web-services atau sering disebut dengan Oracle9i Dynamic Service. Produk-produk Oracle yang masuk dalam kelompok Oracle9i Web-services yang dapat digunakan untuk mengembangkan web-service adalah (Cheng dan Goodell, 2001):

- a. Oracle9i Jdeveloper, digunakan untuk membangun web-service dengan menggunakan bahasa pemrograman Java.
- b. Oracle9i Application Server, berfungsi untuk menyebarkan, mengatur dan memberikan fasilitas keamanan (*security*) bagi web-service. Selain itu, Oracle9i AS juga mempunyai kemampuan untuk mengintegrasikan web-service dengan aplikasi-aplikasi yang telah ada sebelumnya.
- c. Oracle9i Web-service Broker merupakan engine yang melaksanakan service. Web-service broker menerima input dari client berupa service description, kemudian web-service broker akan membuat suatu *execution environment*, mengirimkan permintaan eksekusi service ke service provider, setelah itu menerima respon dari service provider dan merubah respon dari service provider tersebut menjadi informasi yang akan dikirimkan kembali ke client.

3. Perancangan Sistem

Untuk memberikan gambaran mengenai cara kerja dan manfaat web-service, maka akan dibangun sebuah web-service utama dengan nama "WisataBudayaJawa" yang akan memberikan output berupa saran atau rekomendasi dalam bentuk daftar sejumlah lokasi obyek wisata yang dapat dijadikan sebagai panduan bagi wisatawan untuk menyusun rangkaian perjalanan wisata budaya Jawanya. Adapun yang dimaksud dengan wisata budaya Jawa, adalah suatu paket wisata yang

menonjolkan potret-potret kehidupan keseharian masyarakat Jawa. Sedangkan yang dimaksud dengan masyarakat Jawa di sini adalah masyarakat atau komunitas penduduk asli pulau Jawa yang mendiami wilayah Jawa Tengah dan Daerah Istimewa Yogyakarta.

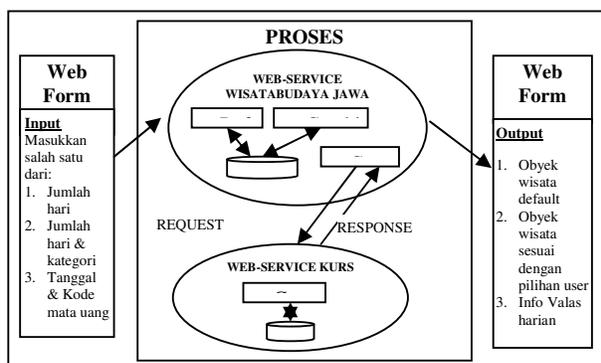
Output yang akan diberikan oleh web-service "WisataBudayaJawa" terdiri-dari 2 model yaitu: output statis (*default*) dan output dinamis. Dalam output statis, wisatawan tidak terlibat dalam pemilihan komponen budaya Jawa karena web-service hanya memberikan output berdasarkan jumlah hari yang tersedia dan dalam urutan yang telah ditentukan. Sedangkan dalam output dinamis, user yang memilih secara langsung komponen budaya Jawa yang ingin diketahuinya. Berdasarkan komponen budaya Jawa yang dipilih tersebut, web-service akan memberikan rekomendasi obyek wisata yang sebaiknya dikunjungi sesuai dengan batas waktu yang tersedia. Semua output yang diberikan tersebut akan dikelompokkan berdasarkan kode hari, sehingga dari pengelompokkan tersebut user mendapatkan bantuan yang akan memberikan panduan dalam merancang rute perjalanan wisata budaya Jawanya.

Web-service "WisataBudayaJawa" ini akan diintegrasikan dengan web-service "Kurs" yang memberikan informasi harian nilai tukar sejumlah mata uang asing terhadap mata uang Rupiah sehingga dari integrasi tersebut, gambaran mengenai perpaduan antar web-service akan semakin jelas. Selain itu, web-service "WisataBudayaJawa" juga akan dijalankan pada sistem operasi Linux untuk mengetahui apakah web-service dapat berjalan pada sistem operasi yang berbeda dengan sistem operasi service provider.

