

PHP Web Service

1. Pengenalan web service

Konsep teknologi Web Service muncul untuk mendukung sistem terdistribusi yang berjalan pada infrastruktur yang berbeda. Dengan adanya kombinasi dari XML (*eXtensible Markup Language*) dan HTTP (*HyperText Transport Protocol*), Web Service yang berbasis XML sangat mungkin untuk diimplementasikan. Bahkan teknologi Web Service ini mampu menggantikan fungsi dari beberapa teknologi yang serupa yang telah berkembang yaitu CORBA (*Common Object Request Broker Architecture*), DCOM (*Dynamic Component Object Model*), dan Java RMI (*Remote Method Invocation*).

SOAP (*Simple Object Appication Protocol*) dan beberapa teknologi yang didukung seperti WSDL (*Web Service Description Language*) dan UDDI (*Universal Description Discovery, and Integration*) merupakan kombinasi dari XML yang dikirimkan melalui HTTP. SOAP menyediakan transmisi data yang sederhana dan fleksibel, pemanggilan fungsi, pengembalian nilai, *multiplatform*, dan dapat dikembangkan dengan menggunakan berbagai bahasa pemrograman.

Berikut merupakan salah satu contoh dari pesan SOAP (SOAP Message):

```
<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body SOAPENV:
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<taxCalc xmlns="urn:soap-whyte wolf-ca:taxcalc" id="o0" SOAP-ENC:root="1">
<rate xmlns="" xsi:type="xsd:string">7</rate>
<sub xmlns="" xsi:type="xsd:string">856</sub>
</taxCalc>
</SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

2. Apa itu XML web service

Dalam pengertian yang sederhana , XML Web Services dapat di definisikan sebagai aplikasi yang diakses oleh aplikasi yang lain. Mungkin orang berpendapat itu semacam web site, tetapi itu bukan demikian. Ada perbedaan – perbedaan yang membedakan dengan web site.

Perbedaan tersebut dapat dilihat dibawah ini

WEB SITE

1. Memiliki web interface
2. Dibuat untuk ber interaksi langsung dengan user
3. Dibuat untuk bekerja pada web browser.

WEB SERVICES

1. Tidak memiliki interface yang bagus
2. Dibuat untuk ber interaksi langsung dengan aplikasi yang lain baik beda OS / Konsep sekalipun.

3. Dibuat untuk bekerja pada semua tipe client aplikasi / perangkat device Sesuai dengan namanya, XML Web Service menyimpan datanya dalam format XML dan karenanya menjadi multi platform dalam hal accessibilitynya.

Web Service dapat disamakan dengan library tersebut tetapi tidak memerlukan registrasi khusus ke dalam sistem operasi yang menyimpannya. Web Service tersimpan di Web Server sehingga dapat diakses oleh berbagai bahasa pemrograman dengan lebih mudah baik dalam lingkungan LAN maupun Internet.

Sistem Web Service ini diharapkan meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam Web Service dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya.

3. Kapan kita gunakan web service?

Web Services itu digunakan saat kita akan mentransformasi sebuah bisnis logik / sebuah class dan object yang terpisah dalam 1 ruang lingkup yang menjadi satu, sehingga tingkat keamanan dan security dapat di tangani dengan baik. Selain itu Web Service juga lebih mudah dalam process deploymentnya, karena tidak memerlukan registrasi khusus ke dalam sistem operasi. Web Service cukup diupload ke Web Server dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.

Web Service berjalan di port 80 yang merupakan protokol standar HTTP, dengan demikian mengurangi resiko terblokir oleh firewall. Kendala arsitektur COM/DCOM adalah memerlukan konfigurasi khusus di sisi firewall, dan ini tidak perlu dilakukan untuk

mengakses Web Service. Beberapa vendor luar negeri mulai berkolaborasi satu sama lain dengan konsep web services , diantaranya : IBM , Microsoft , SUN , ORACLE Diantaranya contoh web services yang sudah jadi dan dipakai adalah web services keluaran Microsoft (Microsoft Passport) – web services untuk user name dan password yang sudah dipasang di web site Microsoft dan HOTMAIL.

4. Penerapan web service untuk aplikasi zodiak

Web Service yang akan dikembangkan menggunakan NuSOAP API (*Application Programming Interface*). Anda dapat memperoleh API tersebut pada situs <http://deitrich.ganx4.com/nusoap/>. Penulis menggunakan database MySQL 5 (<http://www.mysql.com>) sebagai media penyimpanan data zodiak. Sedangkan Web Server yang digunakan adalah Apache 2 (<http://www.apache.org>).

Berikut adalah contoh enerapan web service :

1. Menyiapkan database yang akan digunakan

Sebelum menulis kode program untuk aplikasi server, terlebih dahulu Anda siapkan database yang akan digunakan, yaitu data mengenai zodiak.

