



MODUL PRAKTIKUM

SISTEM TERDISTRIBUSI DAN KEAMANAN

TEKNIK INFORMATIKA

UIN MAULANA MALIK IBRAHIM MALANG

TAHUN AJARAN 2022-2023

**MODUL 1
PEMROGRAMAN SOCKET**

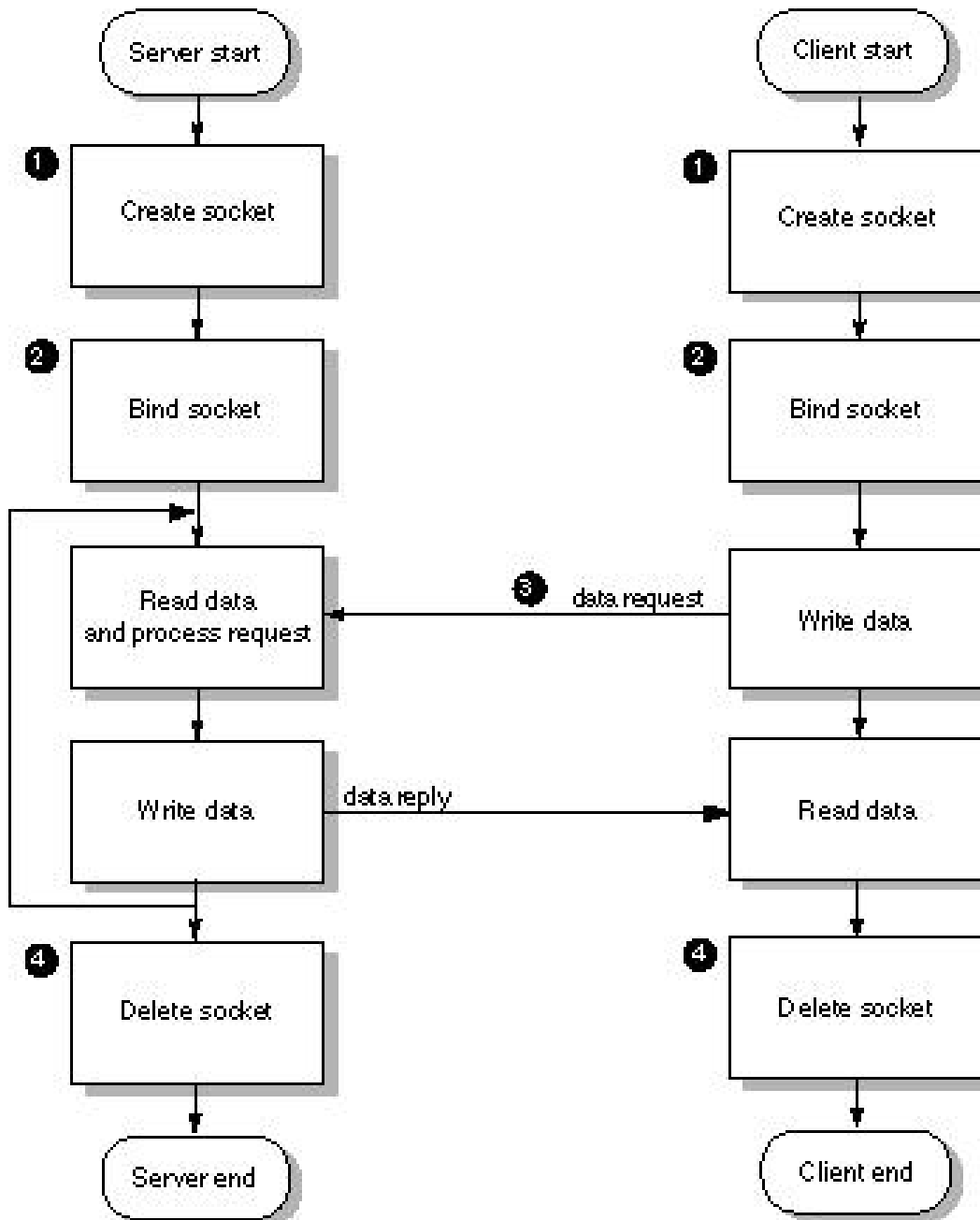
1.1 Pengertian

Socket adalah *interface* pada jaringan yang menjadi titik komunikasi antar mesin pada Internet Protocol. Socket adalah titik komunikasi dari lalu lintas komunikasi dari lalu lintas komunikasi antar proses di dalam sebuah jaringan komputer.

- Contoh :
 1. Chat
 2. Transfer File (seperti FileZilla, WinSCP)
 3. Transaksi-transaksi penting seperti Mesin ATM di Bank
 4. Game online
 5. Monitoring jaringan
 6. Aplikasi-aplikasi broadcast seperti Youtube

- Elemen Utama
 1. Protokol (TCP/UDP)
 2. Local IP Address
 3. Local Port
 4. Remote IP Address
 5. Remote Port

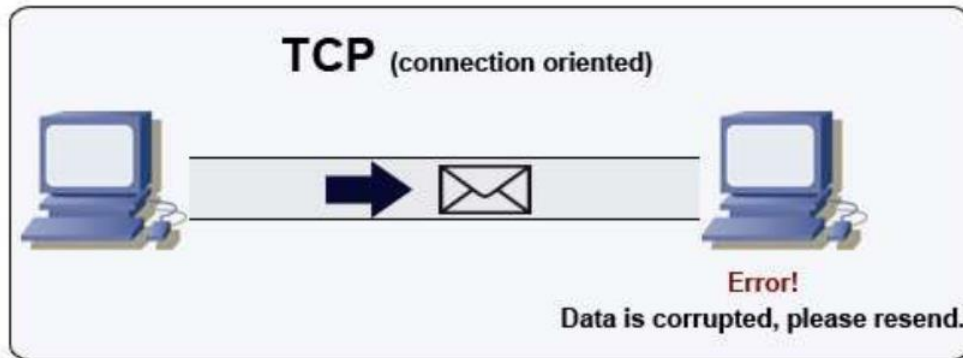
PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG



Gambar 1. Bagan Socket Client Server

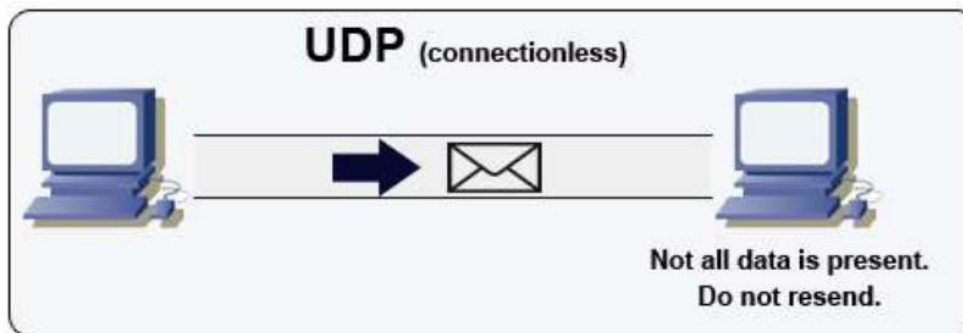
PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

- Pengiriman data pada aplikasi socket
 1. TCP (*Transmission Control Protocol*) Aplikasi FTP, SMTP, SSH, TELNET, aplikasi chat, transfer file, ataupun transaksi-transaksi penting seperti Mesin ATM di Bank.



Gambar 2. TCP

2. UDP (*User Data Protocol*) Aplikasi SunRPC, SNMP, DNS, TFTP, aplikasi monitoring jaringan, game online dan aplikasi-aplikasi broadcast seperti Youtube.



Gambar 3. UDP

- Port
Port merupakan sebuah koneksi data virtual yang digunakan aplikasi untuk bertukar data secara langsung. Contoh :
 - HTTP 80
 - HTTPS 443
 - FTP 21
 - SSH 22
 - TELNET 23
 - DNS 53

- Jenis Port

1. Well-known ports

Merupakan port yang telah digunakan secara internal oleh Sistem, misalnya port untuk service HTTP, HTTPS, FTP, SSH, DNS dan seterusnya. Port yang telah digunakan ini adalah port 0 sampai dengan port 1023.

2. Registered ports

Port ini dapat digunakan dalam aplikasi Anda, range-nya adalah port 1024 hingga port 49151 dan dapat dipilih secara bebas.

3. Dynamic/Private ports

Dari port 49152 sampai dengan port 65535.

1.2 Pemrograman Socket

Pada praktikum kali ini, pastikan Anda memiliki **VSCode** / **Sublime Text** untuk mengetikkan *source code*. Buatlah folder *socket* yang berisikan file *socket.php* dan *server.php* pada folder **xampp/htdocs/**. *Source code socket.php* dan *server.php* ini hanya dapat dijalankan pada **PHP versi 5**. Akses phpinfo pada localhost menggunakan web browser <http://localhost/dashboard/phpinfo.php> dan pastikan PHP extension **sockets** sudah aktif.



Gambar 4. *PHP Sockets Extension*

Jika belum aktif, edit file `php.ini` di `D:\>xampp\etc\php.ini`. Hilangkan semicolon (;) pada ekstensi `php_sockets`
`extension=php_sockets.dll`
lalu *reboot* Apache

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

1.2.1 Aplikasi sederhana untuk mengambil Laman *Web Google* dengan Pemrograman Socket

Berikut ini merupakan contoh mengambil laman web di google menggunakan pemrograman socket.

1. Pada file *socket.php*, ketikkan source code berikut dan simpan file tersebut.

```
<?php
// Creating a socket
if(!($sock = socket_create(AF_INET,SOCK_STREAM,0)))
{
    // Error handling
    $errorcode = socket_last_error();
    $errmsg = socket_strerror($errorcode);
    die("Couldn't create socket: [$errorcode] $errmsg \n");
}
echo "Socket created \n-----\n";

// Get the IP address for the target host from internet
$address = gethostbyname('www.google.com');
//$address = '127.0.0.1';

// Connect to a Server
if(!socket_connect($sock,$address,80))
{
    // Error handling
    $errorcode = socket_last_error();
    $errmsg = socket_strerror($errorcode);
    die("Could not connect: [$errorcode] $errmsg \n");
}
echo "Connection established \n-----\n";

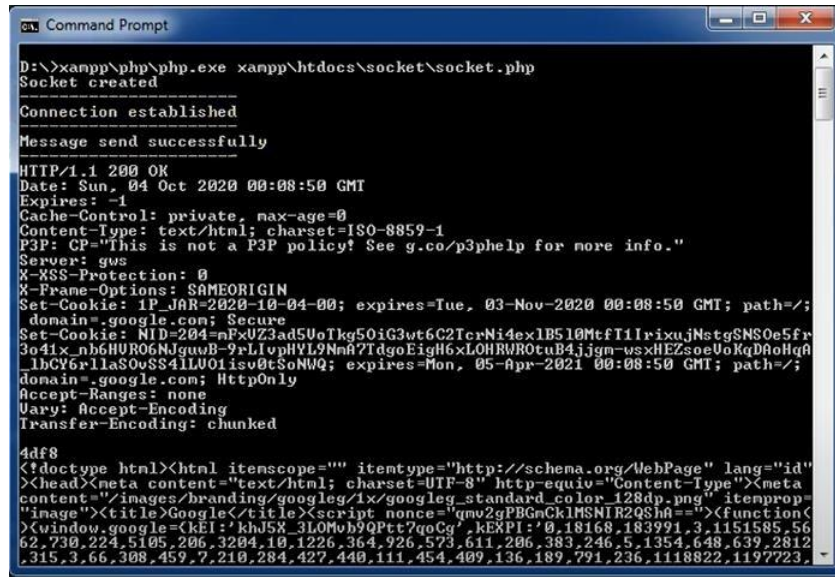
//Send the message to the server
//The message is actually a http command to fetch the mainpage of a website
$message = "GET / HTTP/1.1\n\n\n";
if(!socket_send($sock,$message,strlen($message),0))
{
    // Error handling
    $errorcode = socket_last_error();
    $errmsg = socket_strerror($errorcode);
    die("Could not send data: [$errorcode] $errmsg \n");
}
echo "Message send successfully \n-----\n";

//Now receive reply from server
if(socket_recv($sock,$buf,2045,MSG_WAITALL) === FALSE)
{
    // Error handling
    $errorcode = socket_last_error();
    $errmsg = socket_strerror($errorcode);
    die("Could not receive data: [$errorcode] $errmsg \n");
}
echo $buf." \n-----\n";
// Close socket
socket_close($sock);
?>
```

Gambar 5. File *socket.php*

2. Pastikan perangkat terkoneksi dengan internet.
3. Jalankan file *socket.php* melalui Command Prompt di Windows.

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG



```
cmd Command Prompt
D:\>xampp\php\php.exe xampp\htdocs\socket\socket.php
Socket created
-----
Connection established
-----
Message send successfully
-----
HTTP/1.1 200 OK
Date: Sun, 04 Oct 2020 00:08:50 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1
P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."
Server: gus
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
Set-Cookie: IP_JAR=2020-10-04-00; expires=Tue, 03-Nov-2020 00:08:50 GMT; path=/; domain=.google.com; Secure
Set-Cookie: NID=204=nFxUZ3ad5UoTkg50iG3wt6C2TcrNi4exlB5l0MtfT1IrixujNstgSMS0e5fr3o41x_nb6HURO6NjquvB-9rLIuphYL9NnaR7Tdg0EigH6xLOHRWR0tuB4jgm-vsxEZsoeUoKqDAoHqA_lbcY6r1laS0vSS4lLU01isu0tSoNWQ; expires=Mon, 05-Apr-2021 00:08:50 GMT; path=/; domain=.google.com; HttpOnly
Accept-Ranges: none
Vary: Accept-Encoding
Transfer-Encoding: chunked

4df8
<doctype html><html itenscope="" itemtype="http://schema.org/WebPage" lang="id"
<head><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta
content="images/branding/google/4x/googleg_standard_color_128dp.png" itemprop=
"image"><title>Google</title><script nonce="qmw2gPBGmCk1MSNTR2QSjha="><function<
><window.google=<klEl:'khJ5X_3LOMub9QPt7qoCg',kEXPI:'0,18168,183991,3,1151585,56
62,730,224,5105,206,3204,10,1226,364,926,573,611,206,383,246,5,1354,648,639,2812
,315,3,66,308,459,7,210,284,427,440,111,454,409,136,189,791,236,1118822,1197723,
```

Gambar 6. Menjalankan file *socket.php* melalui Command Prompt di Windows

1.2.2 Aplikasi *Chatting* dengan Pemrograman Socket

Berikut ini merupakan contoh membangun aplikasi chatting menggunakan pemrograman socket.

1. Buat file *server.php* dan ketikkan source code berikut dan simpan file tersebut.

```
<?php
error_reporting(1);
set_time_limit (0);

$address = "0.0.0.0";
$port = 5000;
$max_clients = 5;

// Creating a socket
if(!($sock = socket_create(AF_INET, SOCK_STREAM, 0)))
{
    // Error handling
    $errorcode = socket_last_error();
    $errmsg = socket_strerror($errorcode);
    die("Couldn't create socket: [$errorcode] $errmsg \n");
}
echo "Socket created \n";

// Bind the source address
if(!socket_bind($sock, $address, $port))
{
    // Error handling
    $errorcode = socket_last_error();
    $errmsg = socket_strerror($errorcode);
    die("Could not bind socket : [$errorcode] $errmsg \n");
}
```

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

```
}
echo "Socket bind OK \n";

// Listens for a connection on a socket
if(!socket_listen ($sock, $max_clients))
{ // Error handling
  $errorcode = socket_last_error();
  $errmsg = socket_strerror($errorcode);
  die("Could not listen on socket : [$errorcode] $errmsg
\n");
}
echo "Socket listen OK \n";

echo "Waiting for incoming connections... \n";

//array of client sockets
$client_socks = array();

//array of sockets to read
$read = array();

//start loop to listen for incoming connections and process
existing connections
while (true)
{ //prepare array of readable client sockets
  $read = array();

  //first socket is the master socket
  $read[0] = $sock;

  //now add the existing client sockets
  for ($i = 0; $i < $max_clients; $i++)
  { if($client_socks[$i] != null)
    { $read[$i+1] = $client_socks[$i];
    }
  }

  //Runs the select() system call on the given arrays of
sockets
  if(socket_select($read, $write, $except, null) === false)
  { // Error handling
    $errorcode = socket_last_error();
    $errmsg = socket_strerror($errorcode);
    die("Could not listen on socket : [$errorcode]
$errmsg \n");
  }
}
```


PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

```
//if ready contains the master socket, then a new
connection has come in
if (in_array($sock, $read))
{   for ($i = 0; $i < $max_clients; $i++)
    {   if ($client_socks[$i] == null)
        {   //Accepts a connection on a socket
            $client_socks[$i] = socket_accept($sock);

            //display information about the client who is
connected
            if(socket_getpeername($client_socks[$i],
$address, $port))
            {   echo "Client $address : $port is now
connected to Us. \n";
            }

            //Send Welcome message to client
            $message = "Welcome to php socket server
version 1.0 \n";
            $message .= "Enter a message and press enter.
I shall reply back \n";
            socket_write($client_socks[$i], $message);
            break;
        }
    }
}

//check each client if they send any data
for ($i = 0; $i < $max_clients; $i++)
{   if (in_array($client_socks[$i], $read))
    { //Reads a maximum of length bytes from a socket
      $input = socket_read($client_socks[$i], 1024);

      if ($input == null)
      { //zero length string meaning disconnected,
remove and close the socket
        // remove the socket
        unset($client_socks[$i]);
        // close the socket
        socket_close($client_socks[$i]);
      }

      $n = trim($input);
      $output = $client_socks[$i]. " Said: ... $input";
      echo "Sending output to client \n";

      //send response to client
    }
  }
}
```

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

```
//socket_write($client_socks[$i], $output);

// send response to other client
foreach (array_diff_key($client_socks, array($i
=> 0)) as $client_sock) {
    socket_write($client_sock, $output);
}
}
}
```

2. Jalankan file *server.php* tersebut melalui *Command Prompt* di Windows (*IP Address* 192.168.56.1).
D:\>xampp\php\php.exe xampp\htdocs\socket\server.php
3. Buka *VirtualBox* dan jalankan *Debian Server*.
4. Pastikan perangkat terkoneksi dengan internet. Tambahkan *Repository* pada *Debian Server* lalu install *Telnet*.

```
$ su
```

```
# nano /etc/apt/sources.list
```

```
deb http://deb.debian.org/debian/ bullseye main
```

```
deb-src http://deb.debian.org/debian/ bullseye main
```

```
deb http://security.debian.org/debian-security bullseye-security main
```

```
deb-src http://security.debian.org/debian-security bullseye-security main
```

```
deb http://deb.debian.org/debian/ bullseye-updates main
```

```
deb-src http://deb.debian.org/debian/ bullseye-updates main
```

```
deb http://deb.debian.org/debian bullseye main contrib non-free
```

```
deb-src http://deb.debian.org/debian bullseye main contrib non-free
```

```
# apt-get update
```

```
# apt-get install telnet
```

4. Shutdown *Debian Server* dengan perintah

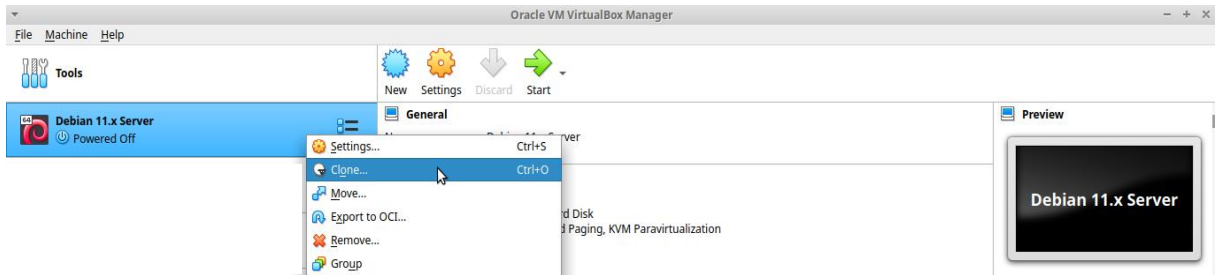
```
# /sbin/init 0
```

```
atau
```

```
# /sbin/shutdown -P now
```

5. Lakukan *Cloning* dengan klik kanan pada *Debian Server*, pilih “*Clone*” lalu pilih “*Full clone*” pada langkah selanjutnya.

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG



Gambar 7. *Debian Server Cloning*

6. Jalankan Debian Server *Clone* dan ganti IP Address pada Debian Server *Clone* agar tidak bentrok dengan IP Address Debian Server. *Reboot* Debian Server *Cloning*.

```
$ su
```

```
# nano /etc/network/interfaces
```

```
# /sbin/reboot
```

7. Jalankan kedua Debian Server tersebut dan masing-masing server menjalankan Telnet ke IP Address 192.168.56.1 (Windows) *port* 5000 untuk mencoba aplikasi *chatting*.

```
$ telnet 192.168.56.1 5000
```

8. Setelah selesai mencoba aplikasi *chatting*, hapus Debian Server *Clone* dan **jangan menghapus Debian Server**. Klik kanan pada Debian Server *Clone* lalu pilih *Remove* untuk menghapus Debian Server *Clone*.

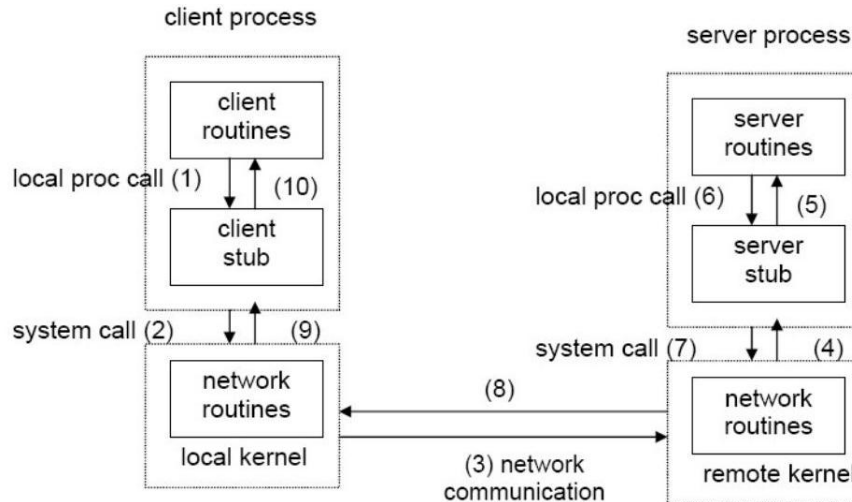
MODUL 2

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 standarisasi 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:

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

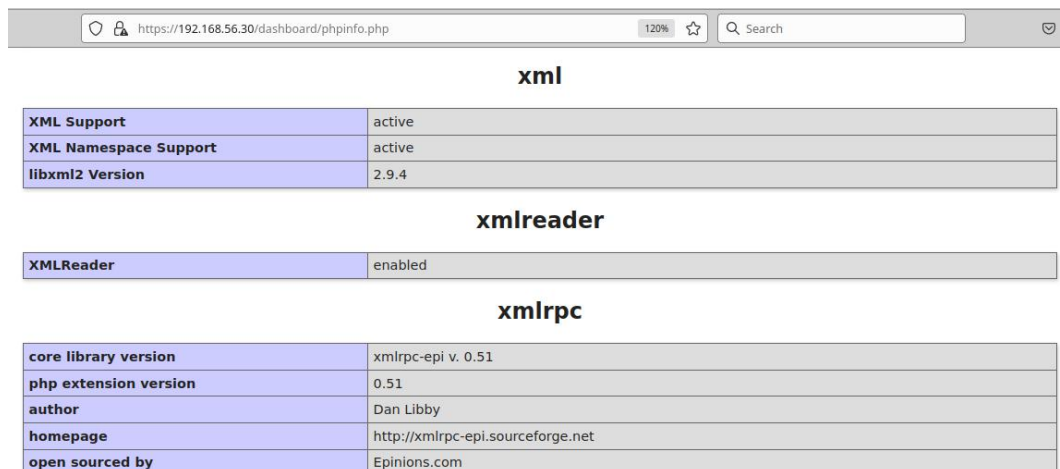
```
<?xml version="1.0"?>
<breakfast_menu title:"Breakfast Menu Restaurant A">
  <food>
    <name>Belgian Waffles</name>
    <price>Rp35.000</price>
    <description>Two of our famous Belgian Waffles with plenty of real
    maple syrup</description>
    <calories>650</calories>
  </food>
</breakfast_menu>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

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



The screenshot shows a web browser window with the address bar displaying `https://192.168.56.30/dashboard/phpinfo.php`. The page content is organized into three sections:

- 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 Extension `xml` dan `xmlrpc`

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. Source code server.php

Akses melalui web browser <http://192.168.56.xx/rpc-xml-simple/server.php> sehingga tampil laman Server yang menampilkan data format *XML*.



Gambar 5. Laman Server menampilkan data format *XML*

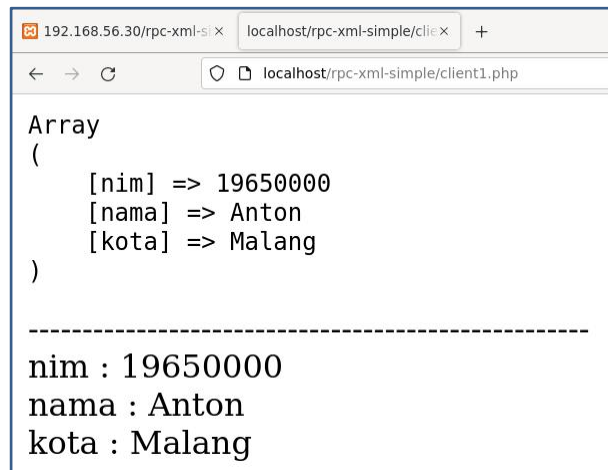
PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

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 6. Source code client1.php

Akses melalui web browser <http://localhost/rpc-xml-simple/client1.php> sehingga tampil laman Client 1. Pada laman Client 1 ini mengambil data dari Server lalu data ditampilkan.



Gambar 7. Laman Client 1

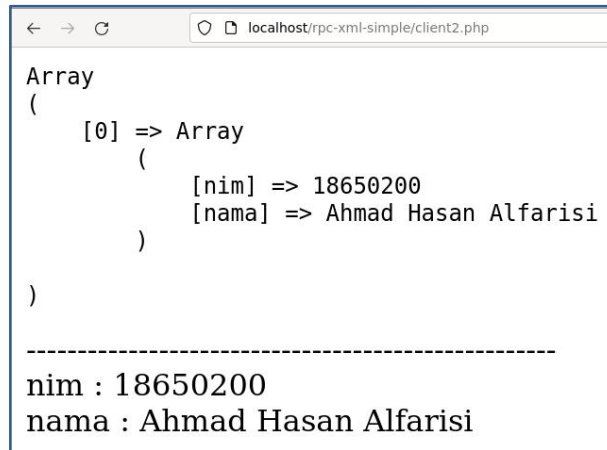
2.2.3 Source code client2.php di Windows

```
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 8. Source code client2.php

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

Akses melalui web browser <http://localhost/rpc-xml-simple/client2.php> sehingga tampil laman Client 2. Laman Client 2 ini mengirim data dengan format XML ke Server dengan method “POST”. Server kemudian merespon dengan mengirim ulang data dengan format XML ke Client 2. Client 2 mengambil data dari Server lalu data ditampilkan.



