
PERANCANGAN MODEL SISTEM INFORMASI BERBASIS WEB SERVICE DENGAN MEMANFAATKAN TEKNOLOGI XML-RPC (EXTENSIBLE MARKUP LANGUAGE – REMOTE PROCEDURE CALL)

Goldie Gunadi

0856-7889766

send2goldie@gmail.com, goldzmail@yahoo.com

Program Studi Teknik Informatika, Fakultas Teknik, Matematika dan IPA
Universitas Indraprasta PGRI

Abstract. *Currently, Web Service technology became very popular in the enterprise, one of the main reasons is because the technology is able to integrate different applications in a more simple platform and able to rectify the shortcomings of conventional middleware. The use of Web services more and more service done to the process of exchanging data or information through the Internet. In this technology, Web services using XML technology to exchange data. A number of methods currently used to provide Web services is SOAP, XML-RPC, and REST. XML-RPC technology itself is widely used because it has a simpler architecture than other techniques. A number of programming languages can be used to implement XML-RPC, which one of them is the PHP language. With PHP, we can create a Web Service that is used to manipulate data or perform an information stored in a database management system (Database Management System).*

Key Words: *Web Service, XML, XML-RPC, SOAP, PHP*

PENDAHULUAN

Web Service adalah komponen yang independen terhadap platform ataupun bahasa. *Web Service* menggunakan web protokol (HTTP) yang sangat mendukung heterogenitas dan interoperabilitas serta memudahkan integrasi. Selain itu *Web Service* mendukung koneksi *loosely coupled*, sehingga sebuah perubahan pada satu aplikasi tidak akan memaksa perubahan pada aplikasi yang lain. Sebuah *Web Service* memiliki *interface* berupa web API (*Application Programming Interface*) yang dapat dipanggil oleh suatu aplikasi untuk mengakses aplikasi yang mengimplementasikan layanan *Web Service*.

Salah satu teknologi yang dapat digunakan untuk mengimplementasikan *Web Service* adalah XML-RPC. XML-RPC adalah akronim dari *eXtensible Markup Language - Remote Procedure Call*. Sebuah spesifikasi XML yang menjelaskan mengenai mekanisme

pemanggilan prosedur jarak jauh dengan menggunakan XML. Bisa dikatakan, XML-RPC adalah salah satu bentuk *Web Service* yang disederhanakan dari standar yang konvensional. Dua sistem yang benar-benar terpisah dan berbeda platform serta lingkungan bisa saling berkomunikasi lewat sarana file XML.

Protokol komunikasi yang digunakan adalah protokol standar yang paling populer di dunia: HTTP. *Request* yang dikirim lewat HTTP harus menggunakan metode POST. Prosedur yang akan dipanggil beserta parameternya dibungkus dalam file XML dalam spesifikasi XMLRPC. Demikian pula *return value*-nya, sebelum dikirim akan dibungkus dulu dalam bentuk XML dan ditransfer diatas lalu lintas protokol HTTP di internet.

Saat ini, beberapa layanan populer telah mendukung XML-RPC. Yang paling mencolok tentu saja adalah layanan blogging. Kini setiap mesin blog saling berkomunikasi satu sama

lain (trackback, pingback) dengan menggunakan standar XML-RPC. Dengan XML-RPC ini, beberapa layanan baru untuk mengelola *blog* di luar *software* standar muncul. Contohnya adalah Wordpress API dan Blogger API.

Pada tulisan ini penulis mencoba untuk mengimplementasikan teknologi XML-RPC untuk membuat sebuah model sistem informasi berbasis *Web Service* dengan menggunakan sebuah bahasa pemrograman web, yakni PHP dan sebuah sistem manajemen basis data (DBMS) MySQL. Model sistem informasi yang dibangun dalam tulisan ini diharapkan dapat menjadi dasar referensi untuk pengembangan sebuah sistem informasi berbasis *Web Service* terintegrasi yang lebih besar dan kompleks.

TINJAUAN PUSTAKA

Pemanfaatan XML pada Teknologi Web Service

Teknologi *Web Service* berhubungan dengan proses pertukaran data antara *server* dan *client*, dengan menggunakan format XML standar untuk "pemaketan" permintaan dan data sehingga kedua sistem dapat saling "mengerti" satu sama lain. Baik *server* maupun *client*, keduanya bisa berupa *server web*, atau pun perangkat elektronik lainnya.

Proses pertukaran data yang terjadi dalam *Web Service* biasanya terjadi melalui TCP *port* 80, menggunakan protokol HTTP standar POST. Dengan kata lain, Web Services beroperasi menggunakan cara yang sama dengan apa yang dilakukan oleh browser Anda, yakni ketika melakukan proses POST sebuah form HTML ke sebuah situs, dan kemudian menerima kembali sebuah halaman *web* sebagai jawaban. Satu-satunya yang membedakan adalah *Web Service* tidak menggunakan HTML melainkan menggunakan XML. Dan ini berarti

Web Service dapat tersedia di mana saja di Internet, melewati firewall seperti halnya menampilkan halaman *web*. Pertukaran data terjadi pada *packaging layer*.