```
mysql> CREATE DATABASE zodiak;  
mysql> CREATE TABLE ramalan (nama_zodiak VARCHAR(20) PRIMARY KEY,  
tanggal VARCHAR(30) NOT NULL, ramalan VARCHAR(255) NOT NULL, keuangan  
VARCHAR(255) NOT NULL, kesehatan VARCHAR(255) NOT NULL,  
angka_keberuntungan INT NOT NULL);
```

2. Menulis kode program untuk aplikasi server

Setelah Anda membuat database 'zodiak', langkah selanjutnya yaitu membuat kode program untuk aplikasi server dari Web Service.

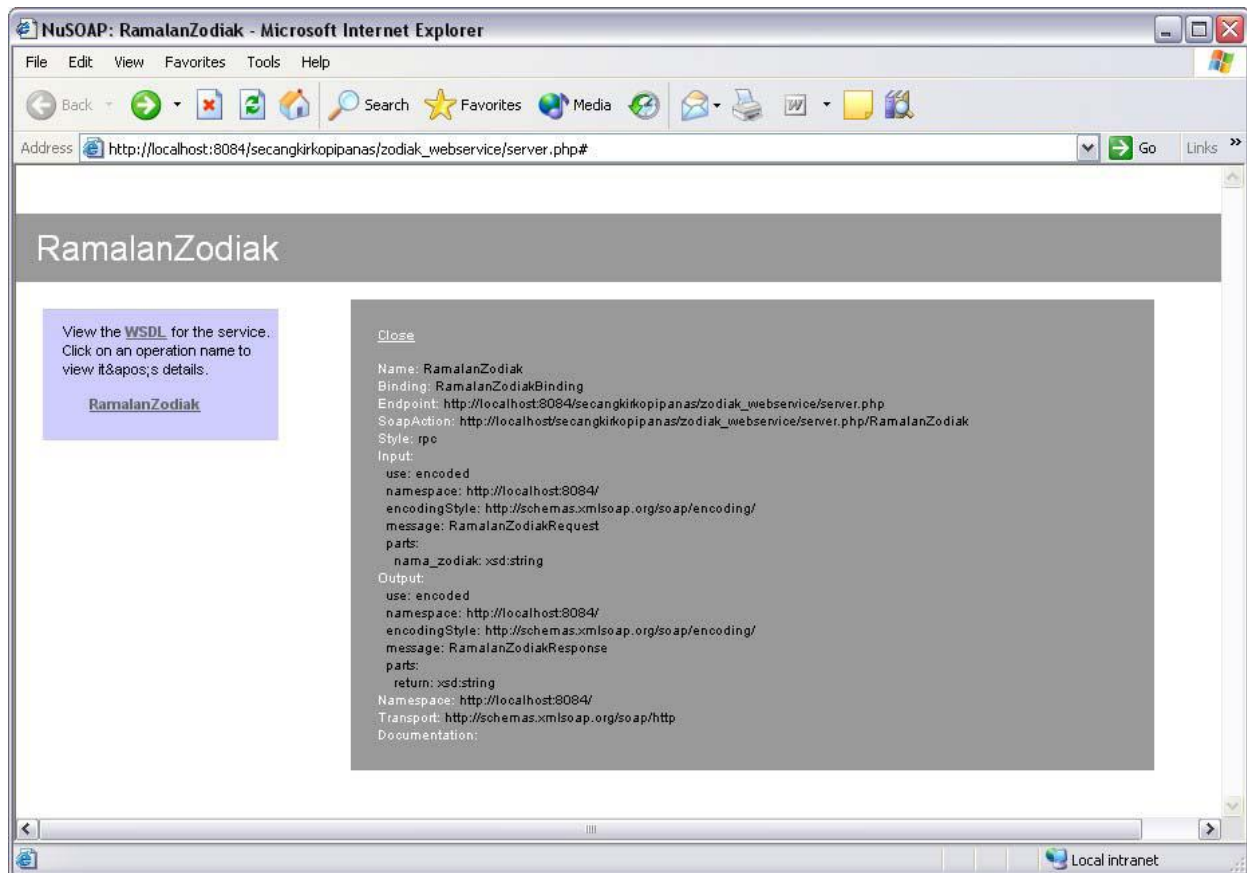
```