Untuk memenuhi semua fungsi tersebut, maka ada 3 method yang akan disediakan pada web-service "WisataBudayaJawa" yaitu: *getDefault*, *getCombination* dan *getCurrency* dengan method *getCurrency* merupakan method yang dikembangkan pada web-service "Kurs". Method *getDefault* berfungsi untuk menampilkan obyek wisata yang telah ditetapkan sebagai output *default* sesuai dengan jumlah hari wisatawan yang tersedia. Method ini hanya menerima 1 masukan yaitu: berupa jumlah hari wisatawan yang tersedia. Method *getCombination* berfungsi untuk menampilkan obyek wisata sesuai dengan komponen budaya Jawa yang dipilih oleh user dan sesuai dengan jumlah hari wisatawan yang tersedia. Method ini menerima 2 masukan yaitu: komponen budaya Jawa dan jumlah hari yang tersedia. Sedangkan method *getCurrency* berfungsi untuk memberikan informasi harian mengenai nilai tukar sejumlah mata uang asing terhadap mata uang Rupiah. Parameter input dari method ini adalah

kode mata uang dan tanggal dimana wisatawan ingin mengetahui nilai tukar tersebut.

Untuk memberikan gambaran mengenai bagaimana pemanfaatan web-service dalam membangun sebuah program aplikasi, maka akan dibangun sebuah aplikasi web yang memanfaatkan method-method yang disediakan dalam web-service “WisataBudayaJawa”. Aplikasi web tersebut terdiri-dari 2 menu yaitu: “Travelling Planning Guide” yang memanfaatkan method *getDefault* dan *getCombination*, yang berfungsi untuk memberikan saran mengenai obyek wisata yang sebaiknya dikunjungi oleh wisatawan dalam rangka wisata budaya Jawa serta “Money Exchange Info” yang memanfaatkan method *getCurrency* dan berfungsi untuk memberikan informasi harian nilai tukar sejumlah mata uang asing terhadap mata uang Rupiah.



Gambar 1. Arsitektur sistem web-service WisataBudayaJawa

Pada sistem ini, service consumer hanya melihat satu buah web-service yaitu: web-service “WisataBudayaJawa” dengan 3 buah method: *getDefault*, *getCombination* dan *getCurrency*. Ketika method *getCurrency* dipanggil, tanpa disadari oleh service consumer, parameter input yang telah diberikan akan dilewatkan ke web-service “Kurs” untuk diolah dan hasilnya akan dikembalikan lagi ke web-service “WisataBudayaJawa” untuk ditampilkan ke service consumer.

Arsitektur sistem yang menggambarkan keseluruhan struktur sistem dari web-service “WisataBudayaJawa” dan web-service “Kurs” terlihat pada gambar 1.

Dari gambar 1 dapat dijelaskan bahwa proses yang akan terjadi adalah sebagai berikut: mula-mula user memilih salah satu dari 3 method yang akan digunakan. Jika method yang dipilih adalah *getDefault*, maka user cukup memasukkan jumlah hari yang tersedia sebagai parameter input dan