Sejumlah metode yang saat ini banyak digunakan untuk menyediakan *Web Service* adalah SOAP, XML-RPC, dan REST.

a. SOAP

SOAP adalah akronim dari *Simple Object Access Protocol* namun akronim ini dibuang pada SOAP versi 1,2. SOAP adalah metode untuk bertukar pesan berbasis XML melalui Internet untuk menyediakan dan memanfaatkan *Web Service*. Pesan SOAP ditransfer membentuk SOAP-*Envelope*.

b. XML-RPC

XML-RPC (remote prosedur call) merupakan alternatif lain yang dapat digunakan sehubungan dengan penyediaan dan pemanfaatan *Web Service*. Metode ini menggunakan XML untuk mengkodekan dan *decode* pemanggilan prosedur *remote (remote procedure call)* bersama dengan parameternya. Dibandingkan dengan arsitektur SOAP, XML-RPC memiliki arsitektur yang lebih sederhana. Setiap tipe data parameter dari prosedur dalam XML-RPC dapat kita definisikan.

c. REST

Representational State Transfer (REST) adalah metode yang relatif sederhana untuk menyediakan dan memanfaatkan *Web Service*. Saat ini, metode ini menjadi populer di lingkungan *Web Service*. Pada metode ini tidak diperlukan penggunaan XML sebagai format pertukaran. Arsitektur REST pada dasarnya terfokus pada dua hal: Sumber daya (*resources*) dan antar-muka

(Interface). RESTful adalah istilah lain untuk mendefinisikan REST Web Service.

Resources (sumber daya) merupakan state dan fungsionalitas aplikasi yang diwakili oleh sebuah URL unik. Masing-masing sumber daya berbagi antarmuka yang seragam untuk mentransfer state antara client dan server.

Layanan Teknologi Web Service

Di atas pertukaran data, Anda juga memerlukan informasi yang menggambarkan antarmuka (atau Application Program Interface - API) ke layanan. Hal ini membuat Web Service berguna untuk seluruh Internet, yang memungkinkan pengembang lain untuk mengembangkan program-program yang dapat mengakses Web Service. Ini disebut description layer, dan yang memungkinkan semuanya ini adalah standar WSDL (Web Service Description Language).

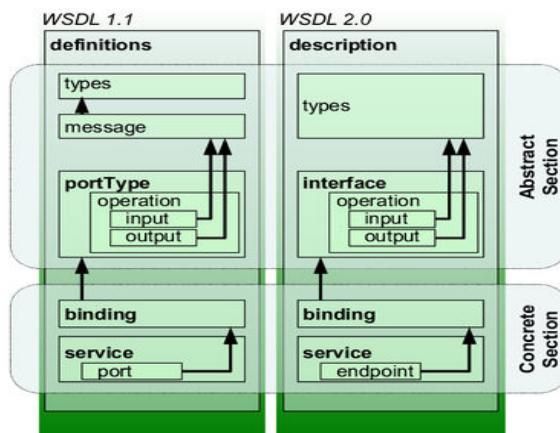
Diatasnya, ada informasi yang menjelaskan sifat dari layanan itu sendiri (tidak seperti HTML-descriptive META tag), sehingga dapat dilakukan proses kategorisasi dan ditemukan pada situs yang menawarkan direktori-direktori Web Service. Ini adalah discovery layer, yang diatur oleh standar UDDI (Universal Description, Discovery dan Integration).

Baik description layer maupun discovery layer hanyalah merupakan XML, yang disajikan menggunakan format tertentu sehingga memungkinkan informasi yang relevan dapat ditemukan untuk semua Web Service di Internet.

WSDL (Web Service Description Language)

File WSDL adalah sebuah dokumen XML yang mendeskripsikan suatu rangkaian pesan SOAP dan bagaimana proses pertukaran masing-masing pesan. WSDL adalah XML sehingga dapat dibaca dan diedit tetapi dalam banyak kasus, WSDL dihasilkan dan dikonsumsi oleh perangkat lunak.

Notasi yang digunakan oleh file WSDL untuk menggambarkan format pesan didasarkan pada standar XML Schema yang artinya kedua bahasa pemrograman, baik netral maupun yang berbasis standar, cocok untuk menggambarkan antar muka XML Web Service yang dapat diakses oleh berbagai platform dan bahasa pemrograman. Selain menggambarkan isi pesan, WSDL mendefinisikan di mana layanan tersedia dan protokol komunikasi apa yang digunakan untuk berbicara dengan layanan tersebut. Ini berarti bahwa file WSDL mendefinisikan segala sesuatu yang diperlukan untuk menulis sebuah program yang bekerja dengan XML Web Service.