<?
require_once('nusoap.php');
$ns = "http://localhost:8084/";
$server = new soap_server;
$server->configureWSDL('RamalanZodiak', $ns);
$server->wsdl->schemaTargetNamespace = $ns;
$server->register('RamalanZodiak', array('nama_zodiak' => 'xsd:string'),
array('return'=>'xsd:string'), $ns);
function RamalanZodiak($nama_zodiak) {
if (!$nama_zodiak) {
return new soap_fault('Client', '', 'Harus ada nilainya!', '');
}
if ($conn = mysql_connect("host", "user", "password")) {
if ($db = mysql_select_db("zodiak")) {
$result = mysql_query("SELECT * FROM ramalan WHERE
nama_zodiak = '$nama_zodiak'");
while ($row = mysql_fetch_array($result)) {
$zodiak = $row["nama_zodiak"];
$tanggal = $row["tanggal"];
$ramalan = $row["ramalan"];
$keuangan = $row["keuangan"];
$kesehatan = $row["kesehatan"];
$angka_keberuntungan = $row["angka_keberuntungan"];
}
} else {
return new soap_fault('Database Server', '', 'Koneksi ke
database gagal!', '');
}
} else {
return new soap_fault('Database Server', '', 'Koneksi ke database
gagal!', '');
}
return "<b>Zodiak: </b>$zodiak<br>
<b>Tanggal: </b>$tanggal<br>
<b>Ramalan: </b>$ramalan<br>
<b>Keuangan: </b>$keuangan<br>
<b>Kesehatan: </b>$kesehatan<br>
<b>Angka Keberuntungan: </b>$angka_keberuntungan";
}