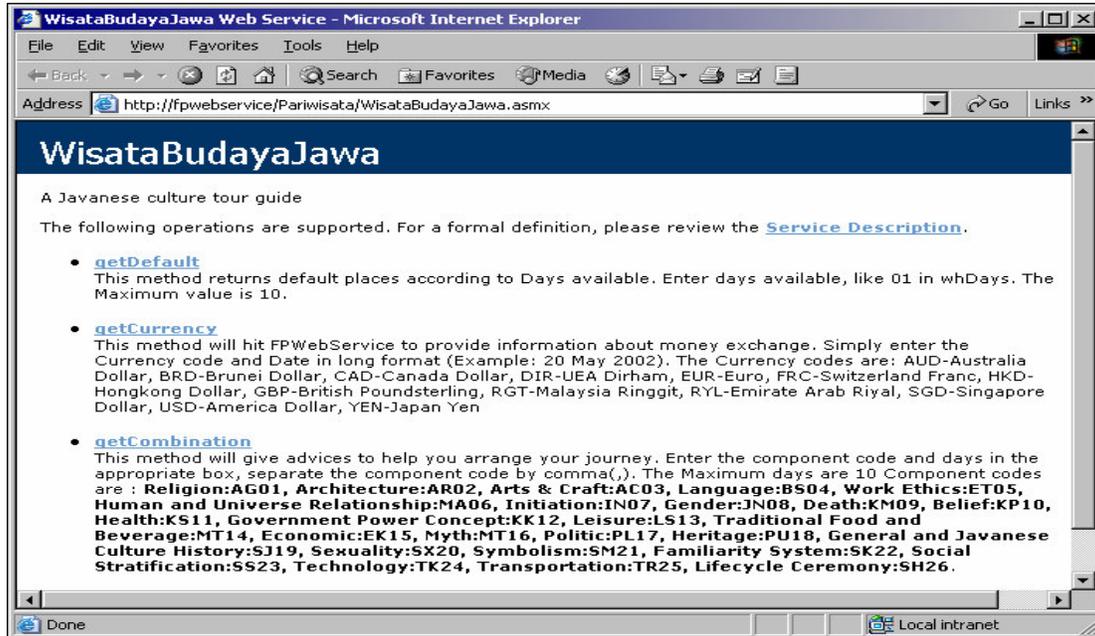
method *getDefault* akan menampilkan daftar obyek-obyek wisata sesuai dengan jumlah hari yang tersedia dan dalam urutan yang telah ditetapkan. Jika method yang dipilih adalah *getCombination*, maka user harus memasukkan jumlah hari yang tersedia dan komponen budaya Jawa yang ingin diketahui dan method *getCombination* akan menampilkan obyek-obyek wisata yang mengandung unsur budaya Jawa seperti yang telah dipilih oleh user serta dalam batas waktu sesuai dengan jumlah hari yang tersedia. Jika method *getCurrency* yang dipilih, maka input yang dibutuhkan adalah tanggal dan kode mata uang. Pada proses ini, parameter input akan dilewatkan oleh web-service “WisataBudayaJawa” ke web-service “Kurs” untuk diolah oleh method *getCurrency* dan outputnya akan dikirim kembali ke web-service “WisataBudayaJawa” untuk ditampilkan ke user.

4. Pembahasan

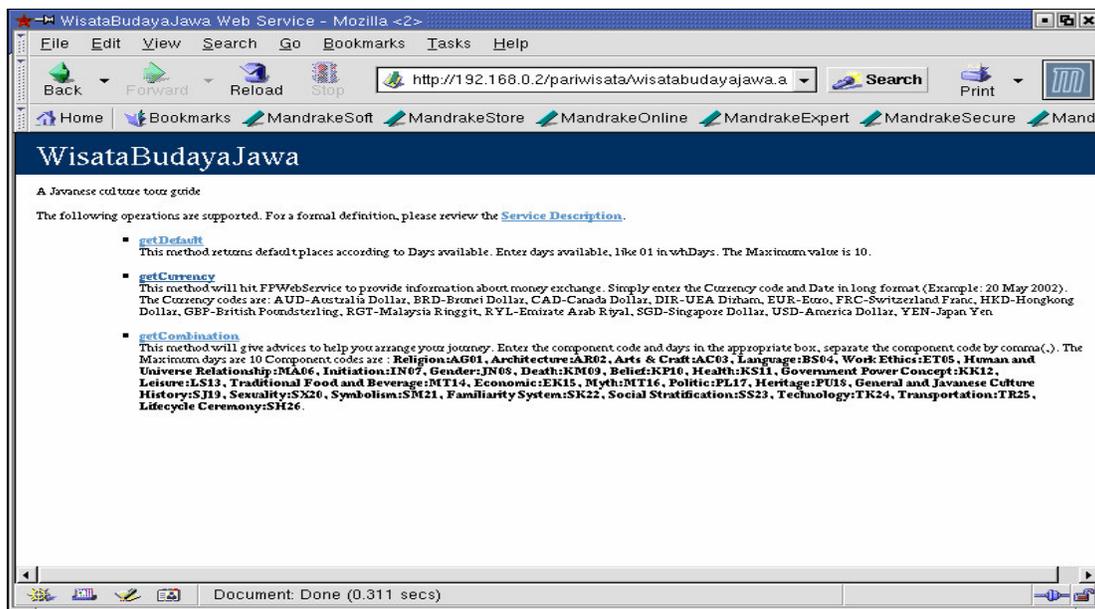
Berdasarkan arsitektur sistem yang telah dirancang pada bab III, maka dikembangkan 2 buah web-service yaitu: web-service “WisataBudayaJawa” dan web-service “Kurs”. Dalam sistem tersebut, web-service “WisataBudayaJawa” merupakan web-service utama yang terlihat oleh client sedangkan web-service “Kurs” akan saling dihubungkan dengan web-service “WisataBudayaJawa”. Tampilan dari web-service “WisataBudayaJawa” pada sistem operasi Windows dan Linux terlihat pada gambar 2 dan 3