Gambar 9. Laman Client 2

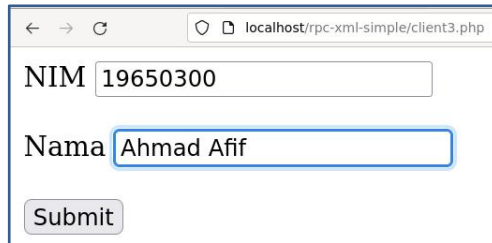
2.2.4 Source code client3.php di Windows

```
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 10. Source code client3.php

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

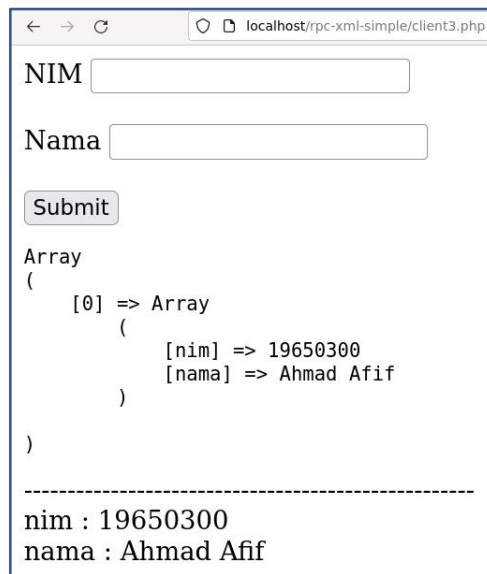
Akses melalui web browser <http://localhost/rpc-xml-simple/client3.php> lalu isi form dan klik Submit.



The screenshot shows a web browser window with the address bar containing 'localhost/rpc-xml-simple/client3.php'. Below the address bar, there is a form with two input fields: 'NIM' containing the value '19650300' and 'Nama' containing the value 'Ahmad Afif'. Below the input fields is a 'Submit' button.

Gambar 11. Laman Client 3

Laman Client 3 terdapat form yang mengirim data dengan format XML ke Server dengan method “POST”. Server kemudian merespon dengan mengirim ulang data dengan format XML ke Client 3. Client 3 mengambil data dari Server lalu data ditampilkan.



The screenshot shows the same web browser window after the form has been submitted. The 'NIM' and 'Nama' input fields are now empty. Below the 'Submit' button, the following data is displayed:

```
Array
(
    [0] => Array
        (
            [nim] => 19650300
            [nama] => Ahmad Afif
        )
)
```

Below a dashed line, the data is displayed in a human-readable format:

```
nim : 19650300
nama : Ahmad Afif
```

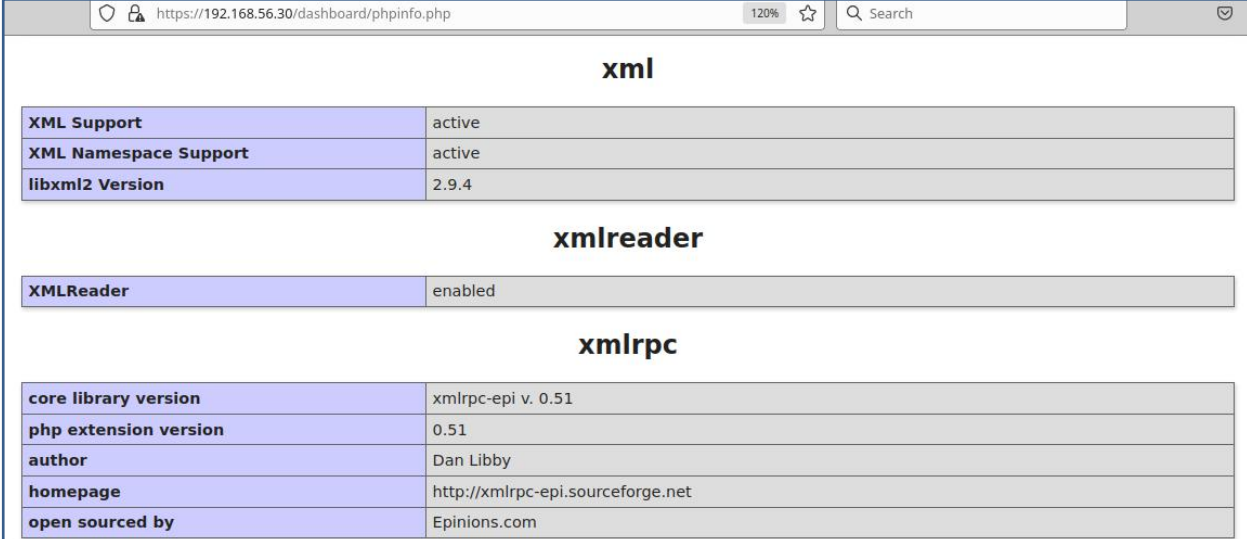
Gambar 12. Tampil data pada laman Client 3 setelah data di-submit ke Server

MODUL 3

Remote Procedure Call (RPC) - Create Read Update Delete (CRUD)

3.1 Aplikasi XML RPC - CRUD

Aplikasi Toko menggunakan XML RPC - CRUD sederhana, dimana **data akan dikirim dari Client (Windows dengan IP Address 192.168.56.1) ke Server (Debian Server dengan IP Address 192.168.56.xx) dan data hanya disimpan pada database Debian Server.** Buatlah database "toko" dan tabel "barang" pada Debian Server. Buat folder /opt/lampp/htdocs/rpc-xml-toko/server pada Debian Server lalu buat file Database.php dan server.php. Buat juga folder xampp\htdocs\rpc-xml-toko\client pada Windows lalu buat file RPCClient.php, proses.php serta index.php. Akses phpinfo pada Client (Windows) dan Server (Debian Server) serta pastikan PHP extension **xml** dan **xmlrpc** sudah aktif seperti pada gambar 1 di bawah ini.



The screenshot shows the PHP info page for the 'xml' and 'xmlrpc' extensions. The 'xml' section shows 'XML Support' as active, 'XML Namespace Support' as active, and 'libxml2 Version' as 2.9.4. The 'xmlreader' section shows 'XMLReader' as enabled. The 'xmlrpc' section shows 'core library version' as xmlrpc-epi v. 0.51, 'php extension version' as 0.51, 'author' as Dan Libby, 'homepage' as http://xmlrpc-epi.sourceforge.net, and 'open sourced by' as Epinions.com.

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

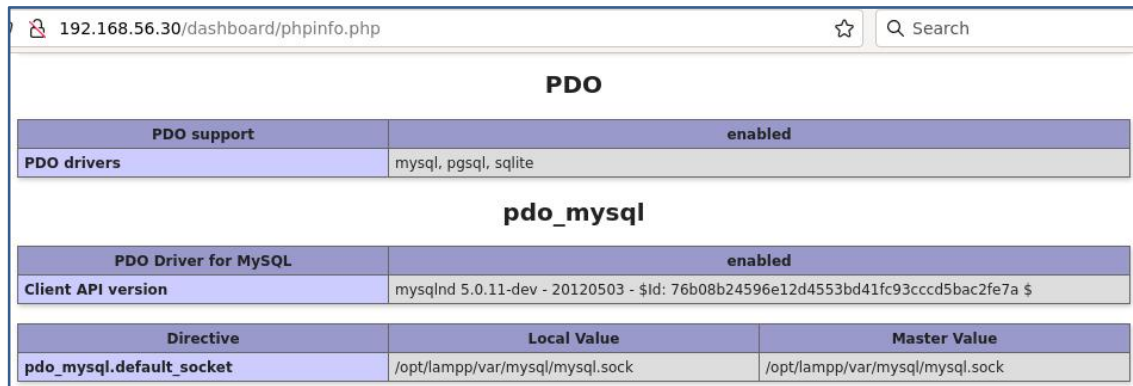
Gambar 1. PHP Extension xml dan xmlrpc

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**

Untuk Server (Debian Server), pastikan juga PHP extension **PDO** dan **pdo_mysql** sudah aktif seperti pada gambar 2 di bawah ini. PDO singkatan dari **PHP Data Objects** yang menawarkan sebuah paradigma Pemrograman Berorientasi Objek (**Object Oriented Programming**) dan didukung oleh banyak database seperti **MySQL/MariaDB, PostgreSQL, MS. SQL Server,**

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

Oracle, SQLite, dll.



The screenshot shows the PHPinfo.php page for PDO and pdo_mysql. The PDO section shows that PDO support is enabled and the supported drivers are mysql, pgsql, and sqlite. The pdo_mysql section shows that the PDO Driver for MySQL is enabled, the Client API version is mysqlnd 5.0.11-dev - 20120503 - \$Id: 76b08b24596e12d4553bd41fc93cccd5bac2fe7a \$, and the default socket is /opt/lampp/var/mysql/mysql.sock.

PDO		
PDO support	enabled	
PDO drivers	mysql, pgsql, sqlite	
pdo_mysql		
PDO Driver for MySQL	enabled	
Client API version	mysqlnd 5.0.11-dev - 20120503 - \$Id: 76b08b24596e12d4553bd41fc93cccd5bac2fe7a \$	
Directive	Local Value	Master Value
pdo_mysql.default_socket	/opt/lampp/var/mysql/mysql.sock	/opt/lampp/var/mysql/mysql.sock

Gambar 2. PHP Extension PDO dan pdo_mysql

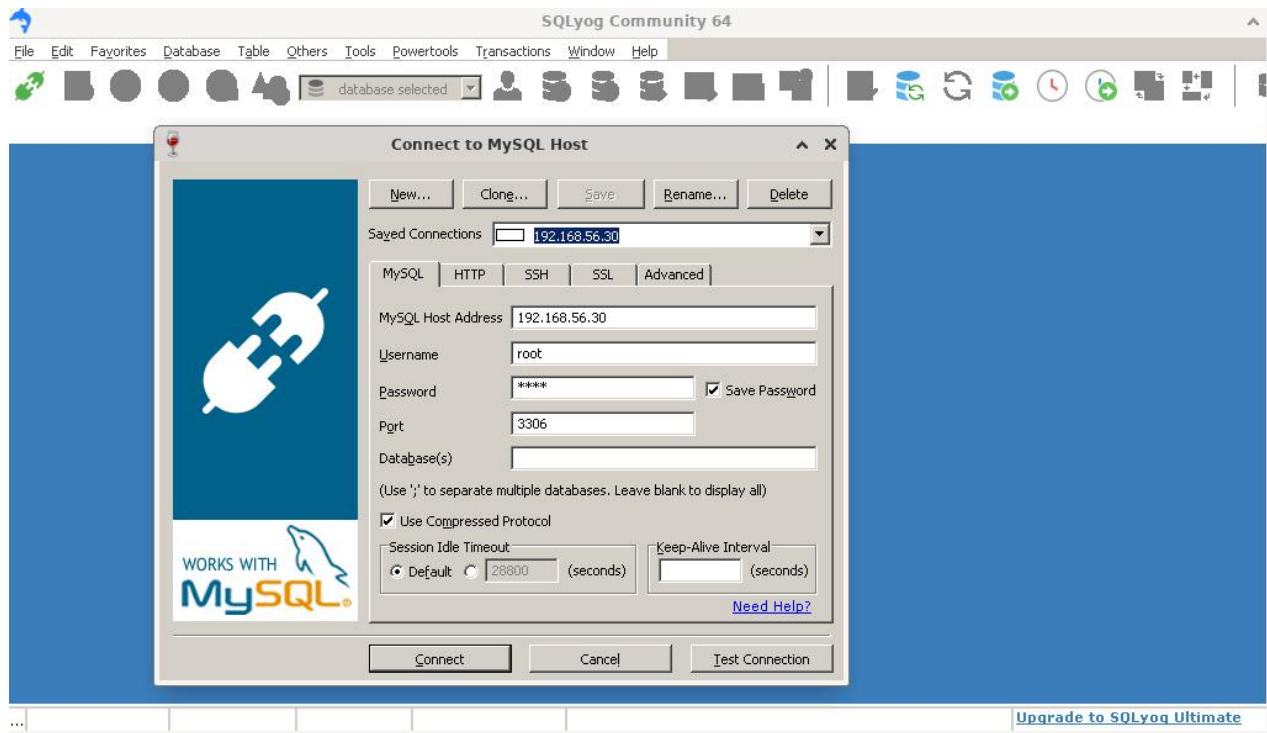
3.1.1 Database di Debian Server

Pada *Debian Server*, buatlah *database* dengan nama “*toko*” dan nama tabel “*barang*” di *phpmyadmin*, akses <http://192.168.56.xx/phpmyadmin>. *Database Debian Server* juga dapat di-*remote* dari *Client Windows*. *Login SSH* ke *Debian Server* menggunakan *PuTTY* dan buat *user* dengan *host %* di *database MySQL/MariaDB* agar dapat diakses dari *IP Address* manapun.

```
$ mysql -u root
atau jika user root sudah di-password
$ mysql -u root -p
MariaDB [(none)]> GRANT ALL PRIVILEGES ON *.* TO 'root'@%' IDENTIFIED BY
'rootpassword' WITH GRANT OPTION;
MariaDB [(none)]> FLUSH PRIVILEGES;
MariaDB [(none)]> quit
```

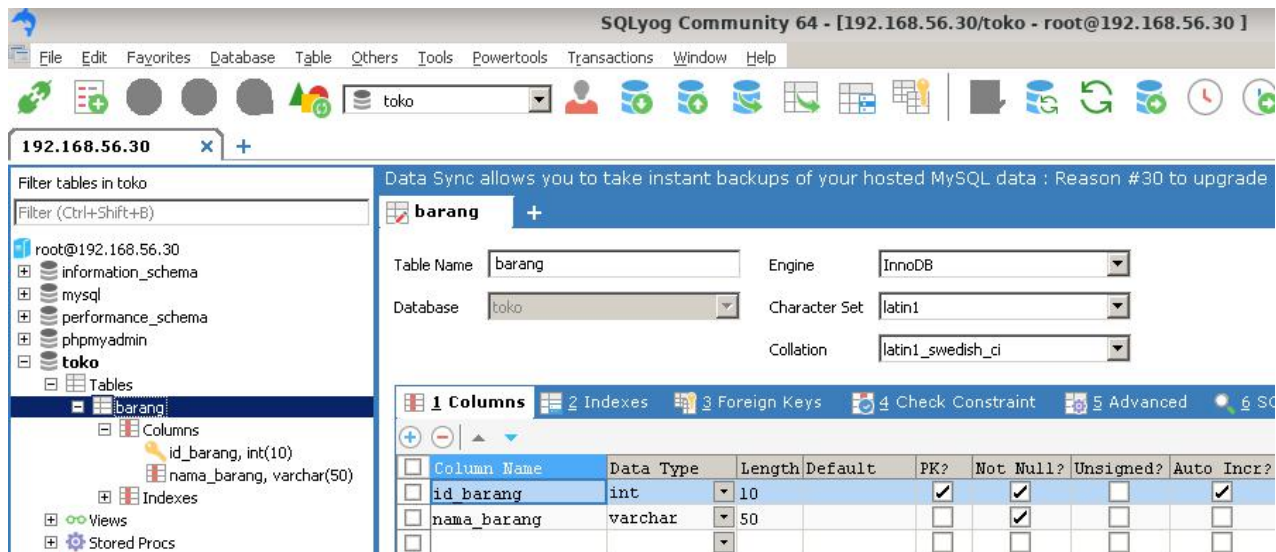
PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

Remote-lah database server tersebut menggunakan *SQLyog* di *Client Windows* menggunakan *user* dan *password* yang sudah dibuat seperti pada gambar 3 di bawah ini.



Gambar 3. Remote Database Debian Server menggunakan *SQLYog*

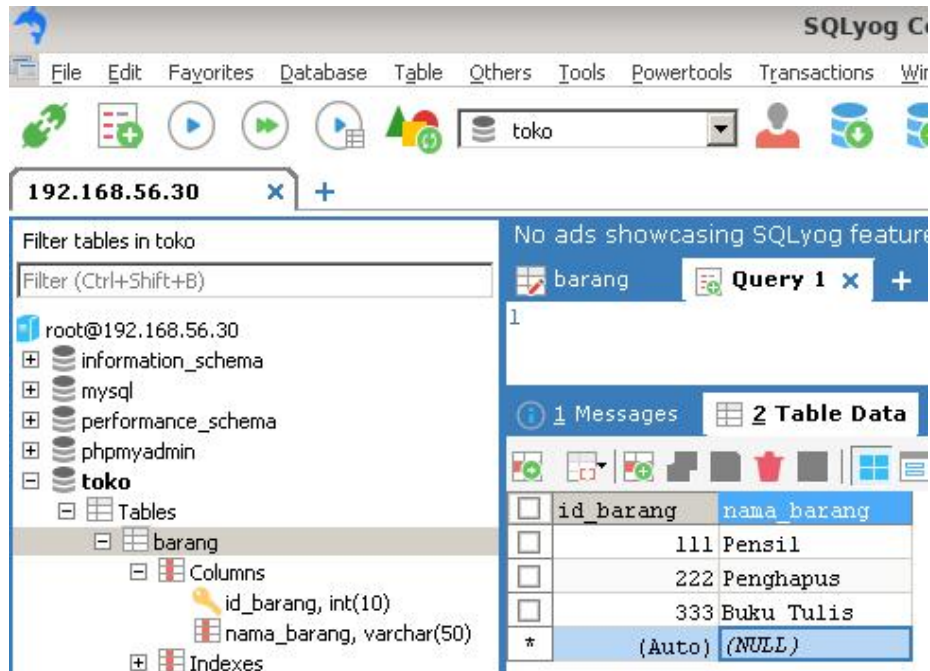
Buat database “*toko*” dan tabel “*barang*” dengan 2 kolom (*id_barang* dan *nama_barang*), tipe data, *length*, dll seperti pada gambar 4 di bawah ini.



Gambar 4. Tabel “*barang*” pada database “*toko*”

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

Isi data pada tabel “barang” seperti pada gambar 5 di bawah ini.



The screenshot shows the SQLyog interface. On the left, the 'Database' pane shows the 'toko' database selected, with the 'barang' table expanded to show its columns: 'id_barang' (int(10)) and 'nama_barang' (varchar(50)). On the right, the 'Table Data' pane displays the following data:

id_barang	nama_barang
111	Pensil
222	Penghapus
333	Buku Tulis
(Auto)	(NULL)

Gambar 5. Data pada tabel “barang”

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

3.1.2 *Source code Database.php* pada Debian Server

```
Database.php x server.php x RPCClient.php x proses.php x index.php x
1 <?php
2 error_reporting(1); // error ditampilkan
3
4 class Database
5 { private $host="localhost";
6   private $dbname="toko";
7   private $user="root";
8   private $password="root";
9   private $port="3306";
10  private $conn;
11
12  // function yang pertama kali di-load saat class dipanggil
13  public function __construct()
14  { // koneksi database
15    try
16    { $this->conn = new PDO("mysql:host=$this->host;port=$this->port;dbname=$this->dbname;charset=utf8",$this->user,$this->password);
17    } catch (PDOException $e)
18    { echo "Koneksi gagal";
19    }
20  }
21
22  public function tampil_data($id_barang)
23  { $query = $this->conn->prepare("select id_barang,nama_barang from barang where id_barang=?");
24    $query->execute(array($id_barang));
25    // mengambil satu data dengan fetch
26    $data = $query->fetch(PDO::FETCH_ASSOC);
27    // mengembalikan data
28    return $data;
29    // hapus variable dari memory
30    $query->closeCursor();
31    unset($id_barang,$data);
32  }
33
34  public function tampil_semua_data()
35  { $query = $this->conn->prepare("select id_barang, nama_barang from barang order by id_barang");
36    $query->execute();
37    // mengambil banyak data dengan fetchAll
38    $data = $query->fetchAll(PDO::FETCH_ASSOC);
39    return $data;
40    $query->closeCursor();
41    unset($data);
42  }
43
44  public function tambah_data($data)
45  { $query = $this->conn->prepare("insert ignore into barang (id_barang,nama_barang) values (?,?)");
46    $query->execute(array($data['id_barang'],$data['nama_barang']));
47    $query->closeCursor();
48    unset($data);
49  }
50
51  public function ubah_data($data)
52  { $query = $this->conn->prepare("update barang set nama_barang=? where id_barang=?");
53    $query->execute(array($data['nama_barang'],$data['id_barang']));
54    $query->closeCursor();
55    unset($data);
56  }
57
58  public function hapus_data($id_barang)
59  { $query = $this->conn->prepare("delete from barang where id_barang=?");
60    $query->execute(array($id_barang));
61    $query->closeCursor();
62    unset($id_barang);
63  }
64 }
65 ?>
66
```

Gambar 6. *Source code Database.php* pada Debian Server

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

3.1.3 *Source code server.php* pada Debian Server