$server->service($HTTP_RAW_POST_DATA);
exit();
?>

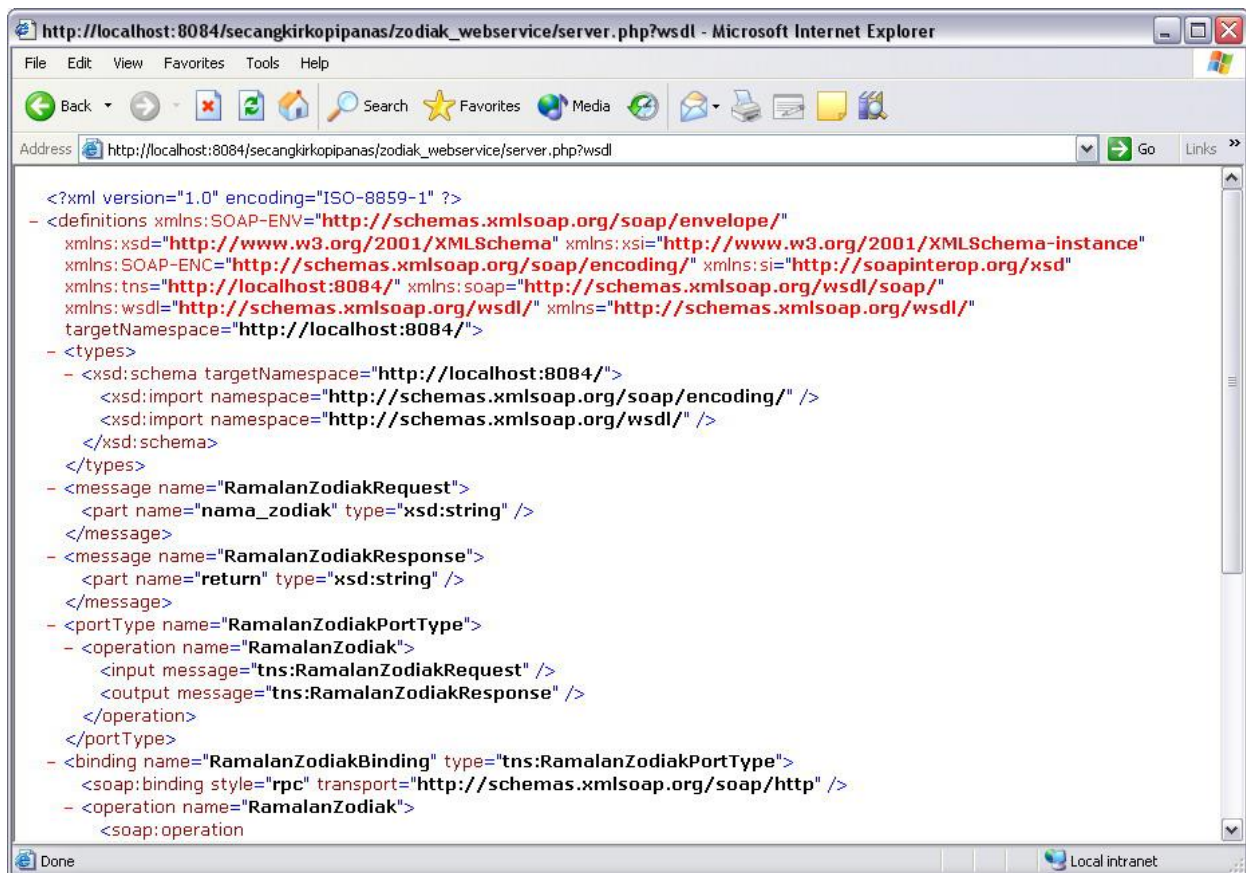
```

Untuk memastikan apakah aplikasi *server* yang telah dibangun dapat berjalan dengan baik atau tidak, ada baiknya kalau Anda melakukan pengujian terlebih dahulu sebelum Anda menulis kode program untuk aplikasi *client* dari Web Service.



Gambar 1. Aplikasi Server dari Web Service Zodiak

Jika pengujian yang Anda lakukan berhasil, maka pada *browser* Anda akan tampil seperti gambar di atas. Anda dapat melihat deskripsi dari Web Service yang Anda bangun dengan memilih menu WSDL pada bagian kiri atas. Tampilan dari deskripsi Web Service Zodiak adalah sebagai berikut:



Gambar 2. WSDL dari Web Service Zodiak

3. Menulis kode program untuk aplikasi *client*

Langkah berikutnya adalah menulis kode program untuk aplikasi *client*. Aplikasi *client* akan melakukan permintaan layanan pada *server* Web Service, dan akan menerima nilai yang dikembalikan oleh *server* Web Service.