Pada *interface* (antarmuka) sebuah web-service terlihat method-method yang disediakan oleh web-service tersebut. Untuk web-service “WisataBudayaJawa” di atas terlihat bahwa method yang disediakan adalah *getDefault*, *getCombination* dan *getCurrency*. Sedangkan output yang diberikan oleh web-service akan ditampilkan dalam format XML seperti gambar 4.

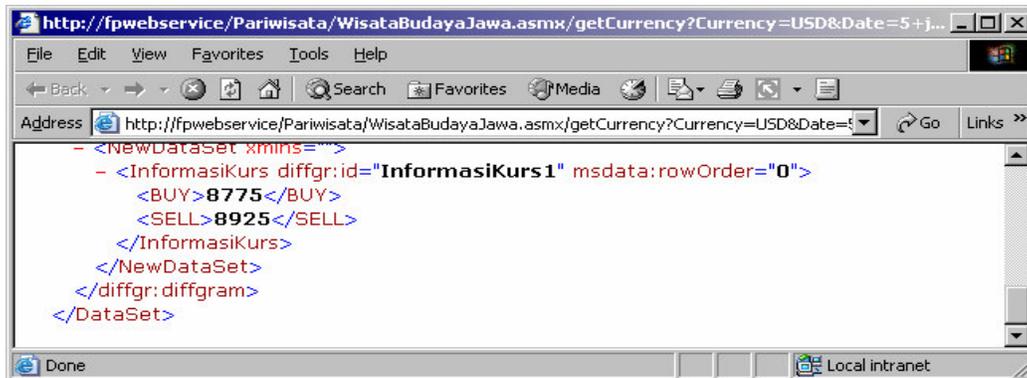
Method-method yang disediakan oleh web-service “WisataBudayaJawa” yang telah diintegrasikan dengan web-service “Kurs” akan dimanfaatkan untuk membangun sebuah aplikasi web. Parameter input dari aplikasi web tersebut dikirimkan ke web-service “WisataBudayaJawa” untuk diproses dan output XML dari web-service tersebut akan ditampilkan ke client. Gambar 5 dan Gambar 6 memperlihatkan menu “Money Exchange Info” dan output menu tersebut yang diambil dari output XML method *getCurrency* pada web-service “WisataBudayaJawa”.



