



**PROSIDING  
SEMINAR NASIONAL ILMU KOMPUTER UNIVERSITAS DIPONEGORO 2010  
SEMARANG, 7 AGUSTUS 2010**

## DESAIN *WEB SERVICE* PADA KATALOG TOKO BUKU

Ragil Saputra

Program Studi Ilmu Komputer FMIPA UNDIP  
Jl. Prof. Soedharto, SH Tembalang Semarang  
[ragil.saputra@gmail.com](mailto:ragil.saputra@gmail.com)

### Abstrak

Perkembangan teknologi informasi terutama internet memberikan berbagai kemudahan akses informasi, tetapi disisi lain menyebabkan informasi berlebih. Sehingga diharapkan setiap situs menyediakan potongan informasi yang dapat diakses situs lain. Permasalahan yang sering dihadapi adalah *interoperability* atau pengintegrasian aplikasi yang berbeda *platform* aplikasi. Pada penelitian ini dibuat *prototype* pengintegrasian dua buah situs toko buku online dengan *database* yang berbeda dengan menggunakan *web service* untuk melakukan pencarian katalog buku. Pertukaran data dalam format *XML* (*eXtensible Markup Language*) menggunakan teknologi SOAP (*Simple Object Acces Protocol*) dan WSDL (*Web Service Description Language*), serta menggunakan *library* NuSOAP. NuSOAP digunakan untuk menyediakan *class soapClient* dan *soapServer*, atau sebagai *listener* yang akan menerima dan merespon permintaan akses terhadap *web service*. Hasil yang diperoleh pada penelitian ini adalah berupa aplikasi *client* yang mampu mengintegrasikan data katalog melalui *web service* dari dua buah *server*.

**Kata kunci :** *web service*, *XML*, *interoperability*, katalog

### 1. Pendahuluan

Perkembangan internet yang pesat membawa banyak perubahan dalam bertukar informasi. Dengan hanya memasukkan kata kunci pada mesin pencari, pengguna internet bisa mendapatkan ribuan bahkan jutaan informasi yang tersedia. Selain kemudahan yang dimiliki internet, disana ada beberapa tantangan. Pengguna harus melompat dari satu situs ke situs yang lain untuk mendapatkan informasi tersebut. Dan hal ini kadang merepotkan pengguna. Sebagai contoh untuk mendapatkan informasi buku-buku terbaru pada situs toko buku *online*, pengguna harus mengunjungi situs penyedia informasi tersebut. Untuk mengatasi permasalahan tersebut, sebuah situs harus menyediakan potongan informasi yang dapat diakses oleh situs lain. Hal ini biasa disebut dengan *syndicated content* [5].

Ada beberapa kategori *syndicate content*, antara lain *content distribution*, *content gathering*, *content brokering*, dan *content networking* [5]. Teknologi yang mendukung pertukaran informasi adalah dengan memanfaatkan komputasi terdistribusi (*distributed computing*) yang dapat melakukan komputasi pada banyak mesin dan hasilnya dimanfaatkan oleh banyak mesin pula. Permasalahan yang sering dihadapi adalah *interoperability* atau kemampuan untuk meng-

integrasikan aplikasi yang berbeda diantara situs internet. *Web service* dapat dijadikan solusinya.

*Web service* adalah sebuah sistem perangkat lunak yang dirancang untuk mendukung inter-operasi dalam interaksi mesin ke mesin dalam sebuah jaringan [7]. Interaksi dilakukan melalui mekanisme atau protokol tertentu. Dengan demikian kemampuan *web service* dapat menambah kemampuan web untuk saling berkomunikasi dan bertukar informasi dengan pola *program-to-program*.

Teknologi *web service* dapat mengintegrasikan sistem, bahasa pemrograman, basis data dan *platform* sistem operasi yang berbeda melalui protokol HTTP (*Hyper Text Transfer Protocol*) , sehingga aplikasi-aplikasi yang berbeda dapat saling berkomunikasi dengan baik.

Amazon adalah pelopor *eCommerce* untuk penjualan buku. Situs amazon juga dilengkapi dengan fasilitas *web service*, sehingga siapa saja dapat mengambil informasi tanpa harus mengunjungi situsnya. Dalam penelitian ini dibuat *prototype* rancangan *web service server* untuk memberikan layanan informasi katalog buku dan *service client* untuk mengakses *service* tersebut.