Gambar 1. Representasi Konsep berdasarkan dokumen WSDL 1.1 dan WSDL 2.0

- a. *Layanan (Service)*
Layanan ini dapat dianggap sebagai wadah untuk satu set fungsi sistem yang telah dipublikasikan oleh protokol berbasis web.
- b. *Port / Endpoint*
Port digunakan untuk mendefinisikan alamat atau *connection point* ke sebuah *Web Service*. Hal ini biasanya diwakili oleh sebuah string url http sederhana.
- c. *Binding*
Menentukan antarmuka (*interface*), mendefinisikan *SOAP binding style* (*RPC / Document*) dan transportasi (*SOAP Protocol*). Bagian ini juga mendefinisikan operasi.
- d. *PortType / Interface*
Elemen *<PortType>*, yang telah diubah namanya menjadi *<interface>* di WSDL 2.0, mendefinisikan layanan Web, operasi yang dapat dilakukan, dan pesan-pesan yang digunakan untuk melakukan operasi.
- e. *Operasi (Operation)*
Setiap operasi serupa dengan proses pemanggilan metode atau fungsi dalam bahasa pemrograman tradisional. Di sini *SOAP actions* didefinisikan dan bagaimana cara pesan dikodekan.
- f. *Pesan (Message)*
Pesan berisi informasi yang dibutuhkan untuk melakukan operasi. Setiap pesan terdiri dari satu atau lebih bagian-bagian logikal. Setiap bagian dihubungkan dengan atribut-atribut pengetikan pesan. Atribut nama pesan memberikan nama yang unik bagi masing-masing pesan. Atribut nama *part* memberikan nama yang unik untuk masing-masing *part* dari pesan yang disertakan. *Part* merupakan deskripsi dari *logical*

content sebuah pesan. Dalam *RPC binding*, sebuah *binding* bisa merupakan referensi dari nama sebuah *part* dalam rangka untuk menentukan informasi *binding* spesifik mengenai *part* tersebut. Sebuah *part* dapat mewakili sebuah parameter dalam pesan, *binding* menentukan makna sebenarnya dari *part* tersebut. Pesan sudah dihapus di WSDL 2.0, di mana langsung merujuk pada tipe skema XML untuk mendefinisikan bagian dari setiap masukan (*input*), keluaran (*output*) dan kesalahan (*fault*).

- g. *Tipe (Types)*
Tujuan dari tipe dalam WSDL adalah untuk menggambarkan data. Skema XML digunakan (*inline* atau *referenced*) untuk tujuan ini.

UDDI (*Universal Description, Discovery and Integration*)

Di atas pertukaran data, Anda juga memerlukan informasi yang menggambarkan antarmuka (atau *Application Program Interface - API*) ke layanan. Hal ini membuat *Web Service* berguna untuk seluruh Internet, yang memungkinkan pengembang lain untuk mengembangkan program-program yang dapat mengakses *Web Service*. Ini disebut *description layer*, dan yang memungkinkan semuanya ini adalah standar WSDL (*Web Service Description Language*).

Diatasnya, ada informasi yang menjelaskan sifat dari layanan itu sendiri (tidak seperti *HTML-descriptive META tag*), sehingga dapat dilakukan proses kategorisasi dan ditemukan pada situs yang menawarkan direktori-direktori *Web Service*. Ini adalah *discovery layer*, yang diatur oleh standar UDDI (*Universal Description, Discovery dan Integration*).

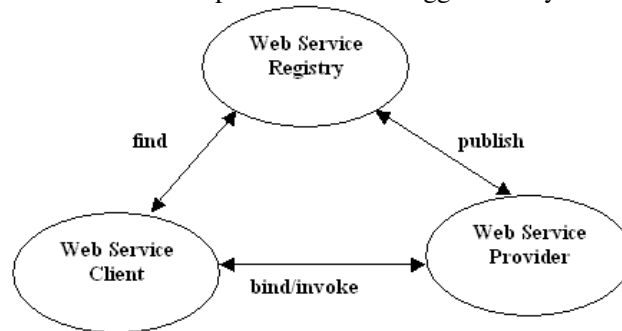
Baik *description layer* maupun *discovery layer* hanyalah merupakan

XML, yang disajikan menggunakan format tertentu sehingga memungkinkan informasi yang relevan dapat ditemukan untuk semua *Web Service* di Internet.

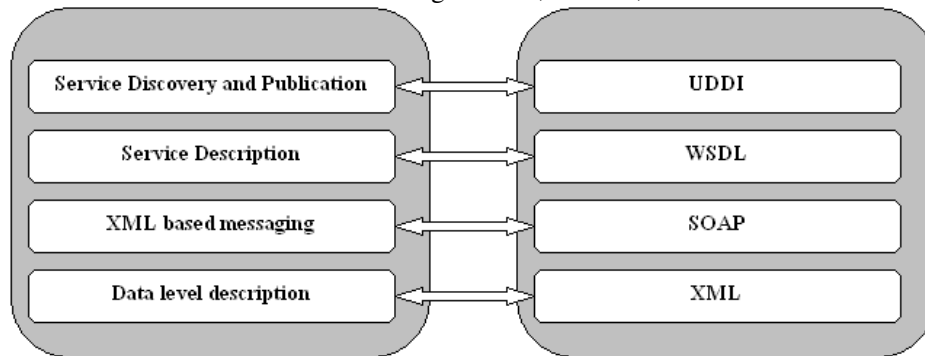
Model Aplikasi *Web Service*

Model *Web Service* mengikuti paradigma *publish, find, dan bind*. Pada langkah pertama (*publish*), menyajikan sebuah *Web Service* yang terdapat didalam registri *Web Service*. Kedua (*find*), *client* melakukan proses