```
1 <?php
2 error_reporting(1); // error ditampilkan
3 header('Content-Type: text/xml; charset=UTF-8');
4
5 include "Database.php";
6 // buat objek baru dari class Database
7 $abc = new Database();
8
9 // function untuk menghapus selain huruf dan angka
10 function filter($data)
11 { $data = preg_replace('/[^a-zA-Z0-9]/', '', $data);
12   return $data;
13   unset($data);
14 }
15
16 if ($_SERVER['REQUEST_METHOD'] == 'POST')
17 { $input = file_get_contents("php://input");
18   $data = xmlrpc_decode($input);
19
20   $aksi = $data[0]['aksi'];
21   $id_barang = $data[0]['id_barang'];
22   $nama_barang = $data[0]['nama_barang'];
23
24   if ($aksi == 'tambah')
25   { $data2=array('id_barang' => $id_barang,
26                 'nama_barang' => $nama_barang
27                 );
28     $abc->tambah_data($data2);
29   } elseif ($aksi == 'ubah')
30   { $data2=array('id_barang' => $id_barang,
31                 'nama_barang' => $nama_barang
32                 );
33     $abc->ubah_data($data2);
34   } elseif ($aksi == 'hapus')
35   { $abc->hapus_data($id_barang);
36   }
37
38 // hapus variable dari memory
39 unset($input,$data,$data2,$id_barang,$nama_barang,$aksi);
40
41 } elseif ($_SERVER['REQUEST_METHOD'] == 'GET')
42 { if ( ($_GET['aksi']=='tampil') and (isset($_GET['id_barang'])) )
43   { $id_barang = filter($_GET['id_barang']);
44     $data=$abc->tampil_data($id_barang);
45     $xml = xmlrpc_encode($data);
46     echo $xml;
47
48   } else //menampilkan semua data
49   { $data = $abc->tampil_semua_data();
50     $xml = xmlrpc_encode($data);
51     echo $xml;
52   }
53
54   unset($xml,$query,$id_barang,$data);
55 }
56 ?>
```

Gambar 7. *Source code server.php* pada Debian Server

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

Akses melalui *web browser* <http://192.168.56.xx/rpc-xml-toko/server/server.php> sehingga tampil laman *Server* yang menampilkan semua data format *XML* seperti pada gambar 8 di bawah ini.

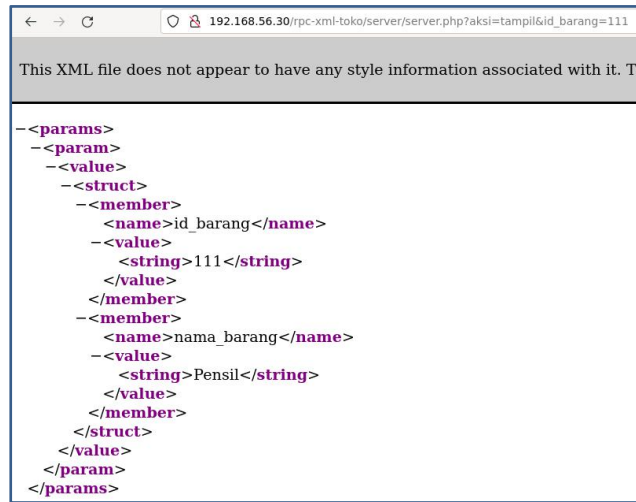


```
-<params>
- <param>
  - <value>
    - <array>
      - <data>
        - <value>
          - <struct>
            - <member>
              <name>id_barang</name>
              - <value>
                <string>111</string>
              </value>
            </member>
            - <member>
              <name>nama_barang</name>
              - <value>
                <string>Pensil</string>
              </value>
            </member>
          </struct>
        </value>
      - <value>
        - <struct>
          - <member>
            <name>id_barang</name>
            - <value>
              <string>222</string>
            </value>
          </member>
          - <member>
            <name>nama_barang</name>
            - <value>
              <string>Penghapus</string>
            </value>
          </member>
        </struct>
      </value>
    </array>
  </value>
</param>
</params>
```

Gambar 8. Laman *Server* menampilkan semua data barang format *XML*

Untuk menampilkan satu data barang dengan format *XML*, misalnya `id_barang=111`, akses http://192.168.56.xx/rpc-xml-toko/server/server.php?aksi=tampil&id_barang=111 pada *web browser* sehingga terlihat seperti pada gambar 9 di bawah ini.

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG



Gambar 9. Menampilkan satu data barang format XML

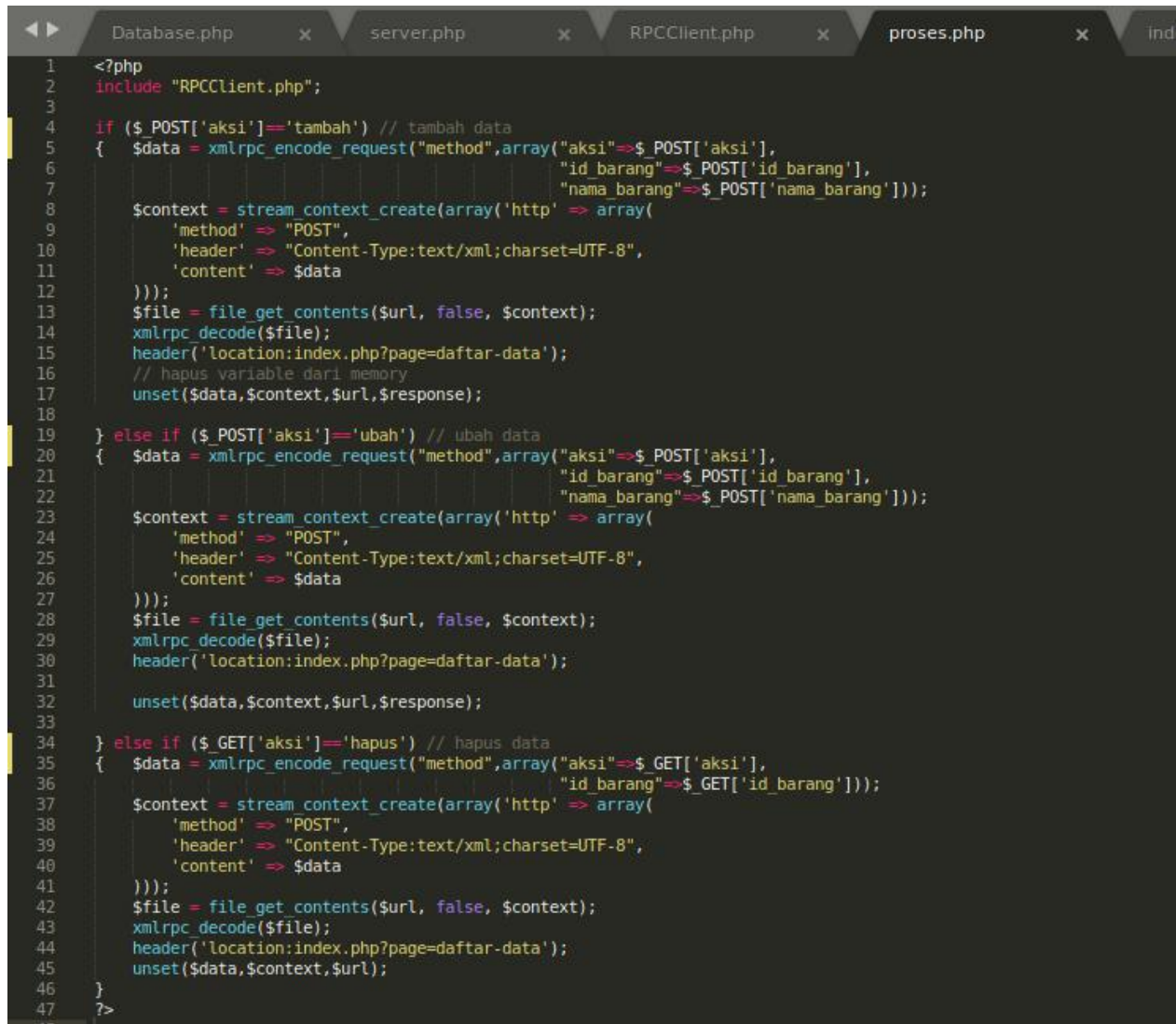
3.1.4 Source code RPCClient.php di Windows

```
RPCClient.php
1 <?php
2 error_reporting(1); // error ditampilkan
3
4 class RPCClient
5 { private $url;
6
7     // function yang pertama kali di-load saat class dipanggil
8     public function __construct($url)
9     { $this->url = $url;
10      unset($url);
11     }
12
13     // function untuk menghapus selain huruf dan angka
14     public function filter($data)
15     { $data = preg_replace('/[^a-zA-Z0-9]/', '', $data);
16       return $data;
17       unset($data);
18     }
19
20     public function tampil semua_data()
21     { $context = stream_context_create(array('http' => array(
22         'method' => "GET",
23         'header' => "Content-Type:text/xml;charset=UTF-8"
24     )));
25     $response = file_get_contents($this->url, false, $context);
26     $data = xmlrpc_decode($response);
27     return $data;
28     unset($context,$response,$data);
29     }
30
31     public function tampil_data($id_barang)
32     { $id_barang = $this->filter($id_barang);
33       $context = stream_context_create(array('http' => array(
34         'method' => "GET",
35         'header' => "Content-Type:text/xml;charset=UTF-8"
36     )));
37     $response = file_get_contents($this->url."?id_barang=".$id_barang."&aksi=tampil", false, $context);
38     $data = xmlrpc_decode($response);
39     return $data;
40     // hapus variable dari memory
41     unset($id_barang,$context,$response,$data);
42     }
43
44     // function yang terakhir kali di-load saat class dipanggil
45     public function __destruct()
46     { unset($this->url);
47     }
48 }
49
50 // url server
51 $url = 'http://192.168.56.30/rpc-xml-toko/server/server.php';
52
53 // buat objek baru dari class Client
54 $abc = new RPCClient($url);
55 ?>
56
```

Gambar 10. Source code RPCClient.php di Windows

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

3.1.5 *Source code proses.php di Windows*

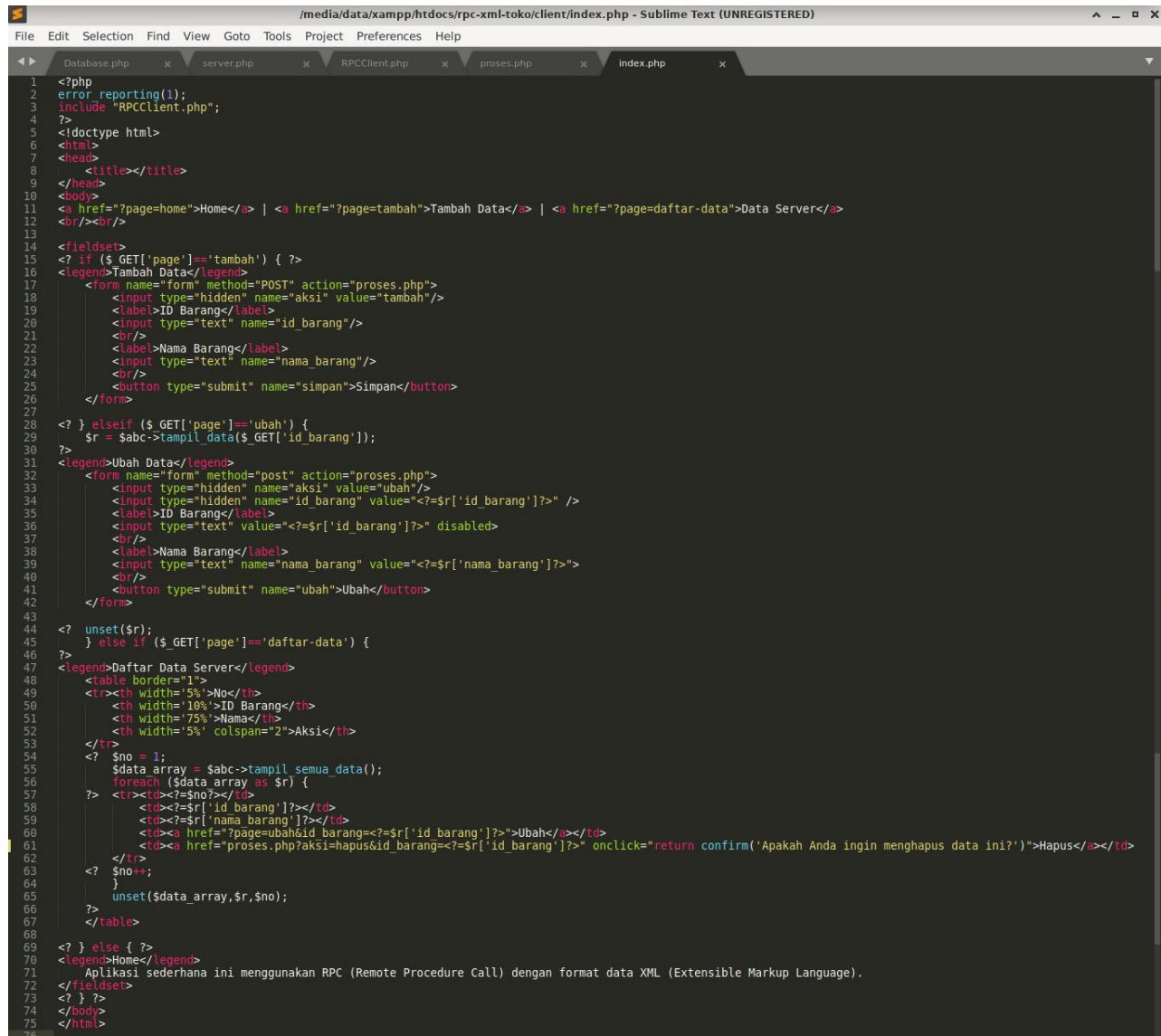


```
1 <?php
2 include "RPCClient.php";
3
4 if ($_POST['aksi']=='tambah') // tambah data
5 { $data = xmlrpc_encode_request("method",array("aksi"=>$_POST['aksi'],
6                                     "id_barang"=>$_POST['id_barang'],
7                                     "nama_barang"=>$_POST['nama_barang']));
8
9     $context = stream_context_create(array('http' => array(
10         'method' => "POST",
11         'header' => "Content-Type:text/xml;charset=UTF-8",
12         'content' => $data
13     )));
14     $file = file_get_contents($url, false, $context);
15     xmlrpc_decode($file);
16     header('location:index.php?page=daftar-data');
17     // hapus variable dari memory
18     unset($data,$context,$url,$response);
19 }
20 } else if ($_POST['aksi']=='ubah') // ubah data
21 { $data = xmlrpc_encode_request("method",array("aksi"=>$_POST['aksi'],
22                                     "id_barang"=>$_POST['id_barang'],
23                                     "nama_barang"=>$_POST['nama_barang']));
24
25     $context = stream_context_create(array('http' => array(
26         'method' => "POST",
27         'header' => "Content-Type:text/xml;charset=UTF-8",
28         'content' => $data
29     )));
30     $file = file_get_contents($url, false, $context);
31     xmlrpc_decode($file);
32     header('location:index.php?page=daftar-data');
33
34     unset($data,$context,$url,$response);
35 }
36 } else if ($_GET['aksi']=='hapus') // hapus data
37 { $data = xmlrpc_encode_request("method",array("aksi"=>$_GET['aksi'],
38                                     "id_barang"=>$_GET['id_barang']));
39
40     $context = stream_context_create(array('http' => array(
41         'method' => "POST",
42         'header' => "Content-Type:text/xml;charset=UTF-8",
43         'content' => $data
44     )));
45     $file = file_get_contents($url, false, $context);
46     xmlrpc_decode($file);
47     header('location:index.php?page=daftar-data');
48     unset($data,$context,$url);
49 }
50 }
51 ?>
```

Gambar 11. *Source code proses.php di Windows*

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

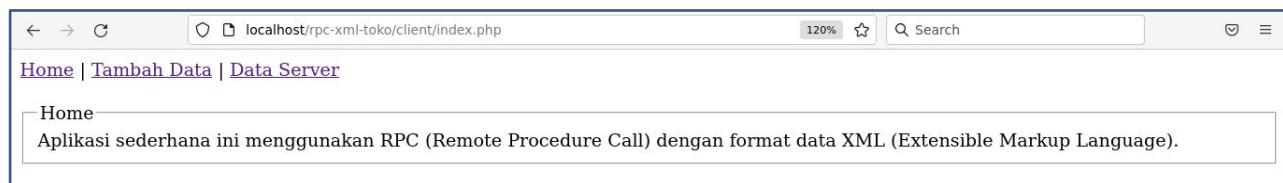
3.1.6 *Source code index.php* di *Windows*



```
1 <?php
2 error_reporting(1);
3 include "RPCClient.php";
4 ?>
5 <!doctype html>
6 <html>
7 <head>
8 <title></title>
9 </head>
10 <body>
11 <a href="?page=home">Home</a> | <a href="?page=tambah">Tambah Data</a> | <a href="?page=daftar-data">Data Server</a>
12 <br/><br/>
13
14 <fieldset>
15 <? if ($ GET['page']=='tambah') { ?>
16 <legend>Tambah Data</legend>
17 <form name="form" method="POST" action="proses.php">
18 <input type="hidden" name="aksi" value="tambah"/>
19 <label>ID Barang</label>
20 <input type="text" name="id_barang"/>
21 <br/>
22 <label>Nama Barang</label>
23 <input type="text" name="nama_barang"/>
24 <br/>
25 <button type="submit" name="simpan">Simpan</button>
26 </form>
27
28 <? } elseif ($ GET['page']=='ubah') {
29 $r = $abc->tampil_data($ GET['id_barang']);
30 ?>
31 <legend>Ubah Data</legend>
32 <form name="form" method="post" action="proses.php">
33 <input type="hidden" name="aksi" value="ubah"/>
34 <input type="hidden" name="id_barang" value="<?=$r['id_barang']?>" />
35 <label>ID Barang</label>
36 <input type="text" value="<?=$r['id_barang']?>" disabled/>
37 <br/>
38 <label>Nama Barang</label>
39 <input type="text" name="nama_barang" value="<?=$r['nama_barang']?>" />
40 <br/>
41 <button type="submit" name="ubah">Ubah</button>
42 </form>
43
44 <? unset($r);
45 } else if ($ GET['page']=='daftar-data') {
46 ?>
47 <legend>Daftar Data Server</legend>
48 <table border="1">
49 <tr><th width="5%">No</th>
50 <th width="10%">ID Barang</th>
51 <th width="75%">Nama</th>
52 <th width="5%" colspan="2">Aksi</th>
53 </tr>
54 <? $no = 1;
55 $data array = $abc->tampil_semua_data();
56 foreach ($data array as $r) {
57 <tr><td><?=$no?></td>
58 <td><?=$r['id_barang']?></td>
59 <td><?=$r['nama_barang']?></td>
60 <td><a href="?page=ubah&id_barang=<?=$r['id_barang']?>">Ubah</a></td>
61 <td><a href="proses.php?aksi=hapus&id_barang=<?=$r['id_barang']?>" onclick="return confirm('Apakah Anda ingin menghapus data ini?')">Hapus</a></td>
62 </tr>
63 <? $no++;
64 }
65 }
66 <? unset($data_array,$r,$no);
67 </table>
68
69 <? } else { ?>
70 <legend>Home</legend>
71 Aplikasi sederhana ini menggunakan RPC (Remote Procedure Call) dengan format data XML (Extensible Markup Language).
72 </fieldset>
73 <? } ?>
74 </body>
75 </html>
76
```

Gambar 12. *Source code index.php* di *Windows*

Akses melalui *web browser* **http://localhost/rpc-xml-toko/client/index.php** di *Windows*. Coba jalankan aplikasi sederhana ini dengan menambah, menampilkan, mengubah dan menghapus datanya. Data akan dikirim dari *Windows* ke *Debian Server* dan data hanya disimpan di *database Debian Server*. Data yang tersimpan tersebut akan ditampilkan ke *Client di Windows*.



Gambar 13. Laman *Client*

MODUL 4

Simple Object Access Protocol (SOAP)

4.1 Pengertian

SOAP (Simple Object Access Protocol) adalah standar untuk bertukar pesan-pesan berbasis *XML* melalui jaringan komputer atau sebuah jalan untuk program yang berjalan pada suatu sistem operasi (OS) untuk berkomunikasi dengan program pada OS yang sama maupun berbeda dengan menggunakan HTTP dan XML sebagai mekanisme untuk pertukaran data.

SOAP menspesifikan secara jelas bagaimana cara untuk meng-encode header HTTP dan file XML sehingga program pada suatu komputer dapat memanggil program pada komputer lain dan mengirimkan informasi, dan bagaimana program yang dipanggil memberikan tanggapan.

SOAP adalah protokol ringan yang ditujukan untuk pertukaran informasi struktur pada lingkup desentralisasi, dan terdistribusi. *SOAP* menggunakan teknologi XML untuk mendefinisikan rangka kerja pemesanan terekstrensi di mana menyediakan konstruksi pesan yang dapat dipertukarkan pada protokol berbeda. Rangka kerja dirancang bebas dari model pemrograman dan spesifikasi implementasi semantik.

4.2 Aplikasi SOAP - CRUD

Aplikasi Toko menggunakan SOAP - CRUD sederhana, dimana **data akan dikirim dari Client (Windows dengan IP Address 192.168.56.1) ke Server (Debian Server dengan IP Address 192.168.56.xx) dan data hanya disimpan pada database Debian Server.** Database tetap menggunakan database "toko" dan tabel "barang" pada Debian Server. Buat folder */opt/lampp/htdocs/soap-toko/soap-server/* pada Debian Server lalu buat file *Database.php* dan *server.php*. Buat juga folder *xampp\htdocs\soap-toko\soap-client* pada Windows lalu buat file *Client.php*, *proses.php* serta *index.php*. Akses *phpinfo* pada Client (Windows) dan Server (Debian Server) serta pastikan *PHP extension xml* dan *soap* sudah aktif seperti pada gambar 1 di bawah ini.

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

soap		
Soap Client	enabled	
Soap Server	enabled	
Directive	Local Value	Master Value
soap.wsdl_cache	1	1
soap.wsdl_cache_dir	/tmp	/tmp
soap.wsdl_cache_enabled	1	1
soap.wsdl_cache_limit	5	5
soap.wsdl_cache_ttl	86400	86400

Gambar 1. *PHP Extension soap*

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

extension=php_soap.dll

lalu *reboot Apache*

Untuk *Server (Debian Server)*, pastikan juga *PHP extension PDO* dan *pdo_mysql* sudah aktif seperti pada gambar 2 di bawah ini.

PDO		
PDO support	enabled	
PDO drivers	mysql, pgsql, sqlite	
pdo_mysql		
PDO Driver for MySQL	enabled	
Client API version	mysqlnd 5.0.11-dev - 20120503 - \$id: 76b08b24596e12d4553bd41fc93cccd5bac2fe7a \$	
Directive	Local Value	Master Value
pdo_mysql.default_socket	/opt/lampp/var/mysql/mysql.sock	/opt/lampp/var/mysql/mysql.sock

Gambar 2. *PHP Extension PDO dan pdo_mysql*

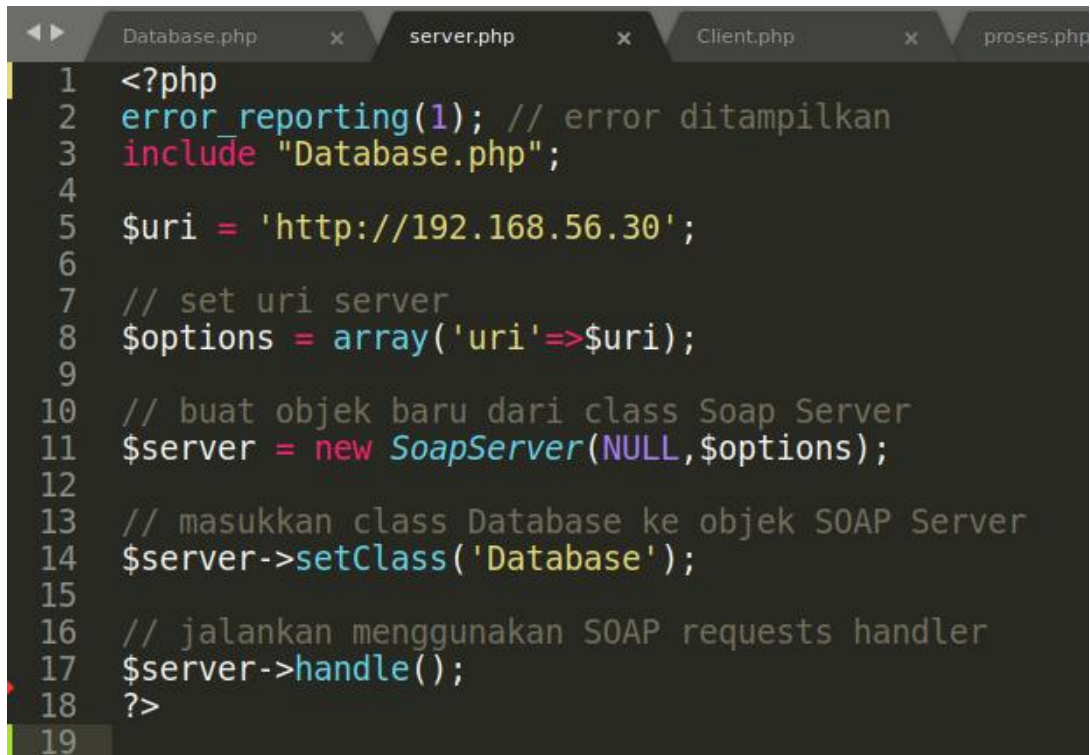
PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

4.2.1 Source code *Database.php* di Debian Server