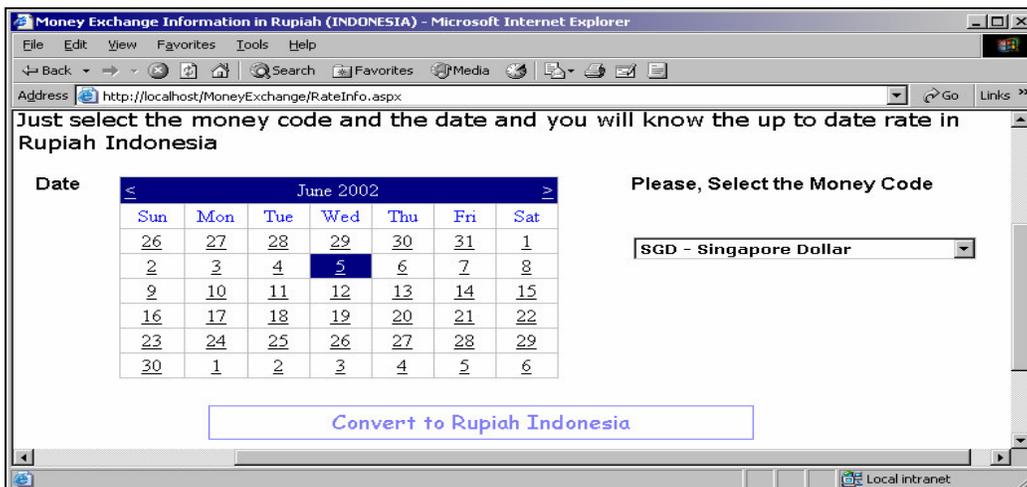
Gambar 2. Tampilan web-service WisataBudayaJawa pada sistem operasi Windows



Gambar 3. Tampilan web-service WisataBudayaJawa pada sistem operasi Linux



Gambar 4. Output XML dari web-service



Gambar 5. Tampilan menu Money Exchange Info



Gambar 6. Tampilan output dari menu Money Exchange Info

Sebuah web-service perlu memiliki service description agar penggunaannya dapat dimengerti oleh service consumer. Service description dari sebuah web-service dideskripsikan dengan menggunakan WSDL (Web-services Description Language) dalam format XML. Elemen-elemen service description dari sebuah web-service adalah:

a. Elemen `<service>` berfungsi untuk mendeskripsikan service. Di dalam elemen ini terdapat nama service, elemen `<port>` yang menyatakan alamat dari web-service serta

protokol yang didukung oleh web-service. Gambar 7 memperlihatkan cuplikan elemen `<service>` dari web-service “WisataBudayaJawa”.

b. Elemen `<message>` berfungsi untuk mendeskripsikan pesan. Di dalam elemen ini terdapat elemen `<part>` yang menggambarkan parameter input dan output beserta tipe datanya dari masing-masing method yang disediakan oleh web-service. Gambar 8 memperlihatkan

cuplikan elemen `<message>` dari web-service “WisataBudayaJawa”.

```
<service name = "WisataBudayaJawa">
  <documentation>A Javanese culture tour
  guide</documentation>
  <port name = "WisataBudayaJawaSoap" binding =
  "s0:WisataBudayaJawaSoap">
    <soap:address location =
    "http://fpwebservice/Pariwisata/WisataBudayaJawa.asmx"
    />
  <port name = "WisataBudayaJawaHttpGet" binding =
  "s0:WisataBudayaJawaHttpGet">
    <http:address location =
    "http://fpwebservice/Pariwisata/WisataBudayaJawa.asmx"
    />
  <port name = "WisataBudayaJawaHttpPost" binding =
  "s0:WisataBudayaJawaHttpPost">
    <http:address location =
    "http://fpwebservice/Pariwisata/WisataBudayaJawa.asmx"
    />
  </port>
</service>
```

Gambar 7. Elemen `<service>` web-service WisataBudayaJawa

```
<message name = "getDefaultSoapIn">
  <part name = "parameters" element =
  "s0:getDefault" />
</message>
<message name = "getDefaultSoapOut">
  <part name = "parameters" element =
  "s0:getDefaultResponse" />
</message>
<message name = "getDefaultHttpGetIn">
  <part name = "whDay" type = "s:string" />
</message>
<message name = "getDefaultHttpGetOut">
  <part name = "Body" element = "s0:DataSet" />
</message>
```