pencarian layanan untuk memenuhi kebutuhan mereka ke dalam registri. Setelah berhasil menemukan sejumlah layanan yang sesuai maka akan dilakukan pemilihan layanan. *Client* memilih layanan berdasarkan preferensi. Ketiga (*bind*), *client* kemudian melakukan *download* deskripsi layanan dan membuat ikatan dengan layanan tersebut untuk memanggil dan menggunakannya.



Gambar 2. Paradigma *Find, Publish, Bind*



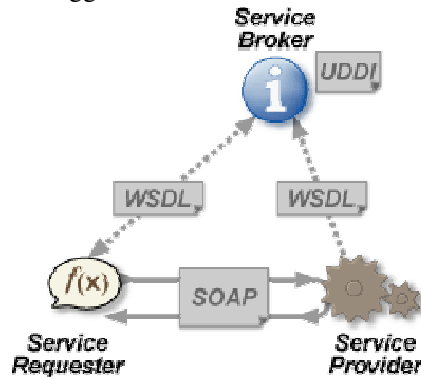
Gambar 3. Lapisan Aplikasi berbasis *Web Service*

Pada lapisan paling bawah adalah standar XML yang melakukan pengalamatan. SOAP (*Simple Object Access Protocol*) adalah sebuah mekanisme berbasis XML untuk pengolahan pesan dan RPC (*Remote Procedure Calls*). Ini mengatur masalah mendasar dari *firewall traversal* yang terjadi dalam sistem RPC dengan menggunakan HTTP sebagai transportasi. SOAP adalah protokol yang digunakan untuk mengajukan permintaan layanan.

WSDL (*Web Services Description Language*) menyediakan sebuah metode berbasis XML untuk menggambarkan Web Service serta memberikan rincian penggunaannya. WSDL merupakan bentuk XML setara dengan IDL (*Interface Definition Language*), yang digunakan dalam implementasi RPC. UDDI (*Universal Description Discovery Integrasi*) menyediakan sebuah direktori "yellow page" dari *Web Service*, sehingga memudahkan bagi *client* untuk menemukan layanan yang diperlukan. Penyedia layanan menyediakan

deskripsi layanan (WSDL) dan rincian lainnya yang dapat dicari pada registri UDDI. *Client* akan menggunakan

UDDI untuk melakukan pencarian layanan.

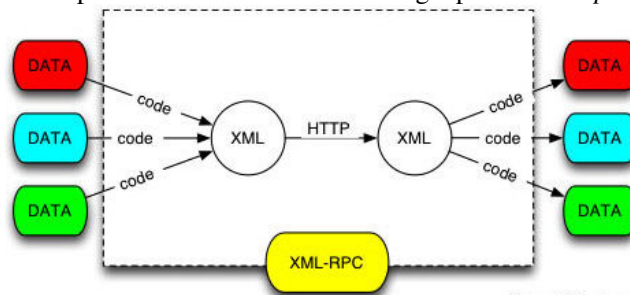


Gambar 4. Arsitektur Web Service

HASIL DAN PEMBAHASAN XML-RPC

XML-RPC bekerja dengan mengirimkan permintaan HTTP ke *server* yang mengimplementasikan protokol. *Client* dalam kasus ini adalah perangkat lunak yang melakukan pemanggilan metode yang terdapat pada sebuah *remote system*. Beberapa parameter masukan dapat dikirimkan ke

remote method, dan kemudian akan dikembalikan sebuah *return value*. Tipe parameter memungkinkan pengulangan sejumlah parameter ke dalam bentuk peta (*maps*) dan daftar (*lists*), sehingga struktur yang lebih besar dapat dikirimkan. Oleh karena itu XML-RPC dapat digunakan untuk mengirimkan sejumlah objek ataupun struktur baik sebagai parameter *input* maupun *output*.



Source: JY Stervinaou

Gambar 5. Model XML-RPC

Identifikasi setiap *client* untuk keperluan otorisasi dapat dicapai dengan menggunakan metode keamanan HTTP populer. Otentikasi akses dasar (*basic access authentication*) digunakan untuk melakukan identifikasi, HTTPS digunakan ketika identifikasi (melalui sertifikat) dan pesan terenkripsi diperlukan. Pada implemetasinya kedua metode tersebut dapat dikombinasikan.

XML-RPC lebih sederhana untuk digunakan dan dimengerti daripada

SOAP karena hanya memungkinkan satu metode dari serangkaian metode, sedangkan SOAP mendefinisikan beberapa *encoding* yang berbeda. Disamping itu model keamanan yang digunakan lebih sederhana, dan XML-RPC tidak mendukung (atau memerlukan) pembuatan deskripsi layanan WSDL. JSON-RPC adalah serupa dengan XML-RPC.