```
<?
require_once('nusoap.php');
$params = array('nama_zodiak' => $nama_zodiak);
$client = new
soapclient('http://localhost:8084/secangkirkopipanas/zodiak_webservice/
server.php');
$response = $client->call('RamalanZodiak', $params);
```

```

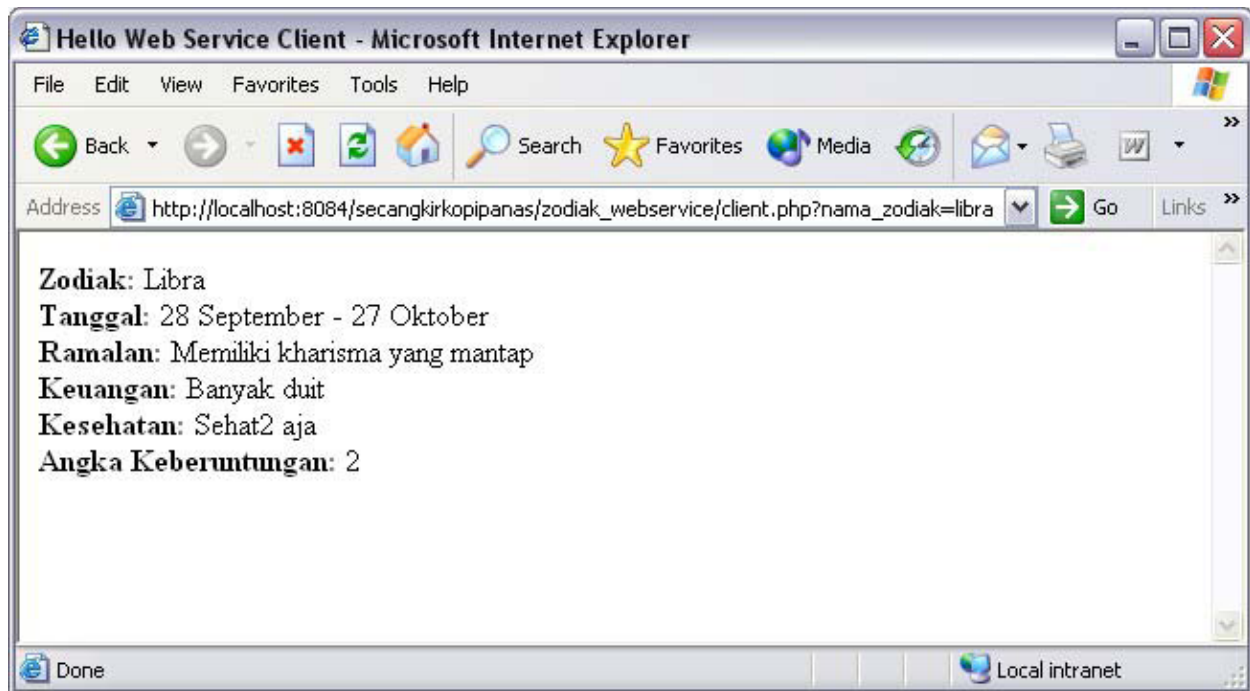
if ($client->fault) {
echo "FAULT:<br>";
echo "Code: { $client->faultcode }<br>";
echo "String: { $client->faultstring }";
} else {
echo $response;
}
$client=new soapclient($wsdl, 'wsdl');
$wsdl="http://localhost:8084/secangkirkopipanas/zodiak_webservice/
server.php?wsdl";
echo $response = $client->call('RamalanZodiak', $param);

?>

```

Untuk memastikan apakah aplikasi *client* yang telah dibangun dapat berjalan dengan baik atau tidak, lakukan pengetesan pada aplikasi *client*. URL dari aplikasi *client* ditambahkan parameter `'nama_zodiak'`.
http://localhost:8084/secangkirkopipanas/zodiak_webservice/client.php?nama_zodiak=libra

Jika berhasil, maka Anda akan mendapat tampilan *browser* seperti gambar berikut:



Gambar 3. Aplikasi Client dari Web Service Zodiak

Aplikasi zodiak menggunakan PHP Web Service yang baru saja Anda bangun, dapat dikembangkan sesuai dengan kebutuhan Anda. Selain itu, Anda juga dapat mengkolaborasikan Web Service zodiac ini dengan aplikasi pada *mobile phone* menggunakan J2ME atau dengan aplikasi pada PDA menggunakan Microsoft .NET.

5. Daftar Pustaka

- <http://dietrich.ganx4.com/nusoap>
- <http://www.google.com>