Gambar 8. Elemen `<message>` dari web-service WisataBudayaJawa

- c. Elemen `<portType>` yang menyatakan kumpulan dari method-method beserta elemen `<operation>` yang menyatakan pesan (*message*) apa yang merupakan input dan pesan (*message*) apa yang merupakan output. Gambar 9 memperlihatkan cuplikan elemen `<portType>` dari web-service “WisataBudayaJawa”.
- d. Elemen `<binding>` berfungsi untuk memetakan method-method dan pesan yang terdefiniskan pada portType ke protokol dan format data tertentu. Atribut *type* di dalam elemen `<binding>` merujuk ke elemen `<portType>` yang bersesuaian. Untuk protokol SOAP pada web-service “WisataBudayaJawa”, transport data yang digunakan adalah HTTP yang terlihat pada atribut *transport*. Gambar 10 memperlihatkan cuplikan elemen `<binding>` dari web-service “WisataBudayaJawa”.

```
<portType name = "WisataBudayaJawaSoap">
  <operation name = "getDefault">
    <input message = "s0:getDefaultSoapIn" />
    <output message = "s0:getDefaultSoapOut" />
  </operation>
</portType>
<portType name = "WisataBudayaJawaHttpGet">
  <operation name = "getDefault">
    <input message = "s0:getDefaultHttpGetIn" />
    <output message = "s0:getDefaultHttpGetOut" />
  </operation>
</portType>
```

Gambar 9. Elemen `<portType>` dan `<operation>` web-service WisataBudayaJawa

```
<binding name = "WisataBudayaJawaSoap" type =
"s0:WisataBudayaJawaSoap">
  <soap:binding transport =
  "http://schemas.xmlsoap.org/soap/http" style = "document"
  />
  <operation name = "getDefault">
    <soap:operation soapAction =
    "http://fpwebservice/WisataBudayaWebService/getDefault"
    style = "document" />
  </operation>
</binding>
<binding name = "WisataBudayaJawaHttpGet" type =
"s0:WisataBudayaJawaHttpGet">
  <http:binding verb = "GET" />
  <operation name = "getDefault">
    <http:operation location = "/getDefault" />
  </operation>
</binding>
```

Gambar 10. Elemen `<binding>` web-service WisataBudayaJawa

Service description yang telah selesai dibuat perlu dipublikasikan ke UDDI registry supaya dapat dicari dan dimanfaatkan oleh service consumer. UDDI registry yang digunakan adalah UDDI registry Microsoft yang dapat diakses pada URL <http://uddi.microsoft.com>. Hal yang perlu diisikan pada saat akan mempublikasikan service description adalah nama web-service, deskripsi singkat web-service serta alamat file WSDL dari web-service yang dipublikasikan.

Service description yang telah dipublikasikan ke UDDI registry dapat dicari oleh service consumer menggunakan *keyword* atau kata kunci dari nama service yang bersesuaian. Jika UDDI registry menemukan kata kunci tersebut, maka UDDI registry akan menampilkan daftar web-service yang mengandung kata kunci tersebut. Dari daftar web-service tersebut, service consumer bisa melihat *detail* (rincian) tModel yang meliputi nama web-service, alamat file WSDL serta *identifier* dari web-service.

5. Penutup

5.1 Kesimpulan

Setelah melakukan studi literatur dan studi kasus dengan pengembangan sistem didapat beberapa kesimpulan sebagai berikut:

- a. 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*.
- Dalam melakukan pengembangan aplikasi client “wisata budaya Jawa”, web-service ternyata sangat efektif untuk digunakan karena method atau fungsi dalam web-service dapat langsung digunakan untuk mengembangkan aplikasi client sehingga tidak perlu dilakukan pemrograman aplikasi client dari awal.
 - Adanya integrasi antara web-service “WisataBudayaJawa” dengan web-service “Kurs” yang telah dibangun, mengakibatkan ketersediaan service atau fungsi (unsur *availability*) menjadi semakin besar sekaligus telah menggambarkan suatu sistem *multi tier* karena service provider hanya melewatkan *request* dari client ke service provider yang bersangkutan untuk diproses dan memberikan hasil pemrosesan dari service provider kembali ke client.
 - Setelah melakukan percobaan penggunaan web-service “WisataBudayaJawa” pada sistem operasi Linux, membuktikan bahwa web-service dapat berjalan pada sistem operasi yang berbeda dengan sistem operasi dimana web-service tersebut dikembangkan.