Tipe Data

Tipe Data yang digunakan pada tabel berikut:
metode XML-RPC dapat dilihat pada

Tabel 1. Tipe Data

Nama	Contoh Tag	Deskripsi
array	<pre><array> <data> <value><i4>1404</i4></value> <value><string>Something here</string></value> <value><i4>1</i4></value> </data> </array></pre>	Array dari sejumlah nilai, tidak menyimpan key
base64	<pre><base64>eW91IGNhbid0IHJlYWQgdGhpcyE=</base64></pre>	Data binary Base64- <i>encoded</i>
boolean	<pre><boolean>1</boolean></pre>	Nilai logika Boolean (0 atau 1)
date/time	<pre><dateTime.iso8601>19980717T14:08:55</dateTime.iso8601></pre>	Tanggal dan Jam dalam format ISO 8601
double	<pre><double>-12.53</double></pre>	Angka dengan format <i>floating point</i> presisi ganda Semua nilai <i>integer</i>
integer	<pre><i4>42</i4> atau <int>42</int></pre>	<i>String</i> dari sejumlah karakter, mengikuti aturan <i>XML encoding</i>
string	<pre><string>Hello world!</string></pre>	Array asosiatif
struct	<pre><struct> <student> <id_num><string>210511001</string></id_num> <name>Jonathan Rivaldo</name> <credit><i4>25</i4></credit> </student> <student> <id_num><string>210511002</string></ id_num> <name>Viona Dahlia</name> <credit><i4>40</i4></credit> </student> </struct></pre>	Nilai Null
nil	<pre><nil/></pre>	Nilai Null

Model Komunikasi XML-RPC

Komunikasi yang terjadi antar 2 sistem pada teknologi XML-RPC, dimulai dengan dengan proses *request* (permintaan) dari XML-RPC *client*, yang kemudian akan diberikan *response* (tanggapan) oleh *server*. Permintaan

(*request*) dapat berupa sebuah metode atau sejumlah parameter yang diperlukan oleh metode. Tanggapan (*response*) berupa balasan dengan sejumlah parameter yang berisi data yang diminta.

XML-RPC Request

Contoh bentuk XML-RPC Request:

```
<?xml version="1.0"?>
<methodCall>
  <methodName>examples.getStudentName</methodName>
  <params>
    <param>
      <id_num><string>210511002</string></id_num>
    </param>
  </params>
</methodCall>
```

XML-RPC Response

Contoh bentuk XML-RPC Response:

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><string>Viona Dahlia</string></value>
    </param>
  </params>
</methodResponse>
```

XML-RPC Fault

Contoh bentuk XML-RPC Fault:

```
<?xml version="1.0"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value><int>4</int></value>
        </member>
        <member>
          <name>faultString</name>
          <value><string>Too many parameters.</string></value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>
```


IMPLEMENTASI XML-RPC DENGAN BAHASA PEMROGRAMAN PHP DAN DATABASE MySQL

Rancangan Tabel untuk Implementasi XML-RPC

Berikut adalah rancangan tabel yang dibuat berdasarkan sistem database MySQL:

Tabel 2. Struktur Tabel: tbl_bukutamu

Nama Tabel	tbl_bukutamu			
Nama Field	Tipe Data	Ukuran	Boleh NULL	Atribut
id	integer		Tidak	Primary Key, Auto Increment
nama	varchar	50	Tidak	
email	varchar	25	Tidak	
website	varchar	25	Ya	
telepon	varchar	20	Ya	
komentar	varchar	500	Tidak	
tgljam	datetime		Tidak	
publikasi	char	1	Tidak	

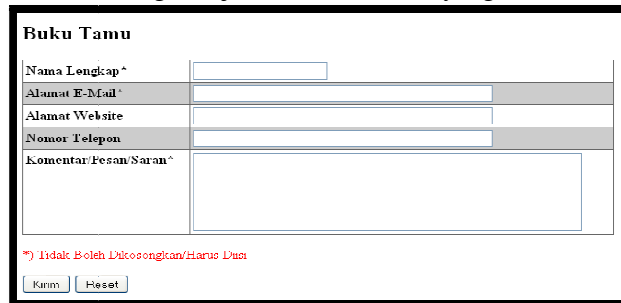
Perintah DDL-SQL (*Data Definition Language-SQL*) dengan MySQL:

```
CREATE TABLE `tbl_bukutamu` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `nama` varchar(50) NOT NULL,
  `email` varchar(25) NOT NULL,
  `website` varchar(25) DEFAULT NULL,
  `telepon` varchar(20) DEFAULT NULL,
  `komentar` varchar(500) NOT NULL,
  `tgljam` datetime NOT NULL,
  `publikasi` char(1) NOT NULL DEFAULT 'T',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Rancangan Tampilan Modul




Modul aplikasi terdiri dari 2 buah modul:

1. Modul Entri Buku Tamu
Menyediakan fasilitas bagi pengguna untuk melakukan proses entri buku tamu melalui sebuah form dengan sejumlah kolom isian yang telah ditentukan.