```
Database.php x server.php x RPCClient.php x proses.php x index.php x
1 <?php
2 error_reporting(1); // error ditampilkan
3
4 class Database
5 { private $host="localhost";
6   private $dbname="toko";
7   private $user="root";
8   private $password="root";
9   private $port="3306";
10  private $conn;
11
12  // function yang pertama kali di-load saat class dipanggil
13  public function __construct()
14  { // koneksi database
15    try
16    { $this->conn = new PDO("mysql:host=$this->host;port=$this->port;dbname=$this->dbname;charset=utf8",$this->user,$this->password);
17    } catch (PDOException $e)
18    { echo "Koneksi gagal";
19    }
20  }
21
22  public function tampil_data($id_barang)
23  { $query = $this->conn->prepare("select id_barang,nama_barang from barang where id_barang=?");
24    $query->execute(array($id_barang));
25    // mengambil satu data dengan fetch
26    $data = $query->fetch(PDO::FETCH_ASSOC);
27    // mengembalikan data
28    return $data;
29    // hapus variable dari memory
30    $query->closeCursor();
31    unset($id_barang,$data);
32  }
33
34  public function tampil_semua_data()
35  { $query = $this->conn->prepare("select id_barang, nama_barang from barang order by id_barang");
36    $query->execute();
37    // mengambil banyak data dengan fetchAll
38    $data = $query->fetchAll(PDO::FETCH_ASSOC);
39    return $data;
40    $query->closeCursor();
41    unset($data);
42  }
43
44  public function tambah_data($data)
45  { $query = $this->conn->prepare("insert ignore into barang (id_barang,nama_barang) values (?,?)");
46    $query->execute(array($data['id_barang'],$data['nama_barang']));
47    $query->closeCursor();
48    unset($data);
49  }
50
51  public function ubah_data($data)
52  { $query = $this->conn->prepare("update barang set nama_barang=? where id_barang=?");
53    $query->execute(array($data['nama_barang'],$data['id_barang']));
54    $query->closeCursor();
55    unset($data);
56  }
57
58  public function hapus_data($id_barang)
59  { $query = $this->conn->prepare("delete from barang where id_barang=?");
60    $query->execute(array($id_barang));
61    $query->closeCursor();
62    unset($id_barang);
63  }
64 }
65 ?>
66
```

Gambar 3. Source code *Database.php* di Debian Server

4.2.2 *Source code server.php* di Debian Server



```
1 <?php
2 error_reporting(1); // error ditampilkan
3 include "Database.php";
4
5 $uri = 'http://192.168.56.30';
6
7 // set uri server
8 $options = array('uri'=>$uri);
9
10 // buat objek baru dari class Soap Server
11 $server = new SoapServer(NULL,$options);
12
13 // masukkan class Database ke objek SOAP Server
14 $server->setClass('Database');
15
16 // jalankan menggunakan SOAP requests handler
17 $server->handle();
18 ?>
19
```

Gambar 4. *Source code server.php* di Debian Server

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

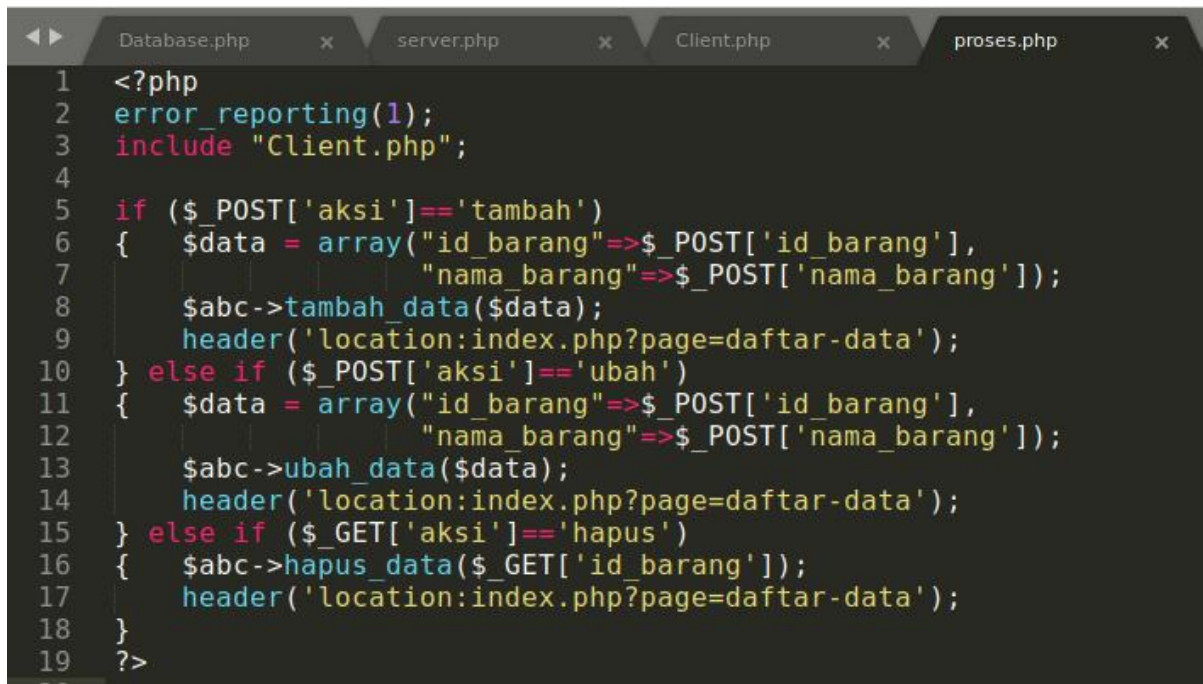
4.2.3 Source code Client.php di Windows

```
Database.php x server.php x Client.php x proses.php x index.php
1 <?php
2 error_reporting(1); // error ditampilkan
3 class Client
4 { private $options,$api;
5
6 // function yang pertama kali di-load saat class dipanggil
7 public function __construct($uri,$location)
8 { $this->options = array('location' => $location,'uri' => $uri);
9
10 // buat objek baru dari class SOAP Client
11 $this->api = new SoapClient(NULL, $this->options);
12
13 // menghapus variabel dari memory
14 unset($uri,$location);
15 }
16
17 // function untuk menghapus selain huruf dan angka
18 public function filter($data)
19 { $data = preg_replace('/[^a-zA-Z0-9]/', '', $data);
20 return $data;
21 unset($data);
22 }
23
24 public function tampil_semua_data()
25 { $data = $this->api->tampil_semua_data();
26 return $data;
27 unset($data);
28 }
29
30 public function tampil_data($id_barang)
31 { $id_barang = $this->filter($id_barang);
32 $data = $this->api->tampil_data($id_barang);
33 return $data;
34 unset($id_barang,$data);
35 }
36
37 public function tambah_data($data)
38 { $this->api->tambah_data($data);
39 unset($data);
40 }
41
42 public function ubah_data($data)
43 { $this->api->ubah_data($data);
44 unset($data);
45 }
46
47 public function hapus_data($id_barang)
48 { $this->api->hapus_data($id_barang);
49 unset($id_barang);
50 }
51
52 // function yang terakhir kali di-load saat class dipanggil
53 public function __destruct()
54 { // hapus variable dari memory
55 unset($this->options,$this->api);
56 }
57 }
58
59 // uri dan location server
60 $uri = 'http://192.168.56.30';
61 $location = $uri.'/soap-toko/soap-server/server.php';
62 // buat objek baru dari class Client
63 $abc= new Client($uri,$location);
64 ?>
```

Gambar 5. Source code Client.php di Windows

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

4.2.4 *Source code proses.php di Windows*



```
1 <?php
2 error_reporting(1);
3 include "Client.php";
4
5 if ($_POST['aksi']=='tambah')
6 {   $data = array("id_barang"=>$_POST['id_barang'],
7                 "nama_barang"=>$_POST['nama_barang']);
8     $abc->tambah_data($data);
9     header('location:index.php?page=daftar-data');
10 } else if ($_POST['aksi']=='ubah')
11 {   $data = array("id_barang"=>$_POST['id_barang'],
12                 "nama_barang"=>$_POST['nama_barang']);
13     $abc->ubah_data($data);
14     header('location:index.php?page=daftar-data');
15 } else if ($_GET['aksi']=='hapus')
16 {   $abc->hapus_data($_GET['id_barang']);
17     header('location:index.php?page=daftar-data');
18 }
19 ?>
```

Gambar 6. *Source code proses.php di Windows*

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

4.2.5 Source code *index.php* di *Windows*

```
1 <?php
2 error_reporting(1);
3 include "Client.php";
4 ?>
5 <!doctype html>
6 <html>
7 <head>
8 <title></title>
9 </head>
10 <body>
11 <a href="?page=home">Home</a> | <a href="?page=tambah">Tambah Data</a> | <a href="?page=daftar-data">Data Server</a>
12 <br/><br/>
13
14 <fieldset>
15 <? if ($ GET['page']=='tambah') { ?>
16 <legend>Tambah Data</legend>
17 <form name="form" method="POST" action="proses.php">
18 <input type="hidden" name="aksi" value="tambah"/>
19 <label>ID Barang</label>
20 <input type="text" name="id_barang"/>
21 <br/>
22 <label>Nama Barang</label>
23 <input type="text" name="nama_barang"/>
24 <br/>
25 <button type="submit" name="simpan">Simpan</button>
26 </form>
27
28 <? elseif ($ GET['page']=='ubah') {
29 $r = $abc->tampil_data($ GET['id_barang']);
30 ?>
31 <legend>Ubah Data</legend>
32 <form name="form" method="post" action="proses.php">
33 <input type="hidden" name="aksi" value="ubah"/>
34 <input type="hidden" name="id_barang" value="<?=$r['id_barang']?>" />
35 <label>ID Barang</label>
36 <input type="text" value="<?=$r['id_barang']?>" disabled>
37 <br/>
38 <label>Nama Barang</label>
39 <input type="text" name="nama_barang" value="<?=$r['nama_barang']?>">
40 <br/>
41 <button type="submit" name="ubah">Ubah</button>
42 </form>
43
44 <? unset($r);
45 } else if ($ GET['page']=='daftar-data') {
46 ?>
47 <legend>Daftar Data Server</legend>
48 <table border="1">
49 <tr><th width="5%">No</th>
50 <th width="10%">ID Barang</th>
51 <th width="75%">Nama</th>
52 <th width="5%" colspan="2">Aksi</th>
53 </tr>
54 <? $no = 1;
55 $data array = $abc->tampil_semua_data();
56 foreach ($data array as $r) {
57 ?> <tr><td><?=$no?></td>
58 <td><?=$r['id_barang']?></td>
59 <td><?=$r['nama_barang']?></td>
60 <td><a href="?page=ubah&id_barang=<?=$r['id_barang']?>">Ubah</a></td>
61 <td><a href="proses.php?aksi=hapus&id_barang=<?=$r['id_barang']?>" onclick="return confirm('Apakah Anda ingin menghapus data ini?')">Hapus</a></td>
62 </tr>
63 <? $no++;
64 }
65 unset($data_array,$r,$no);
66 ?>
67 </table>
68
69 <? } else { ?>
70 <legend>Home</legend>
71 Aplikasi sederhana ini menggunakan Web Service SOAP (Simple Object Access Protocol) dengan format data XML (Extensible Markup Language).
72 </fieldset>
73 <? } ?>
74 </body>
75 </html>
```

Gambar 7. Source code *index.php* di *Windows*

Akses melalui web browser *http://localhost/soap-toko/soap-client/index.php* di *Windows*. Coba jalankan aplikasi sederhana ini dengan menambah, menampilkan, mengubah dan menghapus datanya. Data akan dikirim dari *Windows* ke *Debian Server* dan data hanya disimpan di *database Debian Server*. Data yang tersimpan tersebut akan ditampilkan ke *Client di Windows*.



Gambar 8. Laman *Client*

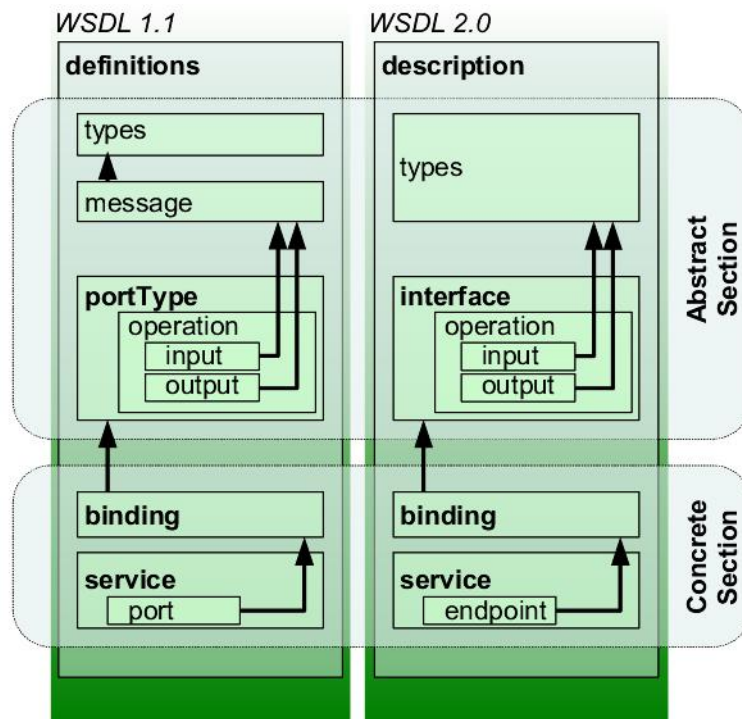
MODUL 5

Web Services Description Language (WSDL)

5.1 Pengertian

Web Services Description Language (WSDL) adalah sebuah dokumen yang berbasis *XML*. *WSDL* sepenuhnya berisi kode *XML* yang digunakan untuk mendeskripsikan suatu layanan yang ada di *Web Service*. *WSDL* juga merupakan rekomendasi dari *W3C*. Secara spesifiknya, *WSDL* ini isinya berupa lokasi layanan dan operasi layanan atau method, dan bagaimana cara mengakses layanan tersebut.

WSDL menyediakan kebutuhan ini dengan mendefinisikan bentuk *XML* untuk mendeskripsikan layanan jaringan sebagai kumpulan-kumpulan titik akhir (end-point) komunikasi yang mampu melakukan pertukaran pesan.



Gambar 1. Struktur *WSDL*

WSDL merupakan bahasa standard yang menyediakan mekanisme untuk mendeskripsikan *Service* yang disediakan oleh sistem (*Web Service*). *WSDL* mendeskripsikan *service* dengan menggunakan elemen sebagai berikut :

- **Type**, yaitu tipe data yang digunakan sebagai argumen dan *return type*.
- **Message**, digunakan untuk mempresentasikan definisi data yang ditransmisikan.

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

- **Port Type**, merupakan sekumpulan operasi yang didukung oleh satu atau lebih *end-point*.
- **Binding**, digunakan untuk mendefinisikan protokol dan format pertukaran data untuk operasi yang didefinisikan oleh *Port type*.
- **Port**, digunakan untuk menspesifikasikan *end-point* yang digunakan untuk *binding*.
- **Service**, merupakan koleksi *end-point* yang berkaitan yang disediakan oleh *Web Service*.
- **Operation**, digunakan untuk mendefinisikan kemampuan yang didukung oleh *service* tertentu.

Elemen – elemen *WSDL*

Elemen berikut adalah elemen penting di dalam *WSDL*:

1. *Message*, sesuatu yang abstrak, definisi tipe data yang akan dikomunikasikan

```
<message name='getItemCountRequest'>
<part name='upc' type='xsd:string' />
</message>
<message name='getItemCountResponse'>
<part name='Result' type='xsd:integer' />
</message>
```

2. *Port Type*, mendeskripsikan sebuah *web service*, operasi-operasi yang dapat dijalankan, dan pesan-pesan yang dilibatkan pada *Web Service*.

```
<portType name='InventoryPortType'>
<operation name='getItemCount'>
<input message='tns:getItemCountRequest' />
<output message='tns:getItemCountResponse' />
</operation>
</portType>
```

3. *Port*, Titik akhir tunggal (*single end-point*) yang didefinisikan sebagai sebuah '*binding*' dan alamat jaringan (*network address*).

```
<port name='InventoryPort' binding='InventoryBinding'>
<soap:address location='http://localhost/soap/server/server.php' />
</port>
```

4. *Service*, Sekumpulan *end-point* yang saling berhubungan, akan menunjukkan file/path mana yang akan ditempatkan pada file *WSDL* ini

```
<service name='InventoryService'>
<port name='InventoryPort' binding='InventoryBinding'>
<soap:address location='http://localhost/soap/server/server.php' />
</port>
</service>
```

5. *Operation*, deskripsi abstrak dari suatu aksi yang didukung oleh *service*. Pada dasarnya menunjukkan nama operasi *web service* dan pesan *input output*

```
<operation name='getItemCount'>
<input message='tns:getItemCountRequest' />
```

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

```
<output message='tns:getItemCountResponse' />
</operation>
```

6. *Binding*, protokol komunikasi yang digunakan oleh *web service*.

```
<binding name='InventoryBinding' type='tns:InventoryPortType'>
<soap:binding style='rpc' transport='http://schemas.xmlsoap.org/soap/http' />
<operation name='getItemCount'>
<soap:operation soapAction='urn:xmethods-delayed-quotes#getItemCount' />
<input>
<soap:body use='encoded' namespace='urn:xmethods-delayed-quotes'
encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' />
</input>
<output>
<soap:body use='encoded' namespace='urn:xmethods-delayed-quotes'
encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' />
</output>
</operation>
</binding>
```

5.2 Aplikasi WSDL - CRUD

Aplikasi Toko menggunakan *WSDL - CRUD* sederhana, dimana **data akan dikirim dari Client (Windows dengan IP Address 192.168.56.1) ke Server (Debian Server dengan IP Address 192.168.56.xx) dan data hanya disimpan pada database Debian Server. Database tetap menggunakan database “toko” dan tabel “barang” pada Debian Server. Langkah selanjutnya adalah sebagai berikut:**

1. Download **nusoap.php** (*NuSoap library*) di https://drive.google.com/file/d/1NR_IMvGQtjUS3SFSSMfTrsHGokKYpzyQ/view
2. Buat folder `/opt/lampp/htdocs/wsd-toko/server/` pada *Debian Server* lalu buat file **Database.php**, **server.php** dan **nusoap.php** (*NuSoap library*) yang sudah di-download.
3. Buat juga folder `xampp\htdocs\wsdl-toko\client-nusoap\` pada *Windows* lalu buat file **Client.php**, **proses.php**, **index.php** dan **nusoap.php** (*NuSoap library*) yang sudah di-download.
4. Akses *phpinfo* pada *Server (Debian Server)* serta pastikan *PHP extension xml, soap, PDO* dan **pdo_mysql** sudah aktif.
5. Akses *phpinfo* pada *Client (Windows)* serta pastikan *PHP extension xml* dan **soap** sudah aktif.

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

5.2.1 *Source code Database.php di Debian Server*

```
Database.php x
1 <?php
2 class Database
3 {
4     private $host="localhost";
5     private $dbname="toko";
6     private $user="root";
7     private $password="root";
8     private $port="3306";
9     private $conn;
10
11 // function yang pertama kali di-load saat class dipanggil
12 public function __construct()
13 {
14     // koneksi database
15     try
16     {
17         $this->conn = new PDO("mysql:host=$this->host;port=$this->port;dbname=$this->dbname;charset=utf8",$this->user,$this->password);
18     }
19     catch (PDOException $e)
20     {
21         echo "Koneksi gagal";
22     }
23 }
24
25 public function tampil_semua_data()
26 {
27     $query = $this->conn->prepare("select id_barang, nama_barang from barang order by id_barang");
28     $query->execute();
29     // mengambil banyak data dengan fetchAll
30     $data = $query->fetchAll(PDO::FETCH_ASSOC);
31     // mengembalikan data
32     return $data;
33     // hapus variable dari memory
34     $query->closeCursor();
35     unset($data);
36 }
37
38 public function tampil_data($id_barang)
39 {
40     $query = $this->conn->prepare("select id_barang,nama_barang from barang where id_barang=?");
41     $query->execute(array($id_barang));
42     // mengambil satu data dengan fetch
43     $data = $query->fetch(PDO::FETCH_ASSOC);
44     return $data;
45     $query->closeCursor();
46     unset($id_barang,$data);
47 }
48
49 public function tambah_data($data)
50 {
51     $query = $this->conn->prepare("insert ignore into barang (id_barang,nama_barang) values (?,?)");
52     $query->execute(array($data['id_barang'],$data['nama_barang']));
53     $query->closeCursor();
54     unset($data);
55 }
56
57 public function ubah_data($data)
58 {
59     $query = $this->conn->prepare("update barang set nama_barang=? where id_barang=?");
60     $query->execute(array($data['nama_barang'],$data['id_barang']));
61     $query->closeCursor();
62     unset($data);
63 }
64
65 public function hapus_data($id_barang)
66 {
67     $query = $this->conn->prepare("delete from barang where id_barang=?");
68     $query->execute(array($id_barang));
69     $query->closeCursor();
70     unset($id_barang);
71 }
72 }
73 ?>
```

Gambar 2. *Source code Database.php di Debian Server*

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

5.2.2 Source code *server.php* di Debian Server

```
server.php x
1 <?php
2 error_reporting(1); // error ditampilkan
3 require_once('nusoap.php');
4 require_once('Database.php');
5
6 // buat objek baru dari class NuSOAP Server
7 $server = new nusoap_server();
8
9 // configure WSDL file
10 $server->configureWSDL('WSDL Toko', 'urn:ServerWSDL');
11
12 $server->register('tampil_semua_data', // method name
13 array(), // no input parameters
14 array('output' => 'xsd:Array'), // output parameters
15 'urn:ServerWSDL', // namespace
16 'urn:ServerWSDL#tampil_semua_data', // soapaction
17 'rpc', // style
18 'encoded', // use
19 'tampil semua data' // documentation
20 );
21
22 $server->register('tampil_data', // method name
23 array('input' => 'xsd:String'), // input parameters
24 array('output' => 'xsd:Array'), // output parameters
25 'urn:ServerWSDL', // namespace
26 'urn:ServerWSDL#tampil_data', // soapaction
27 'rpc', // style
28 'encoded', // use
29 'tampil data' // documentation
30 );
31
32 $server->register('tambah_data', // method name
33 array('input' => 'xsd:Array'), // input parameters
34 array(), // no output parameters
35 'urn:ServerWSDL', // namespace
36 'urn:ServerWSDL#tambah_data', // soapaction
37 'rpc', // style
38 'encoded', // use
39 'tambah data' // documentation
40 );
41
42 $server->register('ubah_data', // method name
43 array('input' => 'xsd:Array'), // input parameters
44 array(), // no output parameters
45 'urn:ServerWSDL', // namespace
46 'urn:ServerWSDL#ubah_data', // soapaction
47 'rpc', // style
48 'encoded', // use
49 'ubah data' // documentation
50 );
51
52 $server->register('hapus_data', // method name
53 array('input' => 'xsd:String'), // input parameters
54 array(), // no output parameters
55 'urn:ServerWSDL', // namespace
56 'urn:ServerWSDL#hapus_data', // soapaction
57 'rpc', // style
58 'encoded', // use
59 'hapus data' // documentation
60 );
```


**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