5.2 Pengembangan selanjutnya

Beberapa hal yang masih dapat dikembangkan atas sistem “wisata budaya Jawa” maupun aplikasi web-service pada umumnya sehingga diharapkan implementasi web-service akan semakin sempurna. Diantaranya adalah sebagai berikut:

- Untuk web-service “WisataBudayaJawa” dapat dikembangkan dengan menambahkan algoritma jarak terpendek sehingga output yang diberikan oleh web-service selain sesuai dengan pilihan user juga akan memberikan efisiensi dalam melakukan perjalanan dari satu kota ke kota yang lain.
- Pemanfaatan web-service dalam membangun aplikasi-aplikasi untuk perangkat *mobile computing* seperti telepon genggam dan PDA (Personal Digital Assistant).
- Pengembangan web-service dengan melibatkan aspek keamanan. Sebagai contoh, akses terhadap method-method dalam web-service hanya diberikan kepada service consumer tertentu saja dan bagaimana menjamin keamanan data yang ditransfer dari service provider ke service consumer.

Daftar Pustaka

- Accenture, Ariba, Commerce One, Compaq, Fujitsu, Hewlett-Packard, I2 Technologies, Intel, IBM, Microsoft, Oracle, SAP AG, Sun

- Microsystems, Verisign, 2001, “UDDI Executive White Paper”, <http://www.uddi.org>.
- Active State, Hewlett-Packard, Intel, IBM, Microsoft, Mountain View Compiler, Netscape, Pico, Plum Hall, Sun Microsystems, 2001, “Draft C# Language Specification”, <http://www.ecma.ch>.
- Ariba, IBM, Microsoft, 2000, “UDDI Technical White Paper”, <http://www.uddi.org>.
- Books, W., 2001, “Introducing .NET Part 2 – The .NET Framework”.
- Cheng, R., Goodell, M., 2001, “Oracle9i Web-services Overview”, Oracle Corporation, USA.
- Gani, A., 2001, “Aplikasi berbasis web (web enabled application)”, PT Puspa Intimedia Internusa, Jakarta.
- Hamids, 2000, “Introduction to web-services”, <http://www.MCPCentral.com>.
- Kraft, R., 2002, “How IBM WebSphere Studio Application Developer Compares with Microsoft Visual Studio .NET -- Part 1: Conceptual Differences”, IBM Corporation, USA.
- Kreger, H., 2001, “Web-services Conceptual Architecture (WSCA 1.0)”, IBM Software Group, USA.
- Manes, A.T., 2001, “Introduction to Web-services”, <http://www.systinet.com>.
- Meiyanto, M.E., 2001, Skripsi: *Extensible Markup Language (XML) untuk Pertukaran Data di Internet*, Yogyakarta.
- Microsoft Corp., 2000, “Application Service Provider: Evolution and Resources (A White Paper)”, USA.
- Microsoft Corp., 2001, “Microsoft .net Framework”, USA.
- Scheinbum, J., 2001, “An Introduction to SOAP”, <http://www.zdnetindia.com/Techzone/coding/stories/29727.html>
- Shohoud, Y., 2001, “Introduction to WSDL”, DevXpert Corporation.
- Tidwell, D., 1999, “Tutorial: Introduction to XML”, Raleigh, NC.
- Tien, T.S., 2001, *Bahasa C# untuk Pemrograman Berorientasi Objek*, PT. Elex Media Komputindo, Jakarta.
- Trupin, J., 2001, “Sharp New Language: C# Offers the Power of C++ and Simplicity of Visual Basic”.
- Walsh, N., 1998, “A Technical Introduction to XML”, ArborText, Inc.

Pernyataan

Makalah ini pernah disampaikan pada Seminar Nasional dalam rangka Dies Natalis ke-47 Fakultas MIPA, Universitas Gadjah Mada, 21 September 2002, Yogyakarta.