Tujuan penelitian ini adalah melakukan analisis perancangan, implementasi pengintegrasian data katalog buku pada situs penjualan toko buku online. Manfaatnya yaitu untuk memberikan gambaran bagaimana langkah-langkah yang diperlukan dan dipersiapkan untuk mewujudkan integrasi data.

## XML

XML (*eXtensible Markup Language*) adalah bagian dari SGML (*Standart Generalized Markup Language*). XML merupakan bagian yang penting bagi pengembang yang ingin membangun web service. XML berbasis teks, sehingga bersifat *platform independent*. Sifat ini membuat pertukaran data antar *platform* menjadi mungkin untuk dilakukan.

Sebuah objek data dapat dikatakan sebagai dokumen XML jika memenuhi kriteria *well-formed*. Adapun kriteria dokumen *well-formed* adalah sebagai berikut [2] :

1. Satu dan hanya satu *root element* yang ada dalam satu dokumen
2. Setiap dokumen harus memiliki *start-tag* dan *end-tag*
3. Setiap elemen harus *properly nested*
4. Karakter pertama dari nama atribut harus berupa huruf *alfabet* atau *underscore*
5. Suatu nama atribut hanya boleh muncul satu kali pada *start-tag* yang sama.

Pada gambar 1 berikut ini adalah contoh dokumen XML yang *well-formed* :

```
<katalog>
<buku idbuku="1">
  <properti>
    <judul>Just XML</judul>
    <pengarang>John E. Simpson</pengarang>
    <penerbit>Andi Offset</penerbit>
    <isbn>979-533-840-4</isbn>
  </properti>
</buku idbuku="1">
</katalog>
```

Gambar 1. Contoh Dokumen XML

Untuk menjelaskan struktur dokumen XML beserta informasi tipenya diperlukan skema. Terdapat dua teknologi dominan yang dapat digunakan untuk mendefinisikan skema XML yaitu *Document Type Definition (DTD)* dan *XML Schema* [4]. Gambar 2 berikut ini memberikan contoh *xml schema* :

```
<?xml version='1.0'?>
<schema
xmlns='http://www.w3.org/2001/XMLSchema'>
  <!-- Response Message (work-in-progress)-->
  <element name='Amount' />
</schema>
```

Gambar 2. Contoh XML Schema

## WEB SERVICE

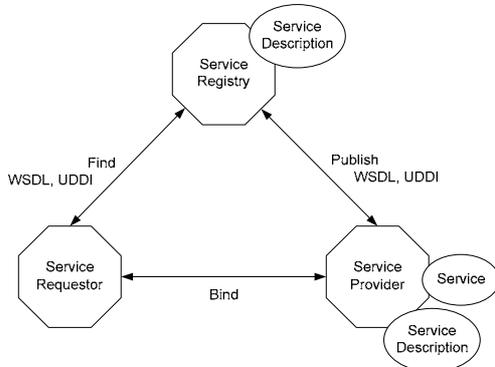
Beberapa pengertian tentang *web service*, *web service* adalah antarmuka yang mendeskripsikan sekumpulan operasi yang dapat diakses dalam sebuah jaringan melalui pesan XML yang telah distandardkan [3]. W3C mendefinisikan *web service* sebagai sebuah sistem perangkat lunak yang dirancang untuk mendukung inter operasi mesin ke mesin di sebuah jaringan [7]. *Web service* merupakan komponen perangkat lunak *loosely coupled*, dapat diguna ulang, membungkus fungsionalitas diskret, didistribusikan, dan diakses secara programatik melalui protokol internet standart [6].

Dari berbagai definisi tersebut dapat disimpulkan bahwa pada dasarnya *web service* merupakan *middleware* internet yang memungkinkan berbagai sistem untuk saling berkomunikasi tanpa pengaruh pada *platform*. *Web service* membungkus operasi-operasi ke dalam sebuah antarmuka yang ditulis dalam notasi XML. Antarmuka ini menyembunyikan detail implementasi dari layanan. Pertukaran informasi yang terjadi dalam web service juga menggunakan pesan dalam format XML.