Gambar 6. Tampilan Form Buku Tamu

2. Modul Tampilkan Daftar Isi Buku Tamu
Menampilkan daftar informasi buku tamu yang telah diinputkan sebelumnya oleh pengguna.

Daftar Isi Buku Tamu				
No	Tanggal Jam	Nama Lengkap	Alamat E-Mail	Lihat Detil
1	1 Jan10 09:50	Goldie Gunadi	send2goldie@gmail.com	
2	6 Jan10 14:30	Salvino Gavrilla A.	sg_arnoldie@yahoo.co.id	
3	10 Jan 10 15:00	Cindy Lamusu	cylamusu@gmail.com	

Gambar 7. Tampilan Daftar Isi Buku Tamu

3. Modul Tampilkan Isi Buku Tamu
Menampilkan isi buku tamu yang dipilih oleh pengguna dari daftar isi buku tamu.

Tanggal : 1 Jan 2010, Jam : 09:50	
Nama Lengkap	Goldie Gunadi
Alamat E-Mail	send2goldie@gmail.com
Alamat Website	goldie.byethost18.com
Nomor Telepon	999888777666
Komentar/Pesan /Saran	Everyone knows PHP is the best scripting language for the web, but sadly some ASP coders still haven't worked it out yet.

Gambar 8. Tampilan Isi Buku Tamu

Daftar Fungsi XML-RPC pada PHP

Tabel 3. Daftar Fungsi XML-RPC

Nama Fungsi	Kegunaan
xmlrpc_decode_request	Melakukan decode (<i>decode</i>) XML menjadi bentuk tipe <i>native</i> PHP Deskripsi: <i>mixed xmlrpc_decode_request (string \$xml , string &\$method [, string \$encoding])</i>
xmlrpc_decode	Melakukan decode (<i>decode</i>) XML menjadi bentuk tipe <i>native</i> PHP Deskripsi: <i>mixed xmlrpc_decode (string \$xml [, string \$encoding = "iso-8859-1"])</i>
xmlrpc_encode_request	Membuat XML untuk setiap <i>method request</i> Deskripsi: <i>string xmlrpc_encode_request (string \$method , mixed \$params [, array \$output_options])</i>
xmlrpc_encode	Membuat XML untuk setiap nilai PHP Deskripsi: <i>string xmlrpc_encode (mixed \$value)</i>
xmlrpc_get_type	Mendapatkan tipe xmlrpc untuk setiap nilai PHP Deskripsi: <i>string xmlrpc_get_type (mixed \$value)</i>

xmlrpc_is_fault	Menentukan apakah sebuah nilai array merepresentasikan sebuah XMLRPC <i>fault</i> Deskripsi: <i>bool xmlrpc_is_fault (array \$arg)</i>
xmlrpc_parse_method_descriptions	Melakukan dekode XML menjadi sebuah daftar deskripsi metode Deskripsi: <i>array xmlrpc_parse_method_descriptions (string \$xml)</i>
xmlrpc_server_add_introspection_data	Menambahkan <i>introspection documentation</i> Deskripsi: <i>int xmlrpc_server_add_introspection_data (resource \$server , array \$desc)</i>
xmlrpc_server_call_method	Melakukan teknik parsing terhadap XML <i>requests</i> dan pemanggilan metode Deskripsi: <i>string xmlrpc_server_call_method (resource \$server , string \$xml , mixed \$user_data [, array \$output_options])</i>
xmlrpc_server_create	Membuat sebuah <i>xmlrpc server</i> Deskripsi: <i>resource xmlrpc_server_create (void)</i>
xmlrpc_server_destroy	Menghapus sumber daya <i>server</i> Deskripsi: <i>int xmlrpc_server_destroy (resource \$server)</i>
xmlrpc_server_register_introspection_callback	Mendaftarkan sebuah fungsi PHP untuk membuat dokumentasi Deskripsi: <i>bool xmlrpc_server_register_introspection_callback (resource \$server , string \$function)</i>
xmlrpc_server_register_method	Melakukan registrasi sebuah fungsi PHP function untuk memastikan <i>resource method</i> sesuai dengan <i>method_name</i> Deskripsi: <i>bool xmlrpc_server_register_method (resource \$server , string \$method_name , string \$function)</i>
xmlrpc_set_type	Menentukan tipe xmlrpc, base64 atau datetime, untuk sebuah nilai <i>string</i> PHP Deskripsi: <i>bool xmlrpc_set_type (string &\$value , string \$type)</i>

Server Code

Untuk kebutuhan aplikasi pada sisi *server* terdapat 3 buah fungsi:

1. Nama Fungsi : *gb_getGuestBookList ()*
Tujuan : Mengambil semua data buku tamu dari dalam tabel: *tbl_bukutamu* yang boleh dipublikasikan.

