

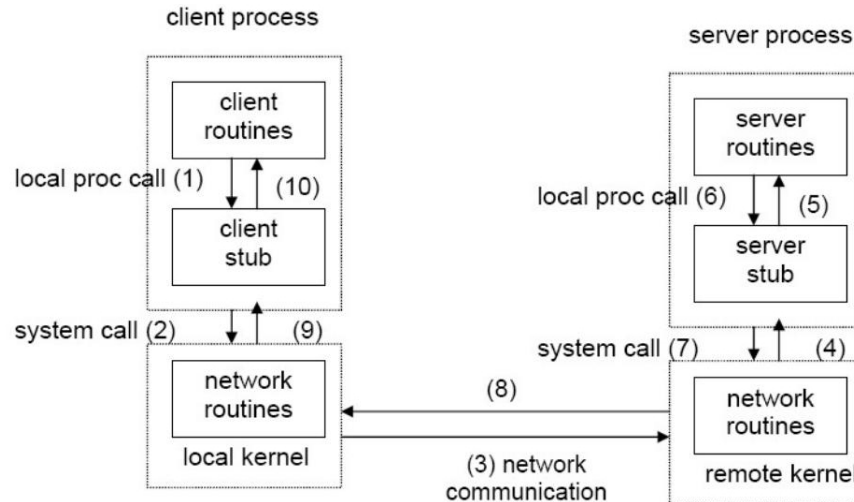
MODUL 2

REMOTE PROCEDURE CALL (RPC)

2.1 Pengertian

2.1.1 Remote Procedure Call (RPC)

Remote Procedure Call (RPC) adalah sebuah metoda yang memungkinkan kita untuk mengakses sebuah prosedur yang berada di komputer lain. Untuk dapat melakukan ini komputer lain. Untuk dapat melakukan ini sebuah komputer (server) harus menyediakan layanan remote prosedur. Pendekatan yang dilakukan adalah, sebuah server membuka socket, menunggu client yang meminta



Gambar 1. RPC Model

1. Client memanggil local procedure, yang disebut client stub. Bagi client, client stub seolah-olah server procedure yang dipanggil. Stub melakukan marshalling: argumen untuk remote procedure diatur dalam paket dengan mengikuti suatu format standard, menjadi sebuah network message.
2. Melalui pemanggilan system call di local kernel, client stub mengirim network message ke remote system.
3. Network message ditransfer ke remote system, menggunakan connection-oriented maupun connectionless protocol.
4. Server stub menunggu request dari client, melakukan unmarshalling argumen yang ada di network message, dan melakukan konversi jika diperlukan.
5. Server stub menjalankan local procedure call yang memanggil server procedure yang

sebenarnya.

6. Setelah server procedure selesai, return values dikembalikan ke server stub.
7. Server stub melakukan konversi (jika diperlukan), kemudian melakukan marshalling menjadi network message, dan mengirimkannya kembali ke client stub.
8. Network message ditransfer kembali ke client stub.
9. Client stub membaca network message dari local kernel.
10. Client stub mengkonversi return values (jika diperlukan), kemudian mengembalikannya ke client routine, seperti return values dari local procedure.

2.1.2 Extensible Markup Language (XML)

Extensible Markup Language (XML) adalah bahasa komputer yang dibuat oleh *World Wide Web Consortium (W3C)* untuk menyederhanakan proses pertukaran dan penyimpanan data. Hal ini disebabkan keunikan dan perbedaan sistem yang digunakan oleh masing-masing server yang terhubung ke internet. Maka dari itu, diperlukan adanya standardisasi proses transfer data antar server.

XML akan menyimpan data dalam format teks yang sederhana. Jadi, data tersebut akan dapat dimengerti oleh server yang menerima data tanpa perlu perubahan atau modifikasi apapun. Hal ini tentunya sangat memudahkan dalam penyimpanan dan pengiriman data antar server.

XML memiliki struktur yang cukup sederhana, yakni terdiri atas tiga segmen. Berikut penjelasannya:

Deklarasi: bagian penting dalam XML untuk menunjukkan versi XML yang digunakan.

Atribut: berisi keterangan objek.

Elemen: berisi terdiri dari tag pembuka, isi, dan tag penutup yang mendeskripsikan objek.

Berikut contoh struktur XML:

```
<?xml version="1.0"?>
<breakfast_menu title="Breakfast Menu Restaurant A">
  <food>
    <name>Belgian Waffles</name>
    <price>Rp35.000</price>
    <description>Two of our famous Belgian Waffles with plenty of real
    maple syrup</description>
    <calories>650</calories>
  </food>
</breakfast_menu>
```

Deklarasi (pointing to `<?xml version="1.0"?>`)
Atribut (pointing to `title="Breakfast Menu Restaurant A"`)
Elemen (pointing to `<name>Belgian Waffles</name>`)

Gambar 2. Contoh Struktur XML

Pada contoh struktur XML di atas, deklarasi menunjukkan versi XML yang digunakan, yaitu 1.0. Sedangkan, atribut yang digunakan pada data breakfast menu adalah “*Breakfast Menu Restaurant A*”. Data terdiri atas empat bagian yakni *name*, *price*, *description* dan *calories*. Setiap bagian tersebut terdiri atas tiga elemen, misalnya `<name>` sebagai tag pembuka, *Belgian Waffles* sebagai isi, dan `</name>` sebagai tag penutup.