*Web service* dibangun dari tiga komponen utama, yaitu *service provider*, *service registry*, dan *service requestor* [3]. Komponen-komponen tersebut saling berinteraksi melalui komponen *web service*, yang berupa deskripsi dan implementasi layanan. Terdapat tiga macam operasi yang memungkinkan komponen-komponen tersebut untuk dapat saling berinteraksi, yaitu *publish*, *find*, dan *bind*.

Dalam skenario yang umum, *service provider* akan menempatkan sebuah modul perangkat lunak yang dapat diakses dalam sebuah jaringan, kemudian mendeskripsikan antarmuka yang sesuai dengan layanan tersebut dan mempublikasikannya secara langsung ke *service requestor* atau melalui *service registry*. Pihak yang hendak mengkonsumsi *web service* akan menjalankan operasi *find* untuk menemukan deskripsi layanan yang diinginkan tersebut untuk mengikatkan diri ke *service provider* dan berinteraksi dengan implementasi dari *web service*, keterkaitan antara peran, operasi, dan

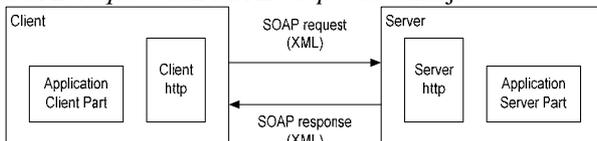
komponen *web service* dapat dilihat pada gambar 3 berikut ini :



Gambar 3. Komponen web service  
 (Sumber : Kreger, H [3])

## SOAP

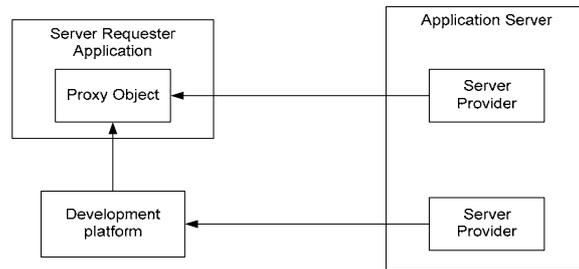
SOAP (*Simple Object Acces Protocol*) adalah merupakan protocol untuk pertukaran informasi dengan desentralisasi dan terdistribusi. SOAP merupakan gabungan antara HTTP dengan XML karena SOAP umumnya menggunakan protokol HTTP sebagai sarana *transport* datanya dan data akan dipertukarkan ditulis dalam format XML. SOAP mengatur bagaimana request dan respon dari suatu service bekerja. Gambar 4 merupakan proses SOAP request dan SOAP response bekerja :



Gambar 4. SOAP dengan standar XML

## WSDL

WSDL (*Web Service Description Language*) adalah merupakan sebuah bahasa berbasis XML yang digunakan untuk mendefinisikan *web service* dan menggambarkan bagaimana cara untuk mengakses *web service* tersebut. Fungsi utama WSDL dalam *web service* adalah untuk mengotomasi mekanisme komunikasi *business-to-business* dalam *web service* melalui protokol internet. Dengan menggunakan WSDL *client* dapat memanfaatkan fungsi-fungsi publik yang disediakan oleh server [5]. Diagram WSDL dapat disajikan pada gambar 5 berikut ini :



Gambar 5. Diagram Dokumen WSDL

## PHP NuSOAP

NuSOAP adalah sebuah kumpulan *class-class* PHP yang memungkinkan user untuk mengirim dan menerima pesan SOAP melalui protokol HTTP. NuSOAP didistribusikan oleh NuSphere Corporation (<http://www.nusphere.com>) sebagai *open source toolkit* di bawah lisensi GNU GPL.

Salah satu keuntungan dari NuSOAP adalah bahwa NuSOAP bukan merupakan PHP *extension*, sehingga penggunaannya tidak membutuhkan registrasi khusus ke sistem operasi maupun *web server*. NuSOAP ditulis dalam kode PHP murni sehingga semua *developer web* dapat menggunakan *tool* ini tanpa tergantung pada jenis *web server* yang digunakan.