Kode Program :

```
function gb_getGuestBookList(){
    /* Membuat Query*/
    $query = "SELECT * FROM tbl_bukutamu ORDER BY tgltjam DESC where publikasi = 'Y'";
    $sql = mysql_query( $query );
    if( $sql ){
```

```

$guestbook_items = array();
while ( $result = mysql_fetch_array( $sql ) ) {

    /*Menampung isi masing-masing kolom dari setiap baris data ke dalam variabel */
    $guestbook_item['id'] = $result['id'];
    $guestbook_item['nama'] = $result['nama'];
    $guestbook_item['email'] = $result['email'];
    $guestbook_item['website'] = $result['website'];
    $guestbook_item['telepon'] = $result['telepon'];
    $guestbook_item['komentar'] = $result['komentar'];
    $guestbook_item['tgljam'] = XMLRPC_convert_timestamp_to_iso8601(
        mysql_datetime_to_timestamp( $result['tgljam'] ) );

    /*Tambahkan isi variabel array $guestbook_items dengan isi variabel $guestbook_item */
    $guestbook_items[] = $guestbook_item;
}

```

/ Konversi nilai array \$guestbook_items menjadi suatu set parameter XML-RPC kemudian memberikan tanggapan dengan XML */*

```

XMLRPC_response(XMLRPC_prepare($guestbook_items), KD_XMLRPC_USERAGENT);
} else {
    /* Apabila terjadi kesalahan (error), tanggapilah dengan pesan kesalahan (error message) */
    XMLRPC_error("I", "gb_getGuestBookList() error: Unable to read guest book:"
        .mysql_error(). "\nQuery was: ". $query, KD_XMLRPC_USERAGENT);
}
}

```

2. Nama Fungsi : *gb_viewGuestBookItem (\$guestbook_id)*
 Tujuan : Mengambil data buku tamu dari dalam tabel: *tbl_bukutamu* dengan id sesuai dengan parameter yang diberikan pada saat pemanggilan fungsi.

Kode Program :
function gb_viewGuestBookItem (\$guestbook_id) {

```

    /*Definisikan query untuk mengambil item buku tamu */
    $query = "SELECT * FROM tbl_bukutamu WHERE id = ". $guestbook_id. "''";
    $sql = mysql_query( $query );

```

```

if( $result = mysql_fetch_array( $sql ) ) {

```

```

    /*Menampung isi masing-masing kolom dari setiap baris data ke dalam variabel */
    $guestbook_item['id'] = $result['id'];
    $guestbook_item['nama'] = $result['nama'];
    $guestbook_item['email'] = $result['email'];
    $guestbook_item['website'] = $result['website'];
    $guestbook_item['telepon'] = $result['telepon'];
    $guestbook_item['komentar'] = $result['komentar'];
    $guestbook_item['tgljam'] = XMLRPC_convert_timestamp_to_iso8601(
        mysql_datetime_to_timestamp( $result['tgljam'] ) );

```

```

    /* Memberikan tanggapan ke client berupa item buku tamu */
    XMLRPC_response(XMLRPC_prepare($guestbook_item), KD_XMLRPC_USERAGENT);

```

```

} else {
    /* Apabila terjadi kesalahan (error), tanggap dengan pesan kesalahan */
    XMLRPC_error("I", "gb_viewGuestBookItem error: Unable to read guest book:"
        .mysql_error(), KD_XMLRPC_USERAGENT);
}
}

```

3. Nama Fungsi : *gb_addGuestBookItem* (*\$guestbook_item=0*)
 Tujuan : Mengambil data buku tamu dari dalam tabel: *tbl_bukutamu* dengan id sesuai dengan parameter yang diberikan pada saat pemanggilan fungsi.

```

Kode Program :
function gb_addGuestBookItem ( $guestbook_item=0 ) {
    /* Memeriksa apakah array $guestbook_item telah diisi dengan nilai yang akan ditambahkan ke dalam
    tabel */
    if ( ( ISSET( $guestbook_item['nama'] ) ) && ( ISSET( $guestbook_item['email'] ) )
    && ( ISSET( $guestbook_item['website'] ) ) && ( ISSET( $guestbook_item['telepon'] ) )
    && ( ISSET( $guestbook_item['komentar'] ) ) ) {
        $nama = $guestbook_item['nama'];
        $email = $guestbook_item['email'];
        $website = $guestbook_item['website'];
        $telepon = $guestbook_item['telepon'];
        $komentar = $guestbook_item['komentar'];

        /* Query SQL untuk menambahkan item buku tamu ke dalam tabel: tbl_bukutamu */
        $query = "INSERT INTO tbl_bukutamu( nama, email, website, telepon, komentar, tgljam ) VALUES(
        '$nama', '$email', '$website', '$telepon', '$komentar', now())";
        $sql = mysql_query( $query );
        if ( $sql ) {
            $result = 1;
            /* Memberikan tanggapan ke client: penambahan item buku tamu sukses */
            XMLRPC_response( XMLRPC_prepare( $result ), KD_XMLRPC_USERAGENT );
        } else {
            /* Tanggapan pada saat terjadi kesalahan penambahan data */
            XMLRPC_error( "I", "gb_addGuestBookItem error: Unable to add Guest Book item:"
                .mysql_error(). "\nQuery was: ". $query, KD_XMLRPC_USERAGENT );
        }
    } else {
        $result = 0;
        /* Memberikan tanggapan ke client: penambahan item buku tamu batal */
        XMLRPC_response( XMLRPC_prepare( $result ), KD_XMLRPC_USERAGENT );
    }
}