```
61
62 // fungsi menghapus selain huruf dan angka
63 function filter($data)
64 {   $data = preg_replace('/[^a-zA-Z0-9]/','',$data);
65     return $data;
66     unset($data);
67 }
68
69 function tampil_semua_data()
70 {   $abc = new Database();
71     $data = $abc->tampil_semua_data();
72     return $data;
73     unset($abc,$data);
74 }
75
76 function tampil_data($id_barang)
77 {   $id_barang = filter($id_barang);
78     $abc = new Database();
79     $data = $abc->tampil_data($id_barang);
80     return $data;
81     unset($id_barang,$abc,$data);
82 }
83
84 function tambah_data($data)
85 {   $abc = new Database();
86     $data = $abc->tambah_data($data);
87     unset($abc,$data);
88 }
89
90 function ubah_data($data)
91 {   $abc = new Database();
92     $data = $abc->ubah_data($data);
93     unset($abc,$data);
94 }
95
96 function hapus_data($id_barang)
97 {   $id_barang = filter($id_barang);
98     $abc = new Database();
99     $data = $abc->hapus_data($id_barang);
100    unset($id_barang,$abc,$data);
101 }
102
103 $HTTP_RAW_POST_DATA = isset($HTTP_RAW_POST_DATA) ? $HTTP_RAW_POST_DATA : '';
104 $server->service($HTTP_RAW_POST_DATA);
105
106 unset($server);
107 ?>
```

Gambar 3. Source code server.php di Debian Server

Akses WSDL menggunakan web browser <http://192.168.56.xx/wsdl-toko/server/server.php>

The screenshot shows a web browser window with the title "WSDL Toko". On the left side, there is a list of operations: [tampil_semua_data](#), [tampil_data](#), [tambah_data](#), [ubah_data](#), and [hapus_data](#). The [tampil_semua_data](#) operation is selected, and its details are shown in a modal window on the right. The details include:

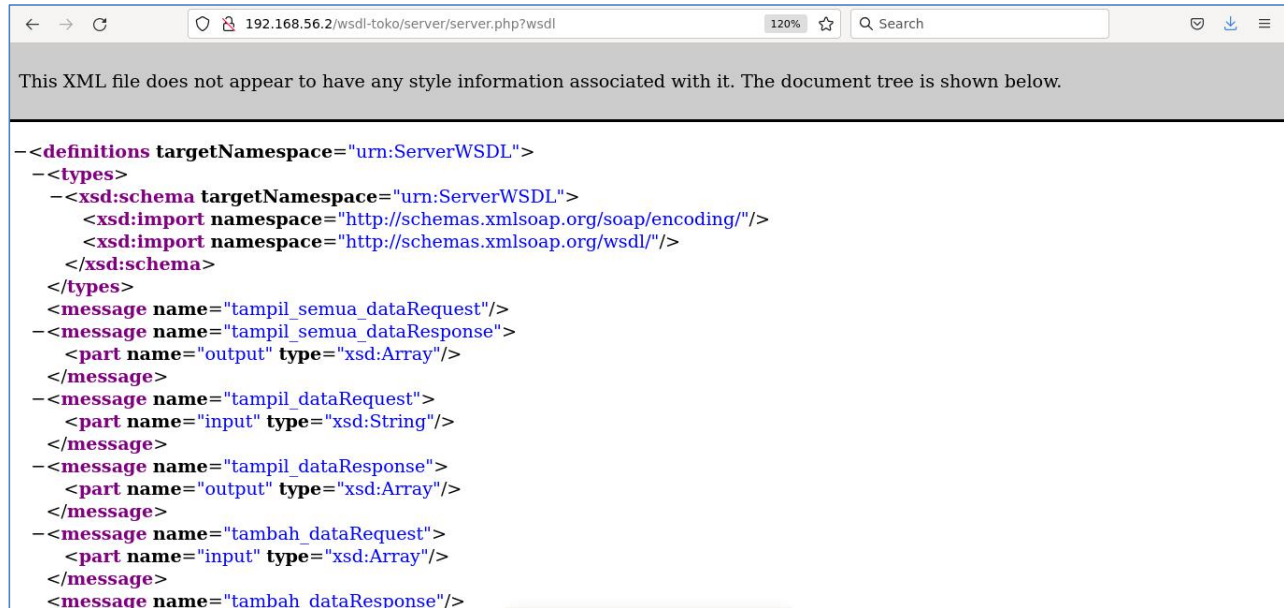
- Name: tampil_semua_data
- Binding: WSDL TokoBinding
- Endpoint: http://192.168.56.2/wsdl-toko/server/server.php
- SoapAction: urn:ServerWSDL#tampil_semua_data
- Style: rpc
- Input: use: encoded, namespace: urn:ServerWSDL, encodingStyle: http://schemas.xmlsoap.org/soap/encoding/, message: tampil_semua_dataRequest
- Output: use: encoded, namespace: urn:ServerWSDL, encodingStyle: http://schemas.xmlsoap.org/soap/encoding/, message: tampil_semua_dataResponse
- Output: xsd:Array
- Namespace: urn:ServerWSDL
- Transport: http://schemas.xmlsoap.org/soap/http
- Documentation: tampil_semua_data

Gambar 4. WSDL di Debian Server

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

Akses *WSDL* format *XML* menggunakan *web browser*

<http://192.168.56.xx/wsdl-toko/server/server.php?wsdl>



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
--<definitions targetNamespace="urn:ServerWSDL">
  --<types>
    --<xsd:schema targetNamespace="urn:ServerWSDL">
      <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding"/>
      <xsd:import namespace="http://schemas.xmlsoap.org/wsdl"/>
    </xsd:schema>
  </types>
  <message name="tampil_semua_dataRequest"/>
  --<message name="tampil_semua_dataResponse">
    <part name="output" type="xsd:Array"/>
  </message>
  --<message name="tampil_dataRequest">
    <part name="input" type="xsd:String"/>
  </message>
  --<message name="tampil_dataResponse">
    <part name="output" type="xsd:Array"/>
  </message>
  --<message name="tambah_dataRequest">
    <part name="input" type="xsd:Array"/>
  </message>
  <message name="tambah_dataResponse"/>
</definitions>
```

Gambar 5. *WSDL* format *XML* di *Debian Server*

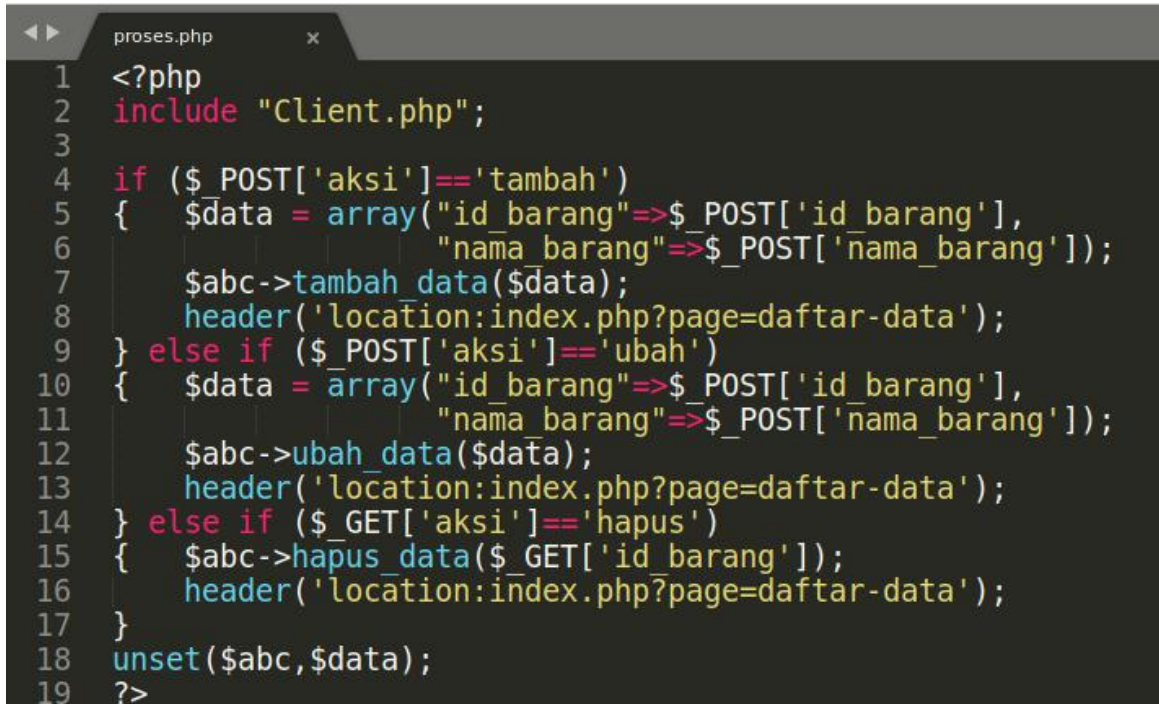
**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

5.2.3 Source code Client.php di Windows

```
Client.php
1 <?php
2 error_reporting(1); // error ditampilkan
3 require_once('nusoap.php');
4
5 class Client
6 { private $api;
7
8 // function yang pertama kali di-load saat class dipanggil
9 public function __construct($url)
10 { // buat objek baru dari class NuSOAP Client
11   $this->api = new nusoap_client($url, true);
12   unset($url);
13 }
14
15 public function tampil_semua_data()
16 { // memanggil method/fungsi yang ada di server dan dimasukkan ke variable $data
17   $data = $this->api->call('tampil_semua_data');
18   // mengembalikan data
19   return $data;
20   // menghapus variable dari memory
21   unset($data);
22 }
23
24 public function tampil_data($id_barang)
25 { $data = $this->api->call('tampil_data',array($id_barang));
26   return $data;
27   unset($id_barang,$data);
28 }
29
30 public function tambah_data($data)
31 { $this->api->call('tambah_data',array($data));
32   unset($data);
33 }
34
35 public function ubah_data($data)
36 { $this->api->call('ubah_data',array($data));
37   unset($data);
38 }
39
40 public function hapus_data($id_barang)
41 { $this->api->call('hapus_data',array($id_barang));
42   unset($id_barang);
43 }
44
45 // function yang terakhir kali di-load saat class dipanggil
46 public function __destruct()
47 { // menghapus variable $api dari memory
48   unset($this->api);
49 }
50 }
51
52 $url = "http://192.168.56.2/wsdl-toko/server/server.php?wsdl";
53
54 // buat objek baru dari class Client
55 $abc = new Client($url);
56 ?>
```

Gambar 6. Source code Client.php di Windows

5.2.4 *Source code proses.php di Windows*



```
1 <?php
2 include "Client.php";
3
4 if ($_POST['aksi']=='tambah')
5 {   $data = array("id_barang"=>$_POST['id_barang'],
6                 "nama_barang"=>$_POST['nama_barang']);
7     $abc->tambah_data($data);
8     header('location:index.php?page=daftar-data');
9 } else if ($_POST['aksi']=='ubah')
10 {   $data = array("id_barang"=>$_POST['id_barang'],
11                 "nama_barang"=>$_POST['nama_barang']);
12     $abc->ubah_data($data);
13     header('location:index.php?page=daftar-data');
14 } else if ($_GET['aksi']=='hapus')
15 {   $abc->hapus_data($_GET['id_barang']);
16     header('location:index.php?page=daftar-data');
17 }
18 unset($abc,$data);
19 ?>
```

Gambar 7. *Source code proses.php di Windows*

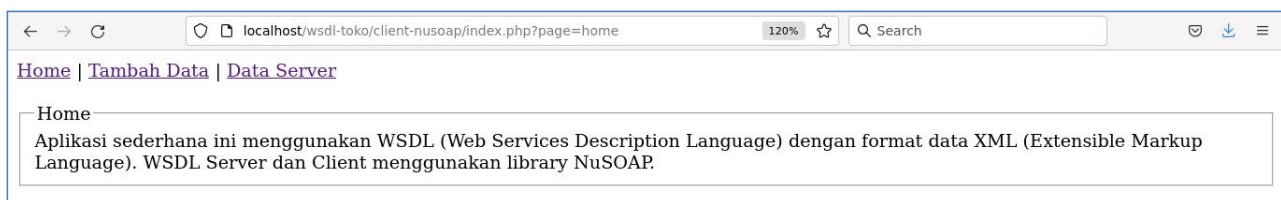
PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

5.2.5 Source code *index.php* di *Windows*

```
1 <?php
2 include "Client.php";
3 ?>
4 <!doctype html>
5 <html>
6 <head>
7 <title></title>
8 </head>
9 <body>
10 <a href="?page=home">Home</a> | <a href="?page=tambah">Tambah Data</a> | <a href="?page=daftar-data">Data Server</a>
11 <br/><br/>
12
13 <fieldset>
14 <? if ($ GET['page']=='tambah') { ?>
15 <legend>Tambah Data</legend>
16 <form name="form" method="POST" action="proses.php">
17 <input type="hidden" name="aksi" value="tambah"/>
18 <label>ID Barang</label>
19 <input type="text" name="id_barang"/>
20 <br/>
21 <label>Nama Barang</label>
22 <input type="text" name="nama_barang"/>
23 <br/>
24 <button type="submit" name="simpan">Simpan</button>
25 </form>
26
27 <? } elseif ($ GET['page']=='ubah') {
28 $r = $abc->tampil_data($ GET['id_barang']);
29 ?>
30 <legend>Ubah Data</legend>
31 <form name="form" method="post" action="proses.php">
32 <input type="hidden" name="aksi" value="ubah"/>
33 <input type="hidden" name="id_barang" value="<?=$r['id_barang']?>" />
34 <label>ID Barang</label>
35 <input type="text" value="<?=$r['id_barang']?>" disabled>
36 <br/>
37 <label>Nama Barang</label>
38 <input type="text" name="nama_barang" value="<?=$r['nama_barang']?>">
39 <br/>
40 <button type="submit" name="ubah">Ubah</button>
41 </form>
42
43 <? unset($r);
44 } else if ($ GET['page']=='daftar-data') {
45 ?>
46 <legend>Daftar Data Server</legend>
47 <table border="1">
48 <tr><th width="5%">No</th>
49 <th width="18%">ID Barang</th>
50 <th width="75%">Nama</th>
51 <th width="5%" colspan="2">Aksi</th>
52 </tr>
53 <? $no = 1;
54 $data array = $abc->tampil_semua_data();
55 foreach ($data array as $r) {
56 <tr><td><?=$no?></td>
57 <td><?=$r['id_barang']?></td>
58 <td><?=$r['nama_barang']?></td>
59 <td><a href="?page=ubah&id_barang=<?=$r['id_barang']?>">Ubah</a></td>
60 <td><a href="proses.php?aksi=hapus&id_barang=<?=$r['id_barang']?>" onclick="return confirm('Apakah Anda ingin menghapus data ini?')">Hapus</a></td>
61 </tr>
62 <? $no++;
63 }
64 unset($data_array,$r,$no);
65 ?>
66 </table>
67
68 <? } else { ?>
69 <legend>Home</legend>
70 Aplikasi sederhana ini menggunakan WSDL (Web Services Description Language) dengan format data XML (Extensible Markup Language). WSDL Server dan Client menggunakan library NuSOAP.
71 </fieldset>
72 <? } ?>
73 </body>
74 </html>
```

Gambar 8. Source code *index.php* di *Windows*

Akses melalui *web browser* <http://localhost/wSDL-toko/client-nusoap/index.php> di *Windows*. Coba jalankan aplikasi sederhana ini dengan menambah, menampilkan, mengubah dan menghapus datanya. Data akan dikirim dari *Windows* ke *Debian Server* dan data hanya disimpan di *database Debian Server*. Data yang tersimpan tersebut akan ditampilkan ke *Client* di *Windows*.



Gambar 9. Laman *Client*

MODUL 6

Web Services Description Language (WSDL) - II

6.1 Aplikasi *WSDL - CRUD*

Aplikasi Toko menggunakan *WSDL - CRUD* sederhana bagian kedua, dimana **data akan dikirim dari Client (Windows dengan IP Address 192.168.56.1) ke Server (Debian Server dengan IP Address 192.168.56.xx) dan data hanya disimpan pada database Debian Server.** Database tetap menggunakan database “toko” dan tabel “barang” pada *Debian Server*. Langkah selanjutnya adalah sebagai berikut:

- *WSDL Server* pada *Debian Server* masih tetap menggunakan *NuSOAP library* seperti pada modul sebelumnya.
- *WSDL Client* pada *Windows* menggunakan *SOAP Client* tanpa *NuSOAP library*. Pada modul sebelumnya, *WSDL Client* menggunakan *NuSOAP library*. Buat folder *xampp\htdocs\wsdl-toko\client-soap* lalu buat file *Client.php*, *proses.php* dan *index.php*.
- Akses *phpinfo* pada *Server (Debian Server)* serta pastikan *PHP extension xml, soap, PDO* dan *pdo_mysql* sudah aktif.
- Akses *phpinfo* pada *Client (Windows)* serta pastikan *PHP extension xml* dan *soap* sudah aktif.

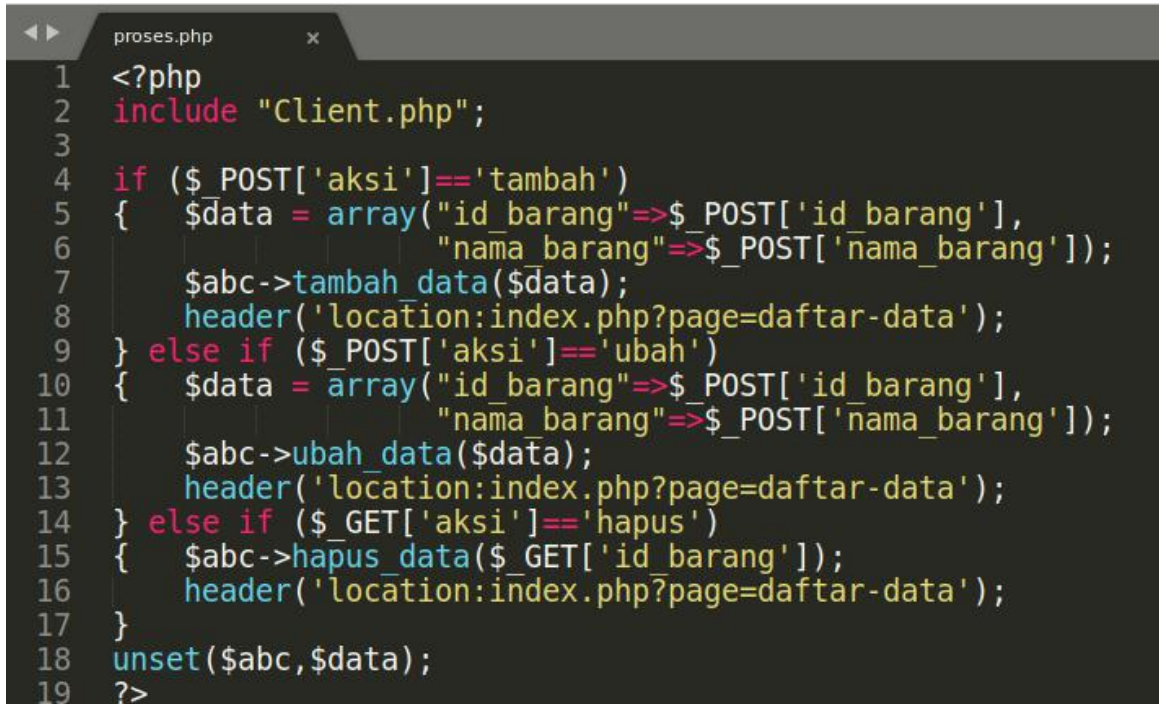
**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

6.1.1 *Source code Client.php di Windows*

```
Client.php x
1 <?php
2 error_reporting(1); // error ditampilkan
3 class Client
4 { private $api;
5
6 // function yang pertama kali di-load saat class dipanggil
7 public function __construct($url)
8 { // buat objek baru dari class SOAP Client
9   $this->api = new SoapClient($url);
10  unset($url);
11 }
12
13 public function tampil_semua_data()
14 { // memanggil method/fungsi yang ada di server dan dimasukkan ke variable $data
15   $data = $this->api->tampil_semua_data();
16   // mengembalikan data
17   return $data;
18   // menghapus variable dari memory
19   unset($data);
20 }
21
22 public function tampil_data($id_barang)
23 { $data = $this->api->tampil_data($id_barang);
24   return $data;
25   unset($id_barang,$data);
26 }
27
28 public function tambah_data($data)
29 { $this->api->tambah_data($data);
30   unset($data);
31 }
32
33 public function ubah_data($data)
34 { $this->api->ubah_data($data);
35   unset($data);
36 }
37
38 public function hapus_data($id_barang)
39 { $this->api->hapus_data($id_barang);
40   unset($id_barang);
41 }
42
43 // function yang terakhir kali di-load saat class dipanggil
44 public function __destruct()
45 { // menghapus variable $api dari memory
46   unset($this->api);
47 }
48 }
49
50 $url = "http://192.168.56.2/wsdl-toko/server/server.php?wsdl";
51
52 // buat objek baru dari class Client
53 $abc = new Client($url);
54 ?>
```

Gambar 1. *Source code Client.php di Windows*

6.1.2 *Source code proses.php di Windows*



```
1 <?php
2 include "Client.php";
3
4 if ($_POST['aksi']=='tambah')
5 {   $data = array("id_barang"=>$_POST['id_barang'],
6                 "nama_barang"=>$_POST['nama_barang']);
7     $abc->tambah_data($data);
8     header('location:index.php?page=daftar-data');
9 } else if ($_POST['aksi']=='ubah')
10 {   $data = array("id_barang"=>$_POST['id_barang'],
11                 "nama_barang"=>$_POST['nama_barang']);
12     $abc->ubah_data($data);
13     header('location:index.php?page=daftar-data');
14 } else if ($_GET['aksi']=='hapus')
15 {   $abc->hapus_data($_GET['id_barang']);
16     header('location:index.php?page=daftar-data');
17 }
18 unset($abc,$data);
19 ?>
```

Gambar 2. *Source code proses.php di Windows*

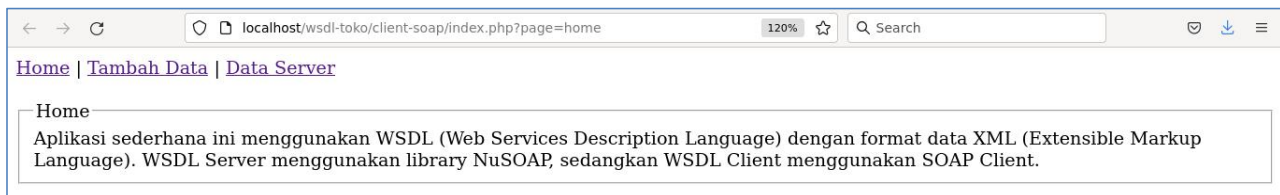
PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

6.1.3 Source code *index.php* di *Windows*

```
1 <?php
2 include "Client.php";
3 ?>
4 <!doctype html>
5 <html>
6 <head>
7     <title></title>
8 </head>
9 <body>
10 <a href="?page=home">Home</a> | <a href="?page=tambah">Tambah Data</a> | <a href="?page=daftar-data">Data Server</a>
11 <br/><br/>
12
13 <fieldset>
14 <? if ($ GET['page']=='tambah') { ?>
15 <legend>Tambah Data</legend>
16 <form name="form" metode="POST" action="proses.php">
17     <input type="hidden" name="aksi" value="tambah"/>
18     <label>ID Barang</label>
19     <input type="text" name="id_barang"/>
20     <br/>
21     <label>Nama Barang</label>
22     <input type="text" name="nama_barang"/>
23     <br/>
24     <button type="submit" name="simpan">Simpan</button>
25 </form>
26
27 <? } elseif ($ GET['page']=='ubah') {
28     $r = $abc->tampil_data($ GET['id_barang']);
29 ?>
30 <legend>Ubah Data</legend>
31 <form name="form" method="post" action="proses.php">
32     <input type="hidden" name="aksi" value="ubah"/>
33     <input type="hidden" name="id_barang" value="<?=$r->id_barang?" />
34     <label>ID Barang</label>
35     <input type="text" value="<?=$r->id_barang?" disabled>
36     <br/>
37     <label>Nama Barang</label>
38     <input type="text" name="nama_barang" value="<?=$r->nama_barang?">
39     <br/>
40     <button type="submit" name="ubah">Ubah</button>
41 </form>
42
43 <? unset($r);
44     } else if ($ GET['page']=='daftar-data') {
45 ?>
46 <legend>Daftar Data Server</legend>
47 <table border="1">
48     <tr><th width="5%">No</th>
49     <th width="10%">ID Barang</th>
50     <th width="75%">Nama</th>
51     <th width="5%" colspan="2">Aksi</th>
52 </tr>
53 <? $no = 1;
54     $data array = $abc->tampil_semua_data();
55     foreach ($data array as $r) {
56 ?> <tr><td=<?=$no?></td>
57     <td=<?=$r->id_barang?></td>
58     <td=<?=$r->nama_barang?></td>
59     <td><a href="?page=ubah&id_barang=<?=$r->id_barang?>Ubah">Ubah</a></td>
60     <td><a href="proses.php?aksi=hapus&id_barang=<?=$r->id_barang?" onclick="return confirm('Apakah Anda ingin menghapus data ini?')">Hapus</a></td>
61 </td>
62 <? $no++;
63     }
64     unset($data_array,$r,$no);
65 ?>
66 </table>
67
68 <? } else { ?>
69 <legend>Home</legend>
70     Aplikasi sederhana ini menggunakan WSDL (Web Services Description Language) dengan format data XML (Extensible Markup Language). WSDL Server menggunakan library NuSOAP, sedangkan WSDL Client menggunakan SOAP Client.
71 </fieldset>
72 <? } ?>
73 </body>
74 </html>
```

Gambar 3. Source code *index.php* di *Windows*

Akses melalui *web browser* ***http://localhost/wsdL-toko/client-soap/index.php*** di *Windows*. Coba jalankan aplikasi sederhana ini dengan menambah, menampilkan, mengubah dan menghapus datanya. Data akan dikirim dari *Windows* ke *Debian Server* dan data hanya disimpan di *database Debian Server*. Data yang tersimpan tersebut akan ditampilkan ke *Client* di *Windows*.