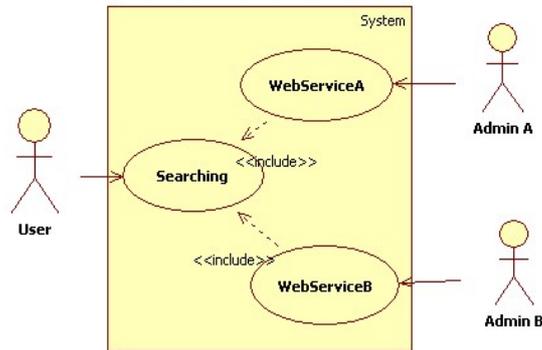
NuSOAP merupakan *toolkit web service* berbasis komponen. NuSOAP memiliki beberapa *class* yang menyediakan *method* seperti serialisasi variabel dan pemaketan *SOAP-Envelope*. Interaksi *web service* dilakukan dengan *class client* yang disebut dengan *class soapClient* dan *class server* yang disebut dengan *class soapServer*. *Class-class* ini mengizinkan *user* untuk melakukan proses pengiriman dan penerimaan pesan-pesan SOAP dengan bantuan beberapa *class-class* pendukung lainnya untuk melengkapi proses tersebut.

## 2. Analisa dan Rancangan

Pada penelitian ini dilakukan perancangan *web service server* dengan menggunakan *tool kit* PHP NuSOAP. Dibuat dua buah *prototype* toko buku online dengan basis data yang berbeda yaitu sistem A menggunakan database MySQL dan sistem B menggunakan database PostgreSQL. Masing-masing server menyediakan *web service* (*service A* dan *service B*). Dan dibuat sebuah aplikasi *client* untuk memanggil *service-service* yang disediakan oleh kedua server tersebut. Aplikasi bersifat satu arah, yaitu *client* melakukan *request* dan *server*

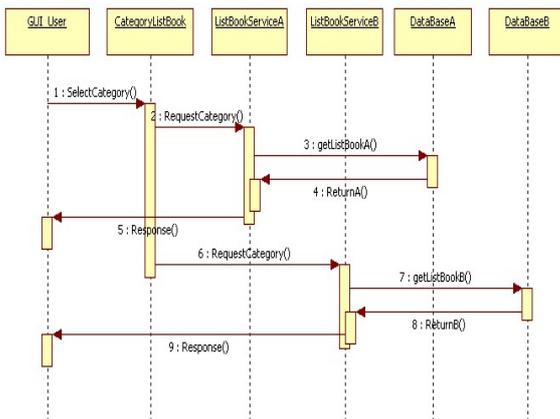
memberikan *responsenya*. Meskipun kedua server memiliki *platform database*, tetapi keduanya harus menyediakan standart *service* yang sama agar aplikasi *client* dapat mengakses parameter yang disediakan.

Rancangan dimulai dengan membuat diagram *use case*. Diagram *use case* menjelaskan fungsionalitas sistem dipandang dari perpektive *user* dan menggambarkan interaksi antara sistem dengan user [1]. Diagram *use case* pada *prototype web service* ini disajikan pada gambar 6 berikut ini :



Gambar 6. Use Case Diagram

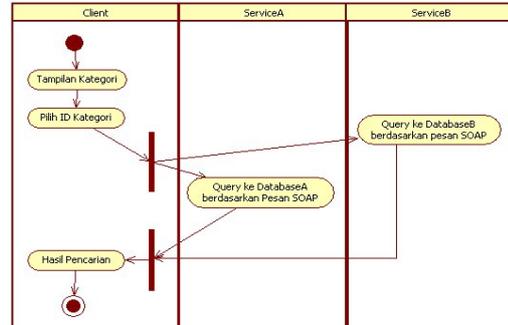
Perancangan berikutnya yaitu dengan membuat *sequence diagram*. *Sequence diagram* digunakan untuk menggambarkan perilaku pada sebuah skenario yang menunjukkan sejumlah obyek-obyek didalam *use case*. Berikut disajikan *sequence diagram* pada gambar 7 yang merupakan proses *searching* dari *client* ke *web service* pada dua buah server.