2.2 Aplikasi XML RPC Sederhana

Buatlah folder `/opt/lampp/htdocs/rpc-xml-simple` pada Debian Server dan buat file `server.php`. Buat juga folder `xampp\htdocs\rpc-xml-simple` pada Windows dan buat file `client1.php`, `client2.php` serta `client3.php`. Akses `phpinfo` pada localhost dan Debian Server serta pastikan PHP extension `xml` dan `xmlrpc` sudah aktif.

https://192.168.56.30/dashboard/phpinfo.php 120% Search	
xml	
XML Support	active
XML Namespace Support	active
libxml2 Version	2.9.4
xmlreader	
XMLReader	enabled
xmlrpc	
core library version	xmlrpc-epi v. 0.51
php extension version	0.51
author	Dan Libby
homepage	http://xmlrpc-epi.sourceforge.net
open sourced by	Epinions.com

Gambar 3. PHP *xml* dan *xmlrpc* Extension

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

Jika belum aktif, edit file php.ini di D:\>xampp\etc\php.ini. Hilangkan semicolon (;) pada ekstensi *php_xmlrpc*

extension=php_xmlrpc.dll

lalu *reboot* Apache

2.2.1 Source code *server.php* di *Debian Server*

```
1 <?php
2 header("Content-Type:text/xml;charset=UTF-8");
3
4 // request dari Client ke Server
5 if (($_SERVER['REQUEST_METHOD']=='POST') and ($_GET['user']=='pengguna') and ($_GET['password']=='pin'))
6 {
7     $input = file_get_contents("php://input");
8     $data = xmlrpc_decode($input);
9 } else
10 {
11     $data = array("nim"=>"19650000", "nama"=>"Anton", "kota"=>"Malang");
12 }
13 // response dari Server ke Client
14 $response = xmlrpc_encode($data);
15 echo ($response);
16 ?>
```

Gambar 4. File Source code *server.php*

Akses melalui web browser <http://192.168.56.xx/rpc-xml-simple/server.php>

2.2.2 Source code *client1.php* di *Windows*

```
1 <?php
2 // ambil data dari Server
3 $file = file_get_contents("http://192.168.56.30/rpc-xml-simple/server.php?user=pengguna&password=pin", false, null);
4
5 // response dari Server ke Client
6 $response = xmlrpc_decode($file);
7 if ($response && xmlrpc_is_fault($response)) {
8     trigger_error("xmlrpc: $response[faultString] ($response[faultCode])");
9 } else {
10     echo "<pre>";
11     print_r($response);
12     echo "</pre>";
13     echo "-----";
14     echo "<br>nim : ".$response['nim'];
15     echo "<br>nama : ".$response['nama'];
16     echo "<br>kota : ".$response['kota'];
17 }
18 ?>
```

Gambar 5. File Source code *client1.php*

Akses melalui web browser <http://localhost/rpc-xml-simple/client1.php>

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

2.2.3 Source code client2.php di Windows

```
1 <?php
2 // request dari Client ke Server
3 $request = xmlrpc_encode_request("method",array("nim=>"18650200","nama=>"Ahmad Hasan Alfarisi"));
4 $context = stream_context_create(array('http' => array(
5     'method' => "POST",
6     'header' => "Content-Type:text/xml;charset=UTF-8",
7     'content' => $request
8 )));
9
10 // ambil data dari Server
11 $file = file_get_contents("http://192.168.56.30/rpc-xml-simple/server.php?user=pengguna&password=pin",false,$context);
12
13 // response dari Server ke Client
14 $response = xmlrpc_decode($file);
15 if ($response && xmlrpc_is_fault($response)) {
16     trigger_error("xmlrpc: $response[faultString] ($response[faultCode])");
17 } else {
18     echo "<pre>";
19     print_r($response);
20     echo "</pre>";
21     echo "-----";
22     echo "<br/>nim : ".$response[0]['nim'];
23     echo "<br/>nama : ".$response[0]['nama'];
24 }
25 ?>
```

Gambar 6. File Source code client2.php

Akses melalui web browser <http://localhost/rpc-xml-simple/client2.php>

2.2.4 Source code client3.php di Windows

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Belajar XML RPC</title>
5 </head>
6 <body>
7     <form action="" method="POST">
8         <label for="fname">NIM</label>
9         <input type="text" name="nim"><br/><br/>
10
11         <label for="lname">Nama</label>
12         <input type="text" name="nama"><br/><br/>
13         <input type="submit" value="Submit">
14     </form>
15 </body>
16 </html>
17
18 <?php
19 if (isset($_POST['nim'])) {
20     // request dari Client ke Server
21     $request = xmlrpc_encode_request("method",array("nim=>".$_POST['nim'], "nama=>".$_POST['nama']));
22     $context = stream_context_create(array('http' => array(
23         'method' => "POST",
24         'header' => "Content-Type:text/xml;charset=UTF-8",
25         'content' => $request
26     )));
27
28     // ambil data dari Server
29     $file = file_get_contents("http://192.168.56.30/rpc-xml-simple/server.php?user=pengguna&password=pin",false,$context);
30
31     // response dari Server ke Client
32     $response = xmlrpc_decode($file);
33     if ($response && xmlrpc_is_fault($response)) {
34         trigger_error("xmlrpc: $response[faultString] ($response[faultCode])");
35     } else {
36         echo "<pre>";
37         print_r($response);
38         echo "</pre>";
39         echo "-----";
40         echo "<br/>nim : ".$response[0]['nim'];
41         echo "<br/>nama : ".$response[0]['nama'];
42     }
43 }
44 ?>
```

Gambar 7. File Source code client3.php

Akses melalui web browser <http://localhost/rpc-xml-simple/client3.php>