```

Kode program untuk mendefinisikan sebuah array untuk masing-masing nama XML-RPC *methods* dan korespondensi dari masing-masing metode tersebut terhadap fungsi PHP-nya.

```

$xmlrpc_methods = array();
$xmlrpc_methods['guestbook.getGuestBookList'] = gb_getGuestBookList;
$xmlrpc_methods['guestbook.viewGuestBookItem'] = gb_viewGuestBookItem;
$xmlrpc_methods['guestbook.addGuestBookItem'] = gb_addGuestBookItem;

```

Client Code

Proses pemanggilan fungsi pada sisi client:

1. Pemanggilan fungsi *gb_getGuestBookList*:

Kode Program:

```
list($success, $response) = XMLRPC_request(
    $site, $location, 'guestbook.getGuestBookList', array(XMLRPC_prepare(null),
    'HarryFsXMLRPCClient')
```

2. Pemanggilan fungsi *gb_viewGuestBookItem*:

Kode Program:

```
list($success, $response) = XMLRPC_request($site, $location, 'guestbook.viewGuestBookItem',
    array(XMLRPC_prepare($_GET['guestbook_id']), 'HarryFsXMLRPCClient')
```

3. Pemanggilan fungsi *gb_addGuestBookItem*:

Kode Program:

```
list($success, $response) = XMLRPC_request($site, $location, 'guestbook.addGuestBookItem',
    array(XMLRPC_prepare($_POST['guestbook_item'],
    'HarryFsXMLRPCClient')
```

PENUTUP

1. Pemanfaatan teknologi *Web Service* dapat memberikan solusi terhadap sejumlah permasalahan berikut:

- a. Interoperabilitas (*Interoperability*)

Sebelumnya, pada sistem terdistribusi terdapat masalah interoperabilitas karena masing-masing *vendor* mengimplementasikan format masing-masing untuk objek yang terdistribusi dalam pengolahan pesan yang dikirimkan melalui jaringan. Dengan menggunakan XML sebagai sebuah standar pengiriman melalui jaringan, maka dua buah vendor seperti halnya Java/J2EE dan .NET / C # sekarang bisa berbicara satu sama lain.

- b. *Firewall traversal*

Proses untuk melakukan kolaborasi di seluruh perusahaan adalah sebuah masalah karena sistem terdistribusi seperti CORBA dan DCOM menggunakan port non-standar. Akibatnya,

kolaborasi berarti melubangi *firewall* anda, yang sering tidak dapat diterima oleh bagian TI. Oleh karena itu, hal ini tidak mengizinkan kolaborasi dinamis, akibatnya kolaborasi dilakukan melalui proses manual untuk bekerjasama dengan mitra. *Web Service* menggunakan HTTP sebagai protokol transportasi dan sebagian besar firewall membolehkan akses port 80 (HTTP), sehingga kolaborasi dapat dilakukan dengan lebih mudah dan dinamis. Sifat dinamis dari interaksi *Web Service* menawarkan beberapa layanan yang menarik bagi pengguna.

2. Implementasi *Web Service* dengan teknologi XML-RPC lebih sederhana untuk digunakan dan dimengerti daripada SOAP karena hanya memungkinkan satu metode dari serangkaian metode, sedangkan SOAP mendefinisikan beberapa *encoding* yang berbeda. Disamping itu model keamanan yang digunakan lebih sederhana, dan XML-RPC tidak mendukung

(atau memerlukan) pembuatan deskripsi layanan WSDL.

DAFTAR PUSTAKA

- Simon St. Laurent, Joe Johnston, Edd Dumbill. Juni 2001. **Programming Web Services with XML-RPC**. O'Reilly. First Edition.
- Benson Wong, 29 November 2001. **XML-RPC Client**. <http://devzone.zend.com/article/1307>.
- Harry Fuecks. 16 Juli 2002. **Build your own Web Service with PHP and XML-RPC**. <http://articles.sitepoint.com/article/own-web-service-php-xml-rpc>.
- Roger Wolter. Desember 2001. **XML Web Services Basics**. <http://msdn.microsoft.com/en-us/library/ms996507.aspx>.
- Roy Daniel. 25 November 2008. **Konsep XML Web Services**. <http://ilmukomputer.org/2008/11/25/konsep-xml-web-services/>.
- Ravi Trivedi. 23 April 2003. **Web Services Tutorial**. <http://www.developer.com/services/article.php/2195981/Web-Services-Tutorial-Understanding-XML-and-XML-SchemamdashPart-1.htm>.
- Memperkenalkan: XML-RPC**. <http://blog.galihatria.com/2008/08/21/memperkenalkan-xml-rpc/>. 6 Januari 2011.