Gambar 7. Sequence Diagram

Langkah selanjutnya membuat *activity diagram*, *activity diagram* digunakan untuk mendeskripsikan logika *prosedural*, proses bisnis dan aliran kerja dalam pertukaran data. Pada *web service* katalog

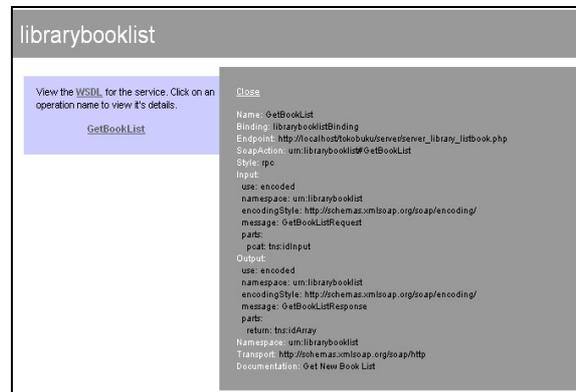
buku ini digunakan *activity diagram* seperti pada gambar 8 sebagai berikut :



Gambar 8. Activity Diagram

### 3. Implementasi

Berdasarkan hasil perancangan tersebut, dibuat *prototype* sistem pencarian katalog. Pada sisi server dibuat *web service getBookList* yang digunakan untuk membuat layanan yang dapat dimanfaatkan oleh client. Pada *service getBookList* mempunyai parameter berupa id kategori. Pada gambar 9 berikut ini adalah *service* yang dimiliki server :



Gambar 9. Service getBookList

Dengan menggunakan *class-class* yang ada pada *library NuSoap* file WSDL secara otomatis dibuat, gambar 10 berikut ini adalah file WSDL web service server :

```

<definitions targetNamespace="urn:librarybooklist">
  <types>
    <xsd:schema targetNamespace="urn:librarybooklist">
      <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
      <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
      <xsd:complexType name="idInput">
        <xsd:all>
          <xsd:element name="category" type="xsd:string" />
        </xsd:all>
      </xsd:complexType>
      <xsd:complexType name="idArray">
        <xsd:complexContent>
          <xsd:restriction base="SOAP-ENC:Array">
            <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="xsd:string[]" />
            <xsd:restriction>
              <xsd:complexContent>
                <xsd:complexType>
                  </xsd:complexType>
                </xsd:complexContent>
              </xsd:restriction>
            </xsd:restriction>
          </xsd:restriction>
        </xsd:complexContent>
      </xsd:complexType>
    </xsd:schema>
  </types>

```

Gambar 10. File WSDL

Tahap implementasi dimulai dengan membuat *service* pada *server*, proses pembuatan *service server* dijelaskan melalui tabel 1 berikut ini :

Tabel 1. Proses pembuatan *service*

1	Pemanggilan kelas NuSOAP <code>require_once('nusoap.php');</code>
2	Membuat server instance <code>\$server = new soap_server();</code>
3	Instalasi WSDL support <code>\$server-&gt;configureWSDL('librarybooklist', 'urn:librarybooklist');</code> <code>\$server-&gt;wsdl-&gt;schemaTargetNamespace = 'urn:librarybooklist';</code>
4	Register struktur data <code>\$server-&gt;register('GetBookList', array('pcat' =&gt; 'tns:idInput'), array('return' =&gt; 'tns:idArray'), 'urn:librarybooklist', 'urn:librarybooklist#GetBookList', 'rpc','encoded','Get New Book List');</code>
5	Mendeskripsikan fungsi <code>function GetBookList(\$pcat)</code> <code>//query ke database, hasilnya disimpan dalam array //</code>
6	Membuat http listener dan keluar <code>\$server-&gt;service(\$HTTP_RAW_POST_DATA);</code> <code>exit();</code>

Perbedaan kedua *service* adalah pada proses *query* ke *database* harus disesuaikan dengan fungsi dan tabel pada masing-masing *database*.