Gambar 4. Laman *Client*

MODUL 7

REpresentational State Transfer (REST)

7.1 Pengertian

REST adalah singkatan dari *REpresentational State Transfer*. *REST* merupakan gaya arsitektur untuk menyediakan standar antara sistem komputer di *web*, sehingga memudahkan sistem untuk berkomunikasi satu sama lain.

Menurut *Code Academy*, sistem yang sesuai dengan *REST*, sering disebut sistem *RESTful*, dicirikan oleh sifatnya yang *stateless* dan dapat memisahkan masalah *client* dan *server*.

Stateless artinya *server* tidak perlu mengetahui apa pun tentang status *client* dan sebaliknya. Sehingga, baik *server* maupun *client* dapat memahami pesan apapun yang diterima, bahkan tanpa melihat pesan sebelumnya.

Dalam gaya arsitektur *REST*, implementasi *client* dan implementasi *server* dapat dilakukan secara independen tanpa saling mengetahui satu sama lain. Ini berarti kode di sisi *client* dapat diubah kapan saja tanpa memengaruhi pengoperasian *server*, dan kode di sisi *server* dapat diubah tanpa memengaruhi operasi *client*.

REST menggunakan serangkaian metode permintaan standar termasuk *GET*, *POST*, *PUT*, *DELETE*, dan kemampuan *HTTP* lain yang ada.

Pada intinya, *REST* terdiri dari empat hal berikut

- Sumber daya yang direferensikan oleh pengenalan global seperti *URI*.
- Representasi sumber daya, yaitu dokumen yang berisi informasi sumber daya yang dibutuhkan.
- Komponen yang berkomunikasi satu sama lain antara *client* dan *server*.
- *Interface* standar yang mengidentifikasi sumber daya, memungkinkan manipulasi sumber daya melalui representasi, termasuk pesan deskriptif mandiri yang menjelaskan cara memproses permintaan dan respons, dan hypermedia yang diidentifikasi secara dinamis.

Menurut *REST API*, terdapat enam prinsip dasar dari *REST*.

1. *Client-server*

Prinsip dasar *REST* yang pertama adalah *client-server*, yaitu dengan memisahkan masalah user *interface* dari masalah penyimpanan data. Dengan memisahkan masalah ini, kita dapat meningkatkan portabilitas antarmuka pengguna di berbagai platform dan meningkatkan skalabilitas dengan menyederhanakan komponen *server*.

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

2. *Stateless*

Prinsip dasar *REST* yang kedua adalah *stateless*. Prinsip ini memudahkan akses informasi dari *client* ke *server* maupun sebaliknya. Setiap permintaan dari *client* ke *server* harus berisi semua informasi yang diperlukan untuk memahami permintaan tersebut, dan tidak dapat memanfaatkan konteks yang disimpan di *server*. Maka dari itu, status sesi disimpan sepenuhnya pada *client*.

3. *Cacheable*

Batasan *cache* mengharuskan data dalam respons terhadap permintaan diberi label secara implisit atau eksplisit sebagai dapat disimpan dalam *cache* atau tidak dapat disimpan dalam *cache*. Jika respons tersebut dapat disimpan dalam *cache*, *cache client* diberikan hak untuk menggunakan kembali data respons tersebut untuk permintaan yang setara nanti.

4. *Interface* yang seragam

Interface yang seragam dibuat dengan menerapkan prinsip umum rekayasa perangkat lunak ke antarmuka komponen, arsitektur sistem secara keseluruhan disederhanakan dan visibilitas interaksi ditingkatkan. Untuk mendapatkan *interface* yang seragam, beberapa batasan arsitektur diperlukan untuk memandu perilaku komponen.

REST didefinisikan oleh empat batasan *interface*.

- Identifikasi sumber daya
- Manipulasi sumber daya melalui representasi
- Pesan deskriptif diri
- Hypermedia sebagai mesin status aplikasi

5. Sistem yang berlapis

Prinsip dasar *REST* selanjutnya adalah sistem yang berlapis. Gaya sistem berlapis memungkinkan arsitektur untuk terdiri dari lapisan hierarki dengan membatasi perilaku komponen sedemikian rupa. Sehingga, setiap komponen tidak dapat “melihat” di luar lapisan langsung yang berinteraksi dengan mereka.

6. *Code on demand* yang opsional

REST memungkinkan fungsionalitas *client* diperluas dengan mengunduh dan menjalankan kode dalam bentuk *applet* atau *script*. Ini menyederhanakan *client* dengan mengurangi jumlah fitur yang diperlukan untuk diterapkan sebelumnya.

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

7.2 Aplikasi

Aplikasi Toko menggunakan *RESTful* format *XML - CRUD* sederhana, dimana **data akan dikirim dari Client (Windows dengan IP Address 192.168.56.1) ke Server (Debian Server dengan IP Address 192.168.56.xx) dan data hanya disimpan pada database Debian Server.** Database tetap menggunakan database “toko” dan tabel “barang” pada *Debian Server*. Langkah selanjutnya adalah sebagai berikut:

1. Buat folder */opt/lampp/htdocs/restful-xml-toko/server/* pada *Debian Server* lalu buat file ***Database.php*** dan ***server.php***.
2. Buat juga folder *xampp\htdocs\restful-xml-toko\client* pada *Windows* lalu buat file ***Client.php***, ***proses.php*** dan ***index.php***.
3. Akses *phpinfo* pada *Server (Debian Server)* serta pastikan *PHP extension xml, PDO* dan ***pdo_mysql*** sudah aktif.
4. Akses *phpinfo* pada *Client (Windows)* serta pastikan *PHP extension xml* sudah aktif.

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

7.2.1 Source code Database.php di Debian Server

```
Database.php x
1 <?php
2 class Database
3 {
4     private $host="localhost";
5     private $dbname="toko";
6     private $user="root";
7     private $password="root";
8     private $port="3306";
9     private $conn;
10
11 // function yang pertama kali di-load saat class dipanggil
12 public function __construct()
13 {
14     // koneksi database
15     try
16     {
17         $this->conn = new PDO("mysql:host=$this->host;port=$this->port;dbname=$this->dbname;charset=utf8",$this->user,$this->password);
18     }
19     catch (PDOException $e)
20     {
21         echo "Koneksi gagal";
22     }
23 }
24
25 public function tampil_semua_data()
26 {
27     $query = $this->conn->prepare("select id_barang, nama_barang from barang order by id_barang");
28     $query->execute();
29     // mengambil banyak data dengan fetchAll
30     $data = $query->fetchAll(PDO::FETCH_ASSOC);
31     // mengembalikan data
32     return $data;
33     // hapus variable dari memory
34     $query->closeCursor();
35     unset($data);
36 }
37
38 public function tampil_data($id_barang)
39 {
40     $query = $this->conn->prepare("select id_barang,nama_barang from barang where id_barang=?");
41     $query->execute(array($id_barang));
42     // mengambil satu data dengan fetch
43     $data = $query->fetch(PDO::FETCH_ASSOC);
44     return $data;
45     $query->closeCursor();
46     unset($id_barang,$data);
47 }
48
49 public function tambah_data($data)
50 {
51     $query = $this->conn->prepare("insert ignore into barang (id_barang,nama_barang) values (?,?)");
52     $query->execute(array($data['id_barang'],$data['nama_barang']));
53     $query->closeCursor();
54     unset($data);
55 }
56
57 public function ubah_data($data)
58 {
59     $query = $this->conn->prepare("update barang set nama_barang=? where id_barang=?");
60     $query->execute(array($data['nama_barang'],$data['id_barang']));
61     $query->closeCursor();
62     unset($data);
63 }
64
65 public function hapus_data($id_barang)
66 {
67     $query = $this->conn->prepare("delete from barang where id_barang=?");
68     $query->execute(array($id_barang));
69     $query->closeCursor();
70     unset($id_barang);
71 }
72 }
73 ?>
```

Gambar 1. Source code Database.php di Debian Server

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

7.2.2 Source code *server.php* di Debian Server

```
1 <?php
2 error_reporting(1); // error ditampilkan
3 header('Content-Type: text/xml; charset=UTF-8');
4 include "Database.php";
5 // buat objek baru dari class Database
6 $abc = new Database();
7
8 // function untuk menghapus selain huruf dan angka
9 function filter($data)
10 { $data = preg_replace('/[^a-zA-Z0-9]/', '', $data);
11   return $data;
12   unset($data);
13 }
14
15 if ($SERVER['REQUEST_METHOD'] == 'POST')
16 { $input = file_get_contents("php://input");
17   $data = simplexml_load_string($input);
18   $aksi = $data->barang->aksi;
19   $id_barang = $data->barang->id_barang;
20   $nama_barang = $data->barang->nama_barang;
21
22   if ($aksi == 'tambah')
23   { $data2=array('id_barang' => $id_barang,
24                'nama_barang' => $nama_barang
25                );
26     $abc->tambah_data($data2);
27   } elseif ($aksi == 'ubah')
28   { $data2=array('id_barang' => $id_barang,
29                'nama_barang' => $nama_barang
30                );
31     $abc->ubah_data($data2);
32   } elseif ($aksi == 'hapus')
33   { $abc->hapus_data($id_barang);
34   }
35   // hapus variable dari memory
36   unset($input,$data,$data2,$id_barang,$nama_barang,$aksi,$abc);
37 } elseif ($SERVER['REQUEST_METHOD'] == 'GET')
38 { if ( ($GET['aksi']=='tampil') and (isset($GET['id_barang'])) )
39   { $id_barang = filter($GET['id_barang']);
40     $data=$abc->tampil_data($id_barang);
41     $xml = "<toko>";
42     $xml .= "<barang>";
43     $xml .= "<id_barang>".$data['id_barang']. "</id_barang>";
44     $xml .= "<nama_barang>".$data['nama_barang']. "</nama_barang>";
45     $xml .= "</barang>";
46     $xml .= "</toko>";
47     echo $xml;
48   } else //menampilkan semua data
49   { $data = $abc->tampil_semua_data();
50     $xml = "<toko>";
51     foreach($data as $a)
52     { $xml .= "<barang>";
53       foreach($a as $kolom => $value)
54       { $xml .= "<$kolom>$value</$kolom>";
55         // atau menggunakan
56         // $xml .= "<$kolom>![CDATA[$value]]</$kolom>";
57       }
58       $xml .= "</barang>";
59     }
60     $xml .= "</toko>";
61     echo $xml;
62   }
63   unset($id_barang,$data,$xml);
64 }
65 ?>
```

Gambar 2. Source code *server.php* di Debian Server

Akses *RESTful* menggunakan web browser <http://192.168.56.xx/restful-xml-toko/server/server.php>

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
-< toko >
  -< barang >
    < id_barang > 111 < /id_barang >
    < nama_barang > Pensil < /nama_barang >
  < /barang >
  -< barang >
    < id_barang > 222 < /id_barang >
    < nama_barang > Penghapus < /nama_barang >
  < /barang >
  -< barang >
    < id_barang > 333 < /id_barang >
    < nama_barang > Buku Tulis < /nama_barang >
  < /barang >
< / toko >
```

Gambar 3. RESTful format xml di Debian Server

7.2.3 Source code Client.php di Windows

```
1 <?php
2 error_reporting(1); // error ditampilkan
3 class Client
4 { private $url;
5
6 // function yang pertama kali di-load saat class dipanggil
7 public function __construct($url)
8 { $this->url=$url;
9   unset($url);
10 }
11
12 // function untuk menghapus selain huruf dan angka
13 public function filter($data)
14 { $data = preg_replace('/[^\a-zA-Z0-9]/', '', $data);
15   return $data;
16   unset($data);
17 }
18
19 public function tampil_semua_data()
20 { $client = curl_init($this->url);
21   curl_setopt($client, CURLOPT_RETURNTRANSFER, 1);
22   $response = curl_exec($client);
23   curl_close($client);
24   $data = simplexml_load_string($response);
25   // mengembalikan data
26   return $data;
27   // menghapus variabel dari memory
28   unset($data, $client, $response);
29 }
30
31 public function tampil_data($id_barang)
32 { $id_barang = $this->filter($id_barang);
33   $client = curl_init($this->url."?aksi=tampil&id_barang=".$id_barang);
34   curl_setopt($client, CURLOPT_RETURNTRANSFER, 1);
35   $response = curl_exec($client);
36   curl_close($client);
37   $data = simplexml_load_string($response);
38   return $data;
39   unset($id_barang, $client, $response, $data);
40 }
41
42 public function tambah_data($data)
43 { $data="<toko>
44   <barang>
45     <id_barang>" . $data['id_barang'] . "</id_barang>
46     <nama_barang>" . $data['nama_barang'] . "</nama_barang>
47     <aksi>" . $data['aksi'] . "</aksi>
48   </barang>
49   </toko>";
50   $c = curl_init();
51   curl_setopt($c, CURLOPT_URL, $this->url);
52   curl_setopt($c, CURLOPT_RETURNTRANSFER, true);
53   curl_setopt($c, CURLOPT_POST, true);
54   curl_setopt($c, CURLOPT_POSTFIELDS, $data);
55   $response = curl_exec($c);
56   curl_close($c);
57   unset($data, $c, $response);
58 }
59
60 public function ubah_data($data)
61 { $data="<toko>
62   <barang>
63     <id_barang>" . $data['id_barang'] . "</id_barang>
64     <nama_barang>" . $data['nama_barang'] . "</nama_barang>
65     <aksi>" . $data['aksi'] . "</aksi>
66   </barang>
67   </toko>";
68   $c = curl_init();
69   curl_setopt($c, CURLOPT_URL, $this->url);
70   curl_setopt($c, CURLOPT_RETURNTRANSFER, true);
71   curl_setopt($c, CURLOPT_POST, true);
72   curl_setopt($c, CURLOPT_POSTFIELDS, $data);
73   $response = curl_exec($c);
74   curl_close($c);
```

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

```
75     unset($data,$c,$response);
76 }
77
78 public function hapus_data($id_barang)
79 { $id_barang = $this->filter($id_barang);
80   $data = "<toko>
81     <barang>
82       <id_barang>".$id_barang."</id_barang>
83       <aksi>hapus</aksi>
84     </barang>
85   </toko>";
86   $c = curl_init();
87   curl_setopt($c,CURLOPT_URL,$this->url);
88   curl_setopt($c,CURLOPT_RETURNTRANSFER,true);
89   curl_setopt($c,CURLOPT_POST,true);
90   curl_setopt($c,CURLOPT_POSTFIELDS,$data);
91   $response = curl_exec($c);
92   curl_close($c);
93   unset($id_barang,$data,$c,$response);
94 }
95
96 // function yang terakhir kali di-load saat class dipanggil
97 public function __destruct()
98 { // hapus variable dari memory
99   unset($this->options,$this->api);
100 }
101 }
102
103 $url = 'http://192.168.56.2/restful-xml-toko/server/server.php';
104 // buat objek baru dari class Client
105 $abc = new Client($url);
106 ?>
```

Gambar 4. Source code Client.php di Windows

7.2.4 Source code proses.php di Windows

```
1 <?php
2 include "Client.php";
3
4 if ($ _POST['aksi']=='tambah')
5 { $data = array("id_barang"=>$ _POST['id_barang'],
6               "nama_barang"=>$ _POST['nama_barang'],
7               "aksi"=>$ _POST['aksi']);
8   $abc->tambah_data($data);
9   header('location:index.php?page=daftar-data');
10 } else if ($ _POST['aksi']=='ubah')
11 { $data = array("id_barang"=>$ _POST['id_barang'],
12               "nama_barang"=>$ _POST['nama_barang'],
13               "aksi"=>$ _POST['aksi']);
14   $abc->ubah_data($data);
15   header('location:index.php?page=daftar-data');
16 } else if ($ _GET['aksi']=='hapus')
17 { $abc->hapus_data($ _GET['id_barang']);
18   header('location:index.php?page=daftar-data');
19 }
20 unset($abc,$data);
21 ?>
```

Gambar 5. Source code proses.php di Windows

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

7.2.5 Source code *index.php* di *Windows*

```
1 <?php
2 include "Client.php";
3 ?>
4 <!doctype html>
5 <html>
6 <head>
7     <title></title>
8 </head>
9 <body>
10 <a href="?page=home">Home</a> | <a href="?page=tambah">Tambah Data</a> | <a href="?page=daftar-data">Data Server</a>
11 <br/><br/>
12
13 <fieldset>
14 <? if ($ GET['page']=='tambah') { ?>
15 <legend>Tambah Data</legend>
16 <form name="form" method="POST" action="proses.php">
17     <input type="hidden" name="aksi" value="tambah"/>
18     <label>ID Barang</label>
19     <input type="text" name="id_barang"/>
20     <br/>
21     <label>Nama Barang</label>
22     <input type="text" name="nama_barang"/>
23     <br/>
24     <button type="submit" name="simpan">Simpan</button>
25 </form>
26
27 <? } elseif ($ GET['page']=='ubah') {
28     $r = $abc->tampil_data($ GET['id_barang']);
29 ?>
30 <legend>Ubah Data</legend>
31 <form name="form" method="post" action="proses.php">
32     <input type="hidden" name="aksi" value="ubah"/>
33     <input type="hidden" name="id_barang" value="<?=$r->barang->id_barang?"/>
34     <label>ID Barang</label>
35     <input type="text" value="<?=$r->barang->id_barang?" disabled>
36     <br/>
37     <label>Nama Barang</label>
38     <input type="text" name="nama_barang" value="<?=$r->barang->nama_barang?">
39     <br/>
40     <button type="submit" name="ubah">Ubah</button>
41 </form>
42
43 <? unset($r);
44 } else if ($ GET['page']=='daftar-data') {
45 ?>
46 <legend>Daftar Data Server</legend>
47 <table border="1">
48     <tr><th width="5%">No</th>
49     <th width="10%">ID Barang</th>
50     <th width="75%">Nama</th>
51     <th width="5%" colspan="2">Aksi</th>
52 </tr>
53 <? $no = 1;
54 $data array = $abc->tampil_semua_data();
55 foreach ($data array as $r) {
56 ?> <tr><td><?=$no?></td>
57     <td><?=$r->id_barang?></td>
58     <td><?=$r->nama_barang?></td>
59     <td><a href="?page=ubah&id_barang=<?=$r->id_barang?">Ubah</a></td>
60     <td><a href="proses.php?aksi=hapus&id_barang=<?=$r->id_barang?" onclick="return confirm('Apakah Anda ingin menghapus data ini?')">Hapus</a></td>
61 </tr>
62 <? $no++;
63 }
64 unset($data array,$r,$no);
65 ?>
66 </table>
67
68 <? } else { ?>
69 <legend>Home</legend>
70     Aplikasi sederhana ini menggunakan RESTful dengan format data XML (Extensible Markup Language).
71 </fieldset>
72 <? } ?>
73 </body>
74 </html>
```

Gambar 6. Source code *index.php* di *Windows*

Akses melalui *web browser http://localhost/restful-xml-toko/client/index.php* di *Windows*. Coba jalankan aplikasi sederhana ini dengan menambah, menampilkan, mengubah dan menghapus datanya. Data akan dikirim dari *Windows* ke *Debian Server* dan data hanya disimpan di *database Debian Server*. Data yang tersimpan tersebut akan ditampilkan ke *Client* di *Windows*.



Gambar 7. Laman *Client*

MODUL 8

REpresentational State Transfer (REST) - II

8.1 Pengertian

8.1.1 *JSON*

JavaScript object notation (JSON) adalah format yang digunakan untuk menyimpan dan mentransfer data. Berbeda dengan XML (*extensive markup language*) dan format lainnya yang memiliki fungsi serupa, JSON memiliki struktur data yang sederhana dan mudah dipahami. Itulah mengapa JSON sering digunakan pada API.

JSON sendiri terdiri dari dua struktur, yaitu:

- Kumpulan *value* yang saling berpasangan. Dalam JSON, contohnya adalah object.
- Daftar *value* yang berurutan, seperti array.

JSON juga tidak hanya dapat digunakan dengan JavaScript. Berikut ini adalah beberapa bahasa pemrograman lain yang mendukung penggunaan format tersebut: PHP, Python, Ruby, C++, Perl.

Fungsi JSON

Fungsi JSON kurang lebih sama dengan XML. Sebelum JSON diciptakan, format bahasa markup ini sering digunakan dalam AJAX JavaScript (*asynchronous JavaScript and XML*) untuk penyimpanan dan perpindahan data.

Berkat gabungan keduanya, developer dapat membuat halaman website di mana informasinya dapat diperbarui tanpa harus memuat ulang halaman tersebut. Ini sudah menjadi hal yang umum, terutama dengan menjamurnya tren aplikasi web.

JSON memiliki fungsi yang sama. Bedanya, ia memiliki beberapa keunggulan yang membuatnya lebih sering digunakan daripada XML, termasuk:

- File yang lebih ringan
- Struktur kode yang lebih sederhana
- Kode yang mudah dipahami oleh manusia

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

Perbandingan JSON dan XML

Seberapa sederhanakah kode JSON jika dibandingkan dengan kode XML? Untuk mengetahuinya, Anda dapat memperhatikan kedua contoh di bawah ini. Contoh pertama adalah kode XML.

```
<users>
  <user>
    <username>Anton</username> <lokasi>Bandung</lokasi>
  </user>
  <user>
    <username>Budi</username> <lokasi>Semarang</lokasi>
  </user>
  <user>
    <username>Nana</username> <lokasi>Surabaya</lokasi>
  </user>
  <user>
    <username>Jamal</username> <lokasi>Tangerang</lokasi>
  </user>
</users>
```

Contoh kode JSON JavaScript adalah sebagai berikut:

```
{"users": [
  {"username": "Anton", "lokasi": "Bandung"},
  {"username": "Budi", "lokasi": "Semarang"},
  {"username": "Nana", "lokasi": "Surabaya"},
  {"username": "Jamal", "lokasi": "Tangerang"}
]}
```

Dapat Anda lihat bahwa kode JSON lebih ringkas karena tidak memerlukan tag pembuka dan penutup. Dengan demikian, kodenya juga lebih mudah dipahami. Di samping itu, kelebihan ini juga memberikan dua manfaat lainnya, yaitu:

- Proses loading data yang lebih ringan karena ukuran file yang kecil.
- Penulisan kode yang lebih cepat dengan kode yang sederhana.

Cara Menggunakan JSON

Ada dua hal yang perlu Anda pelajari agar dapat menggunakan JSON, yaitu syntax (cara penulisan) dan jenis *value*-nya. Untuk memahami keduanya, simak penjelasan di bawah ini.

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

Syntax JSON

Pada contoh di bagian sebelumnya memang menunjukkan bahwa JSON memiliki struktur kode yang sederhana. Namun, mari lihat contoh yang lebih simpel lagi agar Anda tahu apa saja yang ada dalam syntax-nya:

```
{"city": "New York", "country": "United States" }
```

JSON selalu dibuka dan ditutup dengan tanda `{}` atau kurung kurawal. Syntax-nya terdiri dari dua elemen, yaitu **key** dan **value**. Keduanya dipisahkan oleh titik dua agar jelas. Contoh **key** adalah “city”, sedangkan “New York” adalah contoh **value**. Keduanya selalu dibuka dan ditutup dengan tanda kutip ganda. Perlu Anda ingat bahwa kedua tidak ada spasi di antara kedua elemen tersebut. Apabila ada lebih dari satu pasang key dan value, Anda perlu memisahkannya dengan tanda koma yang diikuti spasi. Ini dapat Anda lihat pada contoh di atas. Meskipun **key** harus berupa *string*, Anda dapat menggunakan beberapa jenis data sebagai **value**-nya.

Jenis-Jenis Value JSON

Setidaknya ada enam jenis data yang dapat digunakan sebagai *value* JSON, yaitu:

- *String*
- *Object*
- *Array*
- *Boolean*
- *Number*
- *Null*

String

String adalah data yang terdiri dari karakter *unicode*, seperti “Anton” pada contoh di bawah ini.

```
“nama” : “Anton”
```

Object

Object adalah sepasang *key* dan *value*. Seperti pada contoh syntax tadi, *object* dibuka dan ditutup dengan kurung kurawal. Jika ada lebih dari satu *object*, masing-masing dipisahkan oleh koma dan spasi.

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

```
“karyawan”: {“nama”:”Anton”, “asal”:”Bandung”}
```

Pada contoh di atas, semua yang ada di dalam kurung kurawal adalah *object* yang merupakan *value* dari “karyawan”.

Array

Array adalah kumpulan *object*. Kumpulan ini dibuka dan ditutup dengan kurung siku [].

```
“karyawan”:[  
  {“nama”:”Anton”, “asal”:”Bandung”},  
  {“nama”:”Nana”, “asal”:”Surabaya”},  
  {“nama”:”Jamal”, “asal”:”Tangerang”}  
]
```

Boolean

Boolean adalah jenis data yang hanya berisikan pernyataan benar atau salah. Anda hanya tinggal memasukkan *true* atau *false*, seperti pada contoh berikut ini:

```
“karyawan tetap”:”false”
```

Number

Seperti namanya, jenis data ini berupa angka saja. Dengan catatan, angkanya harus merupakan *integer* atau angka bulat. Artinya, 21,8 atau $\sqrt{2}$ tidak dapat dijadikan sebagai *value*. Contoh penggunaannya seperti di bawah ini:

```
“usia”:”29”
```

Null

Jika sebuah *key* tidak memiliki *value*, Anda dapat mengetikkan *null*. Contohnya seperti pada kode berikut:

```
“golonganDarah”:”null”
```

Membuat Nested Objects dan Nested Arrays

Anda dapat membuat *nested object* dan *nested array* dalam kode JSON. Untuk memahami penggunaannya, amati kedua contoh yang dibahas pada bagian ini.

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

Pertama, mari perhatikan contoh *nested object* berikut:

```
“karyawan”: {  
    “nama”: “Anton”,  
    “asal”: “Bandung”.  
    “hobi”: {  
        “hobi1”: “berenang”,  
        “hobi2”: “melukis”,  
        “hobi3”: “jogging”  
    }  
}
```

Pada contoh tersebut, “hobi1”, “hobi2”, dan “hobi3” merupakan *value* dari *key* “hobi”. Perhatikan bahwa ketiga *value* tersebut ada di dalam kurung kurawal. Dengan demikian, “hobi” beserta ketiga *value*-nya merupakan *object* di dalam *object*.

Penggunaan *nested array* tidak jauh berbeda. Perhatikan contoh di bawah ini:

```
“karyawan”: {  
    “nama”: “Anton”,  
    “asal”: “Bandung”.  
    “hobi”: [ “berenang”, “melukis”, “jogging” ]  
}
```

Pada contoh di atas, “berenang”, “melukis”, dan “jogging” adalah array yang merupakan *value* dari “hobi”.

8.2 Aplikasi

Aplikasi Toko menggunakan *RESTful* format *JSON- CRUD* sederhana, dimana **data akan dikirim dari Client (Windows dengan IP Address 192.168.56.1) ke Server (Debian Server dengan IP Address 192.168.56.xx) dan data hanya disimpan pada database Debian Server. Database tetap menggunakan database “toko” dan tabel “barang” pada Debian Server.** Langkah selanjutnya adalah sebagai berikut:

- a) Buat folder `/opt/lampp/htdocs/restful-json-toko/server/` pada Debian Server lalu buat file ***Database.php*** dan ***server.php***.
- b) Buat juga folder `xampp\htdocs\restful-json-toko\client\` pada Windows lalu buat file ***Client.php***, ***proses.php*** dan ***index.php***.
- c) Akses `phpinfo` pada Server (Debian Server) serta pastikan *PHP extension PDO* dan *pdo_mysql* sudah aktif.

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

8.2.1 *Source code Database.php di Debian Server*

```
Database.php x
1 <?php
2 class Database
3 {
4     private $host="localhost";
5     private $dbname="toko";
6     private $user="root";
7     private $password="root";
8     private $port="3306";
9     private $conn;
10
11 // function yang pertama kali di-load saat class dipanggil
12 public function __construct()
13 {
14     // koneksi database
15     try
16     {
17         $this->conn = new PDO("mysql:host=$this->host;port=$this->port;dbname=$this->dbname;charset=utf8",$this->user,$this->password);
18     }
19     catch (PDOException $e)
20     {
21         echo "Koneksi gagal";
22     }
23 }
24
25 public function tampil_semua_data()
26 {
27     $query = $this->conn->prepare("select id_barang, nama_barang from barang order by id_barang");
28     $query->execute();
29     // mengambil banyak data dengan fetchAll
30     $data = $query->fetchAll(PDO::FETCH_ASSOC);
31     // mengembalikan data
32     return $data;
33     // hapus variable dari memory
34     $query->closeCursor();
35     unset($data);
36 }
37
38 public function tampil_data($id_barang)
39 {
40     $query = $this->conn->prepare("select id_barang,nama_barang from barang where id_barang=?");
41     $query->execute(array($id_barang));
42     // mengambil satu data dengan fetch
43     $data = $query->fetch(PDO::FETCH_ASSOC);
44     return $data;
45     $query->closeCursor();
46     unset($id_barang,$data);
47 }
48
49 public function tambah_data($data)
50 {
51     $query = $this->conn->prepare("insert ignore into barang (id_barang,nama_barang) values (?,?)");
52     $query->execute(array($data['id_barang'],$data['nama_barang']));
53     $query->closeCursor();
54     unset($data);
55 }
56
57 public function ubah_data($data)
58 {
59     $query = $this->conn->prepare("update barang set nama_barang=? where id_barang=?");
60     $query->execute(array($data['nama_barang'],$data['id_barang']));
61     $query->closeCursor();
62     unset($data);
63 }
64
65 public function hapus_data($id_barang)
66 {
67     $query = $this->conn->prepare("delete from barang where id_barang=?");
68     $query->execute(array($id_barang));
69     $query->closeCursor();
70     unset($id_barang);
71 }
72 }
73 ?>
```

Gambar 1. *Source code Database.php di Debian Server*

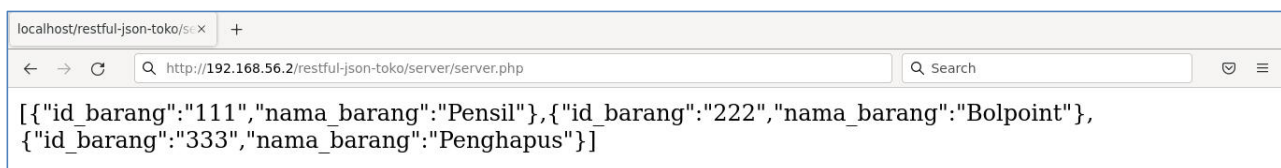
PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

8.2.2 Source code *server.php* di Debian Server

```
1 <?php
2 error_reporting(1);
3 include "Database.php";
4 $abc = new Database();
5
6 if (isset($ SERVER['HTTP_ORIGIN'])) {
7     header("Access-Control-Allow-Origin: {$ SERVER['HTTP_ORIGIN']}");
8     header('Access-Control-Allow-Credentials: true');
9     header('Access-Control-Max-Age: 86400'); // cache for 1 day
10 }
11 if ($ SERVER['REQUEST METHOD'] == 'OPTIONS') {
12     if (isset($ SERVER['HTTP_ACCESS_CONTROL_REQUEST_METHOD']))
13         header("Access-Control-Allow-Methods: GET, POST, OPTIONS");
14     if (isset($ SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']))
15         header("Access-Control-Allow-Headers: {$ SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']}");
16     exit(0);
17 }
18 $postdata = file_get_contents("php://input");
19
20 // function untuk menghapus selain huruf dan angka
21 function filter($data)
22 { $data = preg_replace('/[^a-zA-Z0-9]/', '', $data);
23   return $data;
24   unset($data);
25 }
26
27 if ($ SERVER['REQUEST METHOD'] == 'POST')
28 { $data = json_decode($postdata);
29   $id barang = $data->id barang;
30   $nama barang = $data->nama barang;
31   $aksi = $data->aksi;
32
33   if ($aksi == 'tambah')
34   { $data2=array('id_barang' => $id barang,
35                'nama_barang' => $nama_barang
36                );
37     $abc->tambah_data($data2);
38   } elseif ($aksi == 'ubah')
39   { $data2=array('id_barang' => $id barang,
40                'nama_barang' => $nama_barang
41                );
42     $abc->ubah_data($data2);
43   } elseif ($aksi == 'hapus')
44   { $abc->hapus_data($id_barang);
45   }
46   // hapus variable dari memory
47   unset($postdata,$data,$data2,$id_barang,$nama_barang,$aksi,$abc);
48 } elseif ($ SERVER['REQUEST METHOD'] == 'GET')
49 { if ( ($ GET['aksi']=='tampil') and (isset($ GET['id_barang'])) )
50   { $id_barang = filter($ GET['id_barang']);
51     $data=$abc->tampil_data($id_barang);
52     echo json_encode($data);
53   } else //menampilkan semua data
54   { $data = $abc->tampil_semua_data();
55     echo json_encode($data);
56   }
57   unset($postdata,$data,$id_barang,$abc);
58 }
59 ?>
```

Gambar 2. Source code *server.php* di Debian Server

Akses *RESTful* menggunakan web browser <http://192.168.56.xx/restful-json-toko/server/server.php>



Gambar 3. *RESTful* format json di Debian Server

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

8.2.3 Source code Client.php di Windows

```
1 <?php
2 error_reporting(1); // error ditampilkan
3 class Client
4 { private $url;
5
6 // function yang pertama kali di-load saat class dipanggil
7 public function __construct($url)
8 { $this->url=$url;
9   unset($url);
10 }
11
12 // function untuk menghapus selain huruf dan angka
13 public function filter($data)
14 { $data = preg_replace('/[^\a-zA-Z0-9]/','',$data);
15   return $data;
16   unset($data);
17 }
18
19 public function tampil_semua_data()
20 { $client = curl_init($this->url);
21   curl_setopt($client,CURLOPT_RETURNTRANSFER,1);
22   $response = curl_exec($client);
23   curl_close($client);
24   $data = json_decode($response);
25   // mengembalikan data
26   return $data;
27   // menghapus variabel dari memory
28   unset($data,$client,$response);
29 }
30
31 public function tampil_data($id_barang)
32 { $id_barang = $this->filter($id_barang);
33   $client = curl_init($this->url."?aksi=tampil&id_barang=".$id_barang);
34   curl_setopt($client,CURLOPT_RETURNTRANSFER,1);
35   $response = curl_exec($client);
36   curl_close($client);
37   $data = json_decode($response);
38
39   return $data;
40   unset($id_barang,$client,$response,$data);
41 }
42
43 public function tambah_data($data)
44 { $data = '{ "id_barang":' . $data['id_barang'] . ','
45   "nama_barang":' . $data['nama_barang'] . ','
46   "aksi":' . $data['aksi'] . '
47   }';
48   $c = curl_init();
49   curl_setopt($c,CURLOPT_URL,$this->url);
50   curl_setopt($c,CURLOPT_RETURNTRANSFER,true);
51   curl_setopt($c,CURLOPT_POST,true);
52   curl_setopt($c,CURLOPT_POSTFIELDS,$data);
53   $response = curl_exec($c);
54   curl_close($c);
55   unset($data,$c,$response);
56 }
57
58 public function ubah_data($data)
59 { $data = '{ "id_barang":' . $data['id_barang'] . ','
60   "nama_barang":' . $data['nama_barang'] . ','
61   "aksi":' . $data['aksi'] . '
62   }';
63   $c = curl_init();
64   curl_setopt($c,CURLOPT_URL,$this->url);
65   curl_setopt($c,CURLOPT_RETURNTRANSFER,true);
66   curl_setopt($c,CURLOPT_POST,true);
67   curl_setopt($c,CURLOPT_POSTFIELDS,$data);
68   $response = curl_exec($c);
69   curl_close($c);
70   unset($data,$c,$response);
71 }
72
73 public function hapus_data($data)
74 { $id_barang = $this->filter($data['id_barang']);
75   $data = '{ "id_barang":' . $id_barang . ','
76   "aksi":' . $data['aksi'] . '
77   }';
78   $c = curl_init();
79   curl_setopt($c,CURLOPT_URL,$this->url);
80   curl_setopt($c,CURLOPT_RETURNTRANSFER,true);
81   curl_setopt($c,CURLOPT_POST,true);
82   curl_setopt($c,CURLOPT_POSTFIELDS,$data);
83   $response = curl_exec($c);
84   curl_close($c);
85   unset($id_barang,$data,$c,$response);
86 }
87
88 // function yang terakhir kali di-load saat class dipanggil
89 public function __destruct()
90 { // hapus variabel dari memory
91   unset($this->url);
92 }
93 }
94
95 $url = 'http://192.168.56.2/restful-json-toko/server/server.php';
96 // buat objek baru dari class Client
97 $abc = new Client($url);
98 ?>
```

Gambar 4. Source code Client.php di Windows

8.2.4 *Source code proses.php di Windows*

```
1 <?php
2 include "Client.php";
3
4 if ($_POST['aksi']=='tambah')
5 {   $data = array("id_barang"=>$_POST['id_barang'],
6                 "nama_barang"=>$_POST['nama_barang'],
7                 "aksi"=>$_POST['aksi']);
8     $abc->tambah_data($data);
9     header('location:index.php?page=daftar-data');
10 } else if ($_POST['aksi']=='ubah')
11 {   $data = array("id_barang"=>$_POST['id_barang'],
12                 "nama_barang"=>$_POST['nama_barang'],
13                 "aksi"=>$_POST['aksi']);
14     $abc->ubah_data($data);
15     header('location:index.php?page=daftar-data');
16 } else if ($_GET['aksi']=='hapus')
17 {   $data = array("id_barang"=>$_GET['id_barang'],
18                 "aksi"=>$_GET['aksi']);
19     $abc->hapus_data($data);
20     header('location:index.php?page=daftar-data');
21 }
22 unset($abc,$data);
23 ?>
```

Gambar 5. *Source code proses.php di Windows*

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

8.2.5 Source code *index.php* di *Windows*

```
1 <?php
2 include "Client.php";
3 ?>
4 <!doctype html>
5 <html>
6 <head>
7 <title></title>
8 </head>
9 <body>
10 <a href="?page=home">Home</a> | <a href="?page=tambah">Tambah Data</a> | <a href="?page=daftar-data">Data Server</a>
11 <br/><br/>
12
13 <fieldset>
14 <? if ($ GET['page']=='tambah') { ?>
15 <legend>Tambah Data</legend>
16 <form name="form" method="POST" action="proses.php">
17 <input type="hidden" name="aksi" value="tambah"/>
18 <label>ID Barang</label>
19 <input type="text" name="id_barang"/>
20 <br/>
21 <label>Nama Barang</label>
22 <input type="text" name="nama_barang"/>
23 <br/>
24 <button type="submit" name="simpan">Simpan</button>
25 </form>
26
27 <? } elseif ($ GET['page']=='ubah') {
28 $r = $abc->tampil_data($ GET['id_barang']);
29 ?>
30 <legend>Ubah Data</legend>
31 <form name="form" method="post" action="proses.php">
32 <input type="hidden" name="aksi" value="ubah"/>
33 <input type="hidden" name="id_barang" value="<?=$r->id_barang?" />
34 <label>ID Barang</label>
35 <input type="text" value="<?=$r->id_barang?" disabled>
36 <br/>
37 <label>Nama Barang</label>
38 <input type="text" name="nama_barang" value="<?=$r->nama_barang?">
39 <br/>
40 <button type="submit" name="ubah">Ubah</button>
41 </form>
42
43 <? unset($r);
44 } else if ($ GET['page']=='daftar-data') {
45 ?>
46 <legend>Daftar Data Server</legend>
47 <table border="1">
48 <tr><th width="5%">No</th>
49 <th width="10%">ID Barang</th>
50 <th width="75%">Nama</th>
51 <th width="5%" colspan="2">Aksi</th>
52 </tr>
53 <? $no = 1;
54 $data_array = $abc->tampil_semua_data();
55 foreach ($data_array as $r) {
56 <tr><td><?=$no?></td>
57 <td><?=$r->id_barang?></td>
58 <td><?=$r->nama_barang?></td>
59 <td><a href="?page=ubah&id_barang=<?=$r->id_barang?">Ubah</a></td>
60 <td><a href="proses.php?aksi=hapus&id_barang=<?=$r->id_barang?" onclick="return confirm('Apakah Anda ingin menghapus data ini?')">Hapus</a></td>
61 </tr>
62 <? $no++;
63 }
64 unset($data_array,$r,$no);
65 ?>
66 </table>
67
68 <? } else { ?>
69 <legend>Home</legend>
70 Aplikasi sederhana ini menggunakan RESTful dengan format data JSON (JavaScript Object Notation).
71 </fieldset>
72 <? } ?>
73 </body>
74 </html>
```

Gambar 6. Source code *index.php* di *Windows*

Akses melalui *web browser http://localhost/restful-json-toko/client/index.php* di *Windows*. Coba jalankan aplikasi sederhana ini dengan menambah, menampilkan, mengubah dan menghapus datanya. Data akan dikirim dari *Windows* ke *Debian Server* dan data hanya disimpan di *database Debian Server*. Data yang tersimpan tersebut akan ditampilkan ke *Client* di *Windows*.



Gambar 8. Laman *Client*

MODUL 9

JSON Web Token (JWT)

9.1 Pengertian

JSON Web Token, yang berarti token ini menggunakan *JSON (Javascript Object Notation)* berbentuk string panjang yang sangat random, lalu token ini memungkinkan kita untuk mengirimkan data yang dapat diverifikasi oleh dua pihak atau lebih.

Bagaimana cara JWT bekerja ?

Seperti password jadi ketika users berhasil melakukan Login maka server akan memberikan sebuah Token. Nanti Token tersebut akan disimpan oleh users pada Local Storage atau Cookies Browser dan bila users ingin mengakses halaman halaman tertentu maka harus menyertakan token tersebut. Untuk itu users akan mengirim balik token yang dikasih diawal tadi sebagai bukti bila user ini sudah melakukan login. Sekarang kita akan lihat struktur dasarnya, dimana token terdiri dari tiga bagian yaitu yang pertama *header* lalu kedua bagian *payload* atau data dan yang ketiga adalah bagian *verify signature*.

Struktur JWT

- **Header**

Header biasanya terdiri dari dua bagian: jenis token, yaitu JWT, dan algoritma penandatanganan yang digunakan, seperti HMAC SHA256 atau RSA.

Contoh :

```
{ "alg": "HS256", "typ": "JWT" }
```

- **Payload**

Payload merupakan informasi atau data yang ingin kita kirimkan. Dalam penerapannya di otentikasi atau pun otorisasi, biasanya data ini berupa data yang sifatnya unik bagi user, seperti: email, id/uuid, dan juga data yang berkaitan dengan otorisasi seperti role, karena data tersebut akan digunakan sebagai tanda pengenal si pengirim token.

Contoh :

```
{ "sub": "1234567890",  
  "name": "Minpo",  
  "admin": true }
```

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

- **Verify Signature**

Verify Signature adalah hasil dari Hash atau gabungan dari isi encode Header dan Payloadnya lalu ditambahkan kode secretnya. Signature ini berguna untuk memverifikasi bahwa header maupun payload yang ada dalam token tidak berubah dari nilai aslinya (karena untuk membuat payload dan header palsu itu cukup mudah).

Signature-nya sendiri tidak mungkin dapat diakali, karena sudah dalam berbentuk hash; yang mana adalah fungsi satu arah (tidak dapat dikembalikan ke nilai semula), dan meski kita tahu algoritma hashing-nya, kita juga memerlukan secret key yang mana hanya si pembuat aplikasi yang tahu.

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  secret)
```

Jika kita satukan semuanya, hasil ketiga bagian tersebut akan digabung dan otomatis di encode menjadi Token string random panjang seperti berikut

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ  
JzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpva  
G4gRG9lIiwiaWF0IjoxNTE2MzIyMDUyLmV5LmV5  
wRJSMeKKF2QT4fwpMeJf36P0k6yJV_adQssw5c
```

Jika isi Header atau Payload diubah, maka isi Signature menjadi tidak valid. Satu hal lagi, JWT ini tidak tergantung sama bahasa program tertentu sehingga kita dapat mengimplementasikan di semua bahasa pemrograman.

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

9.2 Aplikasi

Aplikasi Toko menggunakan *JWT - CRUD* sederhana, dimana **data akan dikirim dari Client (Windows dengan IP Address 192.168.56.1) ke Server (Debian Server dengan IP Address 192.168.56.xx) dan data hanya disimpan pada database Debian Server.** Database tetap menggunakan database “toko” dan tabel “barang” pada *Debian Server*. Langkah selanjutnya adalah sebagai berikut:

1. Pada *database* toko, buat **tabel pengguna** yang terdiri dari **id_pengguna (char 20, Primary Key), nama (varchar 50), pin (char 32)**. Isi data pengguna dan pin tanpa dienkripsi.
2. Buat *folder /opt/lampp/htdocs/jwt-toko/server/* pada *Debian Server* lalu buat *file Database.php, core.php* dan *server.php*. Buat juga *folder lib* yang berisi *library firebase/php-jwt v5.5.1*, download di <https://github.com/firebase/php-jwt/archive/refs/tags/v5.5.1.zip> lalu ekstrak *file zip* tersebut pada *folder lib* dan *rename folder php-jwt-5.5.1* menjadi *php-jwt* sehingga *path library firebase/php-jwt* berada di */opt/lampp/htdocs/jwt-toko/server/lib/php-jwt*. *Library firebase/php-jwt v5.5.1* mendukung *PHP v5.6* yang kita gunakan. Untuk mendukung *PHP* yang terbaru, gunakan *library firebase/php-jwt* yang terbaru.
3. Pada *web browser firefox*, *install add ons "RESTED"*.

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

9.2.1 *Source code Database.php di Debian Server*

```
1 <?php
2 class Database
3 {
4     private $host="localhost";
5     private $dbname="toko";
6     private $user="root";
7     private $password="root";
8     private $port="3306";
9     private $conn;
10
11 // function yang pertama kali di-load saat class dipanggil
12 public function __construct()
13 {
14     // koneksi database
15     try
16     {
17         $this->conn = new PDO("mysql:host=$this->host;port=$this->port;dbname=$this->dbname;charset=utf8",$this->user,$this->password);
18     }
19     catch (PDOException $e)
20     {
21         echo "Koneksi gagal";
22     }
23 }
24
25 public function login($data)
26 {
27     $query = $this->conn->prepare("SELECT id_pengguna,nama FROM pengguna WHERE id_pengguna=? AND pin=?");
28     $query->execute(array($data['id_pengguna'],$data['pin']));
29     // mengambil satu data dengan fetch
30     $data = $query->fetch(PDO::FETCH_ASSOC);
31     return $data;
32     $query->closeCursor();
33     unset($data);
34 }
35
36 public function tampil_semua_data()
37 {
38     $query = $this->conn->prepare("select id_barang,nama_barang from barang order by id_barang");
39     $query->execute();
40     // mengambil banyak data dengan fetchAll
41     $data = $query->fetchAll(PDO::FETCH_ASSOC);
42     return $data;
43     $query->closeCursor();
44     unset($data);
45 }
46
47 public function tampil_data($id_barang)
48 {
49     $query = $this->conn->prepare("select id_barang,nama_barang from barang where id_barang=?");
50     $query->execute(array($id_barang));
51     // mengambil satu data dengan fetch
52     $data = $query->fetch(PDO::FETCH_ASSOC);
53     // mengembalikan data
54     return $data;
55     // hapus variable dari memory
56     $query->closeCursor();
57     unset($id_barang,$data);
58 }
59
60 public function tambah_data($data)
61 {
62     $query = $this->conn->prepare("insert ignore into barang (id_barang,nama_barang) values (?,?)");
63     $query->execute(array($data['id_barang'],$data['nama_barang']));
64     $query->closeCursor();
65     unset($data);
66 }
67
68 public function ubah_data($data)
69 {
70     $query = $this->conn->prepare("update barang set nama_barang=? where id_barang=?");
71     $query->execute(array($data['nama_barang'],$data['id_barang']));
72     $query->closeCursor();
73     unset($data);
74 }
75
76 public function hapus_data($id_barang)
77 {
78     $query = $this->conn->prepare("delete from barang where id_barang=?");
79     $query->execute(array($id_barang));
80     $query->closeCursor();
81     unset($id_barang);
82 }
83 }
84 ?>
```

Gambar 1. *Source code Database.php di Debian Server*

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

9.2.2 Source code *core.php* di *Debian Server*

```
1 <?php
2 date_default_timezone_set('Asia/Jakarta');
3
4 // Buat RSA Key 1024 bit atau 2048 bit di Linux/FreeBSD
5 // $ openssl genrsa 1024
6 // $ openssl genrsa 2048
7 $key = "MIICWwIBAAKBgQCZDcDCKRS/ml0muH29w+1vgegJpK3eUrA0yx1KU1vSLABIQwoiNaLWv7RwCagdL+wIUy1lvowA0Y
r+Twr+tjy3jiWY2xLLA3+8g8+kryylf4o3gh5JVycxAe3VsLWmYZUMHE4CG2ZMKiryup4Zseol8o8pLtS/BNGL+V3MCM1SKQID
AQABAoGAAsR15MzbXC+NwAliWykMxQRjTrFul32FRB8cE3j4Yw96ndPgfgfcPuf0emwiRwzTxr7CM93DCjOAKOMC/
uIKrPvTa8l0eQa1jGP6MeVPazcwP5XHbkunfF+jVIAzSvsnf3m7W+PLp1KikgiGFyvn0jR5Sdfdn/uZr79zHGUAwECQQD
HdLqxsUJfecUeam3LNTjas6pTINLqLByEQxR6IqTltvxzm26JckV5pNm50occk4FVVAJ6yaB0qMelMmi9UU/
JAKEAxHFwC3yNyUXkdjUQ+bFMBW9ugoR4Tjr/ukP2dB4YarfXYMaYOCPR4ASR1ie2RrUJHdA4XDNuazeg4+4n/47fYQJBA
MYhCTcEy97lqk7NcCU0yDZP1Ljg2ULu8KDdtChe2YJ0Htiggm1X1A31mQFYlublXT478Gj0hAJDtD14y90bykCQBjJ78ejSgk
SBrs80w4oAVjWPO2/ZacJjuekV99DR0hi5ozRwrei3YMVUInjrP6zLZLTgykvWQHGNujl7qozD0ECP2z9qx2S7YC/2EbIFHzKH
tCeK5Ms/FcYvN7syprpHUGRjuc5i/TGRk0QIYerWxIIbcfwqVth29U9tEw8tu1dg==";
8 $issued_at = time();
9 $expiration_time = $issued_at+(60*60); // valid selama 1 jam
10 $issuer = "RestApiAuthJWT";
11 ?>
```

Gambar 2. Source code *core.php* di *Debian Server*

Buat *RSA Key 1024 bit atau 2048 bit* di *Debian Server* dengan perintah di *console/terminal* sebagai berikut dan masukkan ke *variable \$key*:

\$ openssl genrsa 1024

atau

\$ openssl genrsa 2048

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

9.2.3 Source code *server.php* di Debian Server

```
1 <?php
2 error_reporting(1);
3
4 include_once 'core.php';
5 include_once 'lib/php-jwt/src/BeforeValidException.php';
6 include_once 'lib/php-jwt/src/ExpiredException.php';
7 include_once 'lib/php-jwt/src/SignatureInvalidException.php';
8 include_once 'lib/php-jwt/src/JWT.php';
9 use \Firebase\JWT\JWT;
10
11 include_once "Database.php";
12 $abc = new Database();
13
14 if (isset($_SERVER['HTTP_ORIGIN'])) {
15     header("Access-Control-Allow-Origin: {$_SERVER['HTTP_ORIGIN']}");
16     header("Content-Type: application/json; charset=utf-8");
17     header('Access-Control-Allow-Credentials: true');
18     header('Access-Control-Max-Age: 3600'); // cache selama 1 jam
19 }
20
21 if ($_SERVER['REQUEST_METHOD'] == 'OPTIONS') {
22     if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_METHOD']))
23         header("Access-Control-Allow-Methods: GET,POST,OPTIONS");
24     if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']))
25         header("Access-Control-Allow-Headers: {$_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']}");
26     exit(0);
27 }
28
29 $postdata = file_get_contents("php://input");
30 $data = json_decode($postdata);
31
32 function filter($data)
33 { $data = preg_replace('/[^a-zA-Z0-9]/', '', $data);
34     return $data;
35     unset($data);
36 }
37
38 if ($_SERVER['REQUEST_METHOD'] == 'POST' and isset($data->id_pengguna) and isset($data->pin))
39 { $data2['id_pengguna'] = $data->id_pengguna;
40   $data2['pin'] = $data->pin;
41
42   // cek login pengguna
43   if ($abc->login($data2)){
44       // generate json web token (jwt)
45       $token = array(
46           "iat" => $issued_at,
47           "exp" => $expiration_time,
48           "iss" => $issuer,
49           "data" => array(
50               "id_pengguna" => $data2['id_pengguna'],
51               "pin" => $data2['pin']
52           )
53       );
54       // set response code
55       http_response_code(200);
56       // generate jwt
57       $jwt = JWT::encode($token,$key);
58       echo json_encode(
59           array(
60               "message" => "Login sukses",
61               "jwt" => $jwt
62           )
63       );
64   } else { // login gagal
65       // set response code
66       http_response_code(401);
```

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

```
67     echo json_encode(array("message" => "Login gagal"));
68 }
69 } elseif ($ SERVER['REQUEST_METHOD']=='POST')
70 {
71     $jwt = $data->jwt;
72     $aksi = $data->aksi;
73     $id_barang = $data->id_barang;
74     $nama_barang = $data->nama_barang;
75
76     try { // decode jwt
77         $decoded = JWT::decode($jwt,$key,array('HS256'));
78
79         if ($aksi == 'tambah')
80         { $data2=array( 'aksi' => $aksi,
81                       'id_barang' => $id_barang,
82                       'nama_barang' => $nama_barang
83                     );
84           $abc->tambah_data($data2);
85         } elseif ($aksi == 'ubah')
86         { $data2=array( 'aksi' => $aksi,
87                       'id_barang' => $id_barang,
88                       'nama_barang' => $nama_barang
89                     );
90           $abc->ubah_data($data2);
91         } elseif ($aksi == 'hapus')
92         { $data2=array( 'aksi' => $aksi,
93                       'id_barang' => $id_barang
94                     );
95           $abc->hapus_data($id_barang);
96         }
97
98         // set response code
99         http_response_code(200);
100        echo json_encode($data2);
101    }
102
103    // jika decode gagal, berarti jwt tidak valid
104    catch (Exception $e)
105    { // set response code
106        http_response_code(401);
107        echo json_encode(array(
108            "message" => "Access denied"
109        ));
110    }
111 } elseif ($ SERVER['REQUEST_METHOD']=='GET')
112 {
113     $jwt = $_GET['jwt'];
114     try { // decode jwt
115         $decoded = JWT::decode($jwt,$key,array('HS256'));
116
117         if ( ($ GET['aksi']=='tampil') and (isset($_GET['id_barang'])) )
118         { $id_barang = filter($_GET['id_barang']);
119           $data=$abc->tampil_data($id_barang);
120         } else //menampilkan semua data
121         { $data = $abc->tampil_semua_data();
122         }
123
124         // set response code
125         http_response_code(200);
126         echo json_encode($data);
127     }
128
129     // jika decode gagal, berarti jwt tidak valid
130     catch (Exception $e)
131     { // set response code
132         http_response_code(401);
133         echo json_encode(array(
134             "message" => "Access denied"
135         ));
136     }
137 } else { // error jika tanpa jwt
138     // set response code
139     http_response_code(401);
140     echo json_encode(array("message" => "Access denied"));
141 }
142
143 unset($abc,$postdata,$data,$data2,$token,$key,$issued_at,$expiration_time,$issuer,$jwt,$decoded,$id_barang,$nama_barang,$aksi,$e);
144 ?>
```

Gambar 3. Source code server.php di Debian Server

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

9.2.4 Login di *RESTED*

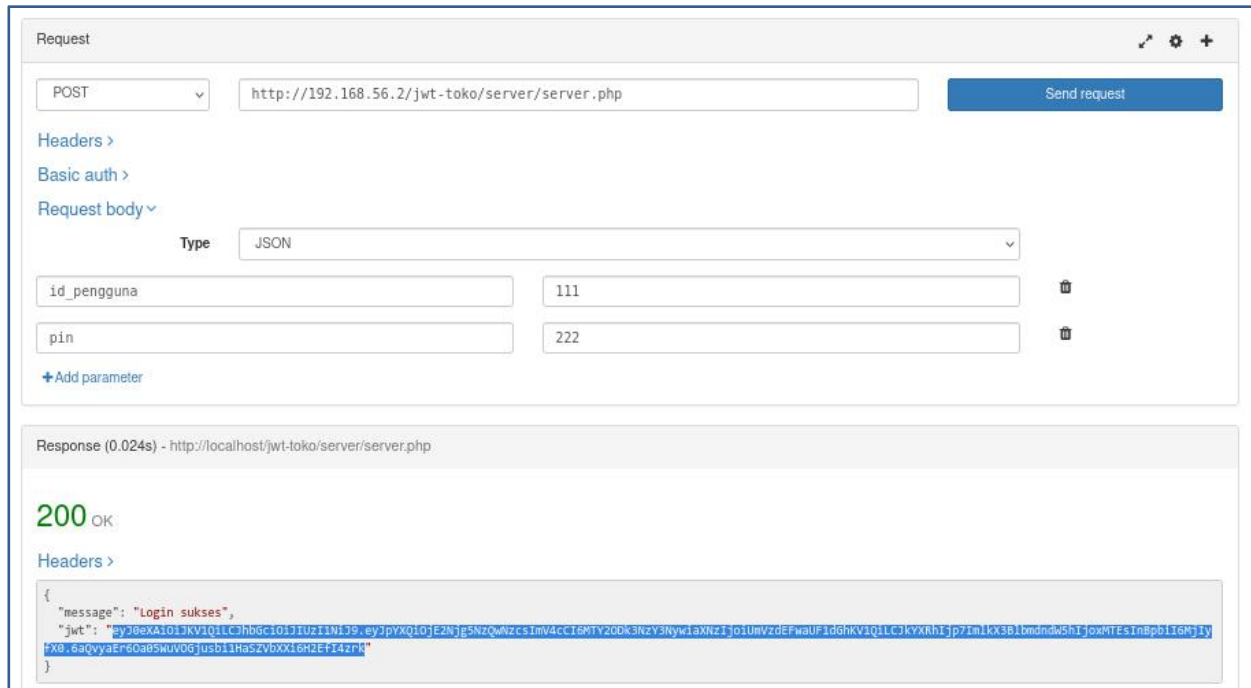
Pada *web browser firefox*, buka *add ons "RESTED"* yang sudah terinstall. Cobalah untuk login di *RESTED* dengan memasukkan beberapa item sebagai berikut:

Request: POST

Type: JSON

url: http://192.168.56.xx/jwt-toko/server/server.php

Parameter: id_pengguna, pin



Gambar 4. *Login di RESTED*

Catat *output jwt* yang ada di paling bawah. *jwt* tersebut akan digunakan untuk proses selanjutnya.

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

9.2.5 Tambah data di *RESTED*

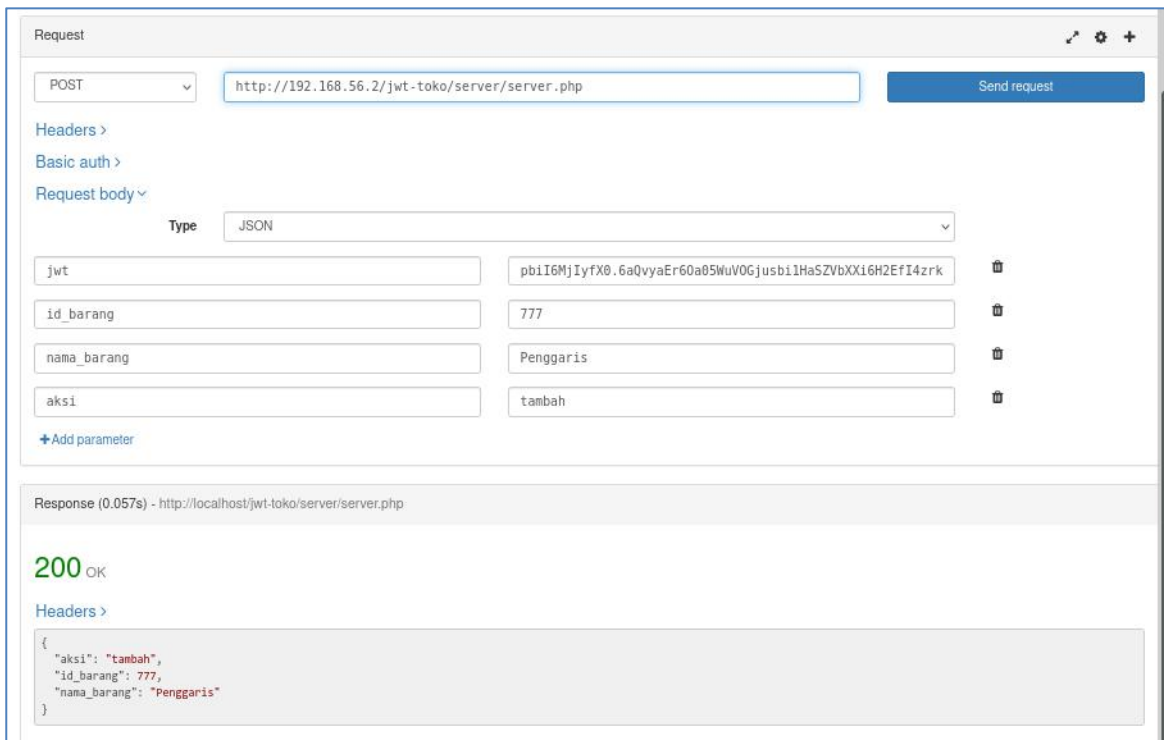
Cobalah untuk tambah data di *RESTED* dengan memasukkan beberapa item sebagai berikut:

Request: POST

Type: JSON

url: http://192.168.56.xx/jwt-toko/server/server.php

Parameter: jwt (output hasil login), id_barang, nama_barang, aksi (tambah)



Gambar 5. Tambah data di *RESTED*

Kemudian lihat data di tabel barang menggunakan *SQLYog* atau *PHPMYAdmin*.

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

9.2.6 Ubah data di *RESTED*

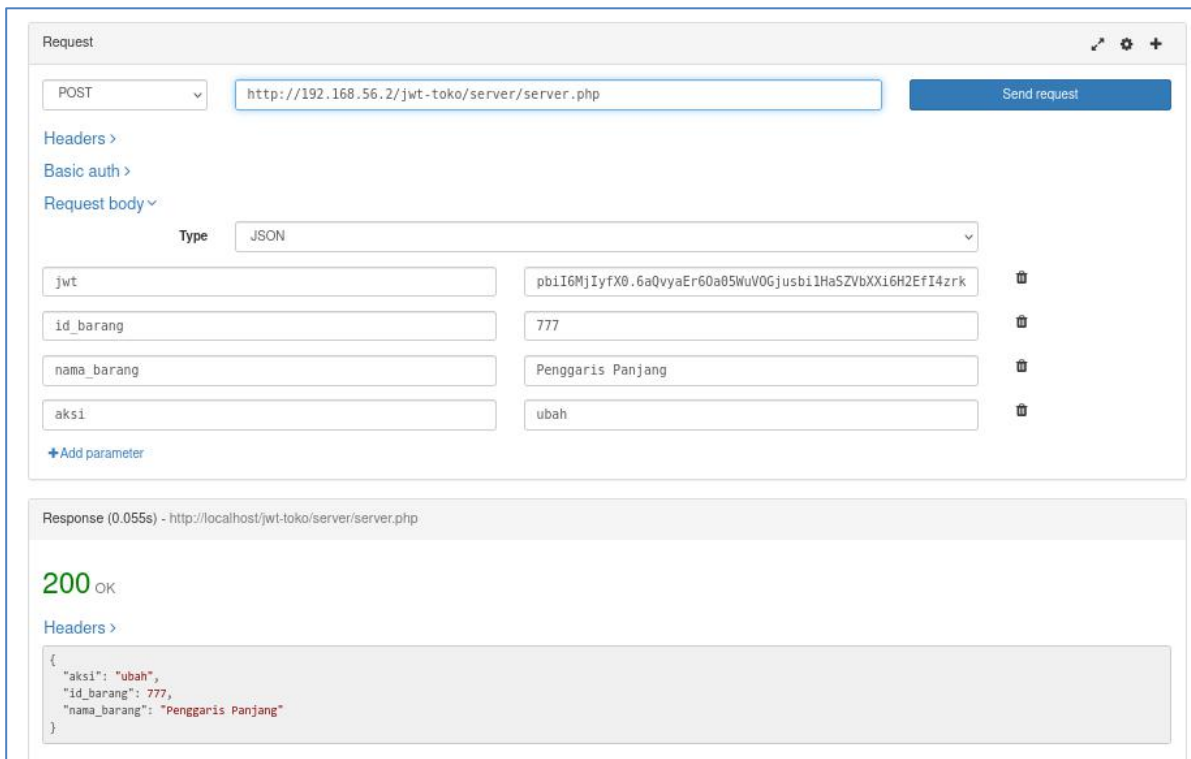
Cobalah untuk ubah data di *RESTED* dengan memasukkan beberapa item sebagai berikut:

Request: POST

Type: JSON

url: http://192.168.56.xx/jwt-toko/server/server.php

Parameter: jwt (output hasil login), id_barang, nama_barang, aksi (ubah)



Gambar 6. Ubah data di *RESTED*

Kemudian lihat data di tabel barang menggunakan *SQLYog* atau *PHPMYAdmin*.

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

9.2.7 Hapus data di *RESTED*

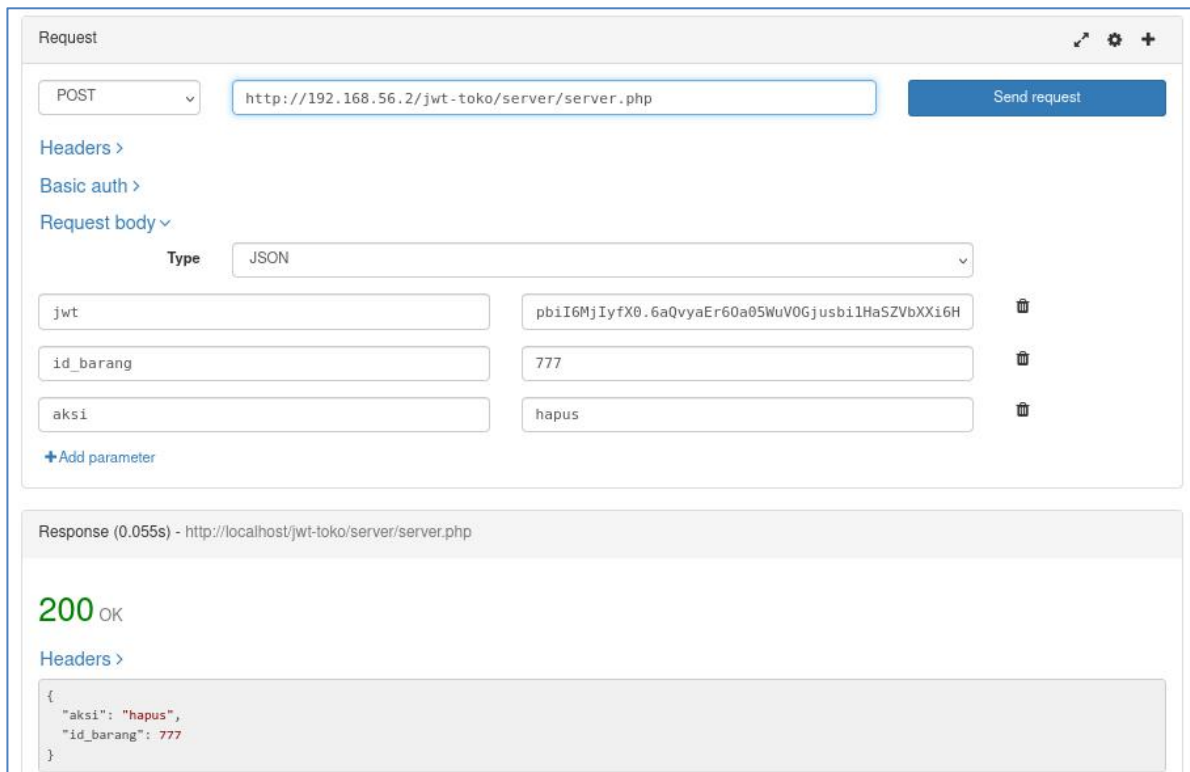
Cobalah untuk hapus data di *RESTED* dengan memasukkan beberapa item sebagai berikut:

Request: POST

Type: JSON

url: http://192.168.56.xx/jwt-toko/server/server.php

Parameter: jwt (output hasil login), id_barang, aksi (hapus)



Gambar 7. Hapus data di *RESTED*

Kemudian lihat data di tabel barang menggunakan *SQLYog* atau *PHPMysqlAdmin*.

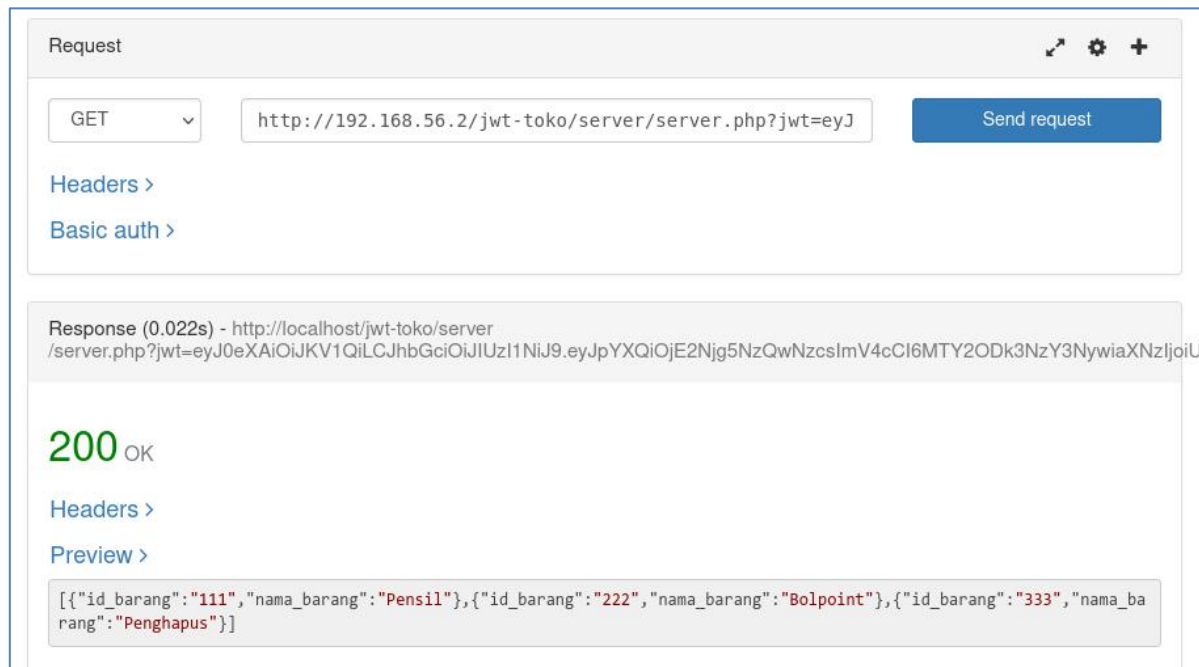
PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

9.2.8 Tampil semua data di *RESTED*

Cobalah untuk menampilkan semua data di *RESTED* dengan memasukkan beberapa item sebagai berikut:

Request: GET

url: http://192.168.56.xx/jwt-toko/server/server.php?jwt=xxxxxxxxxxxxx (output hasil login)



Gambar 8. Tampil semua data di *RESTED*