Kemudian dibuat aplikasi *client* yang mengakses *service* dan mengirimkan parameter. Serta membuat format untuk menampilkan output dalam bentuk HTML. Berikut ini pada gambar 11 disajikan proses pemanggilan salah satu *service* pada sisi *client* :

```

$pcat = array('category'=>$category); $client =
new nusoap_client(
'http://localhost/tokobuku/server/
server_library_listbook.php?wsdl', 'wsdl');

```

```

$proxy = $client->getProxy();
$result = $proxy->GetBookList($pcat);
$error = $proxy->getError();

```

Gambar 11. Pemanggilan Service

#### 4. Hasil Implementasi

Sistem pencarian katalog merupakan sistem berbasis web yang dibuat pada sisi *client*, *client* melakukan *request* ke *service server*. Dengan mengirimkan parameter berupa *id* kategori buku. Tampilan aplikasi pada sisi *client* disajikan pada gambar 12 berikut ini :

Gambar 12. Menu pilihan katalog

Proses pencarian dilakukan dengan memanggil *service* ke dua buah *server*, dan hasil yang diperoleh dikirim kesisi *client*. Hasil pencarian dapat disajikan pada gambar 13 berikut ini :

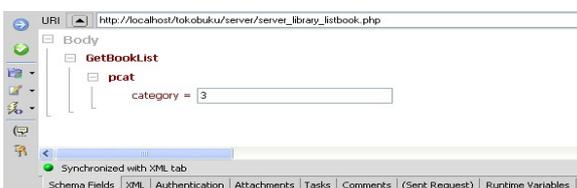
Gambar 13. Hasil Pencarian

#### 5. Pengujian

Prosedur pengujian melibatkan proses yang memanfaatkan keseluruhan layanan yang ada pada aplikasi *server web service*. Dengan demikian maka dapat dipastikan apakah setiap layanan yang ada

telah berjalan dengan baik dan sesuai dengan yang diharapkan.

Pengujian menggunakan *automation tool* SOAPSonar yang mempunyai kemampuan menguji *white box* melalui file WSDL yang merupakan model *web service* dari masing-masing sistem. Pengujian parameter dan fungsi request dan response menggunakan tool SOAPSonar. Langkah pengujian dimulai dengan meng-*capture* URL (*Uniform Resource Locator*) ke tool SOAPSonar. Kemudian dilakukan *test case* terhadap *service* yang dimiliki. Gambar 14 merupakan tampilan test case pada *service* *getBookList*.



Gambar 14. Input parameter

Dilakukan ujicoba dengan memasukkan parameter, misalkan *id* kategori dimasukkan nilai 3 atau kategori *programming* (pada menu *client*). Didapatkan pesan SOAP dalam format XML yang dikirim seperti pada gambar 15 sebagai berikut :



Gambar 15. Request SOAP

Kemudian dari sisi *server* memberikan SOAP *response* dalam format XML berupa *variabel array* yang berisi 4 buku yang tersedia sesuai dengan kategori, hasilnya dapat disajikan pada gambar 16 sebagai berikut :



Gambar 16. SOAP Response

Berdasarkan pengujian *service* pada *server* menunjukkan bahwa *service* berfungsi dengan baik, dengan memberikan *response* sesuai dengan parameter yang di masukkan.

## 6. Kesimpulan

Dalam makalah ini telah dijabarkan cara penggunaan teknologi *web service* untuk menunjang proses penggabungan data dari dua sistem toko buku online dengan *platform database* berbeda. Dengan adanya fleksibilitas *web service* diharapkan proses integrasi data antar web lebih mudah dilakukan dan lebih banyak data yang dapat dipertukarkan. Semakin banyak situs yang menyediakan potongan informasi atau *syndicated content*, secara tidak langsung akan membantu mempermudah *user* dalam mencari informasi di internet.

## Referensi

- [1] Bennett, Simon. Steve McRobb dan Ray Farmer. *Object-Oriented System Analysis and Design Using UML 3e*, New York, McGraw-Hill, 2006.
- [2] Jorgensen, David, *Developing .Net Web Service with XML*. Syngress, 2002.
- [3] Kreger, H. *Web Service Conceptual Architecture* (WSA 1.0). 2001 <http://www.cs.uoi.gr> (di akses terakhir 09 Mei 2010)
- [4] Short, Scott. *Buliding XML Web Service for the Microsoft .Net Platform*. Microsoft Press. 2002.
- [5] Siswoutomo, Wiwit. *Membangun Web Service Open Source Menggunakan PHP*. Jakarta, Elex Media Komputindo. 2004
- [6] The Stencil Group. *Defining Web Services*. 2001.
- [7] W3C. Web Service Architecture. <http://www.w3c/TR/ws-arch/> (di akses terakhir 09 Mei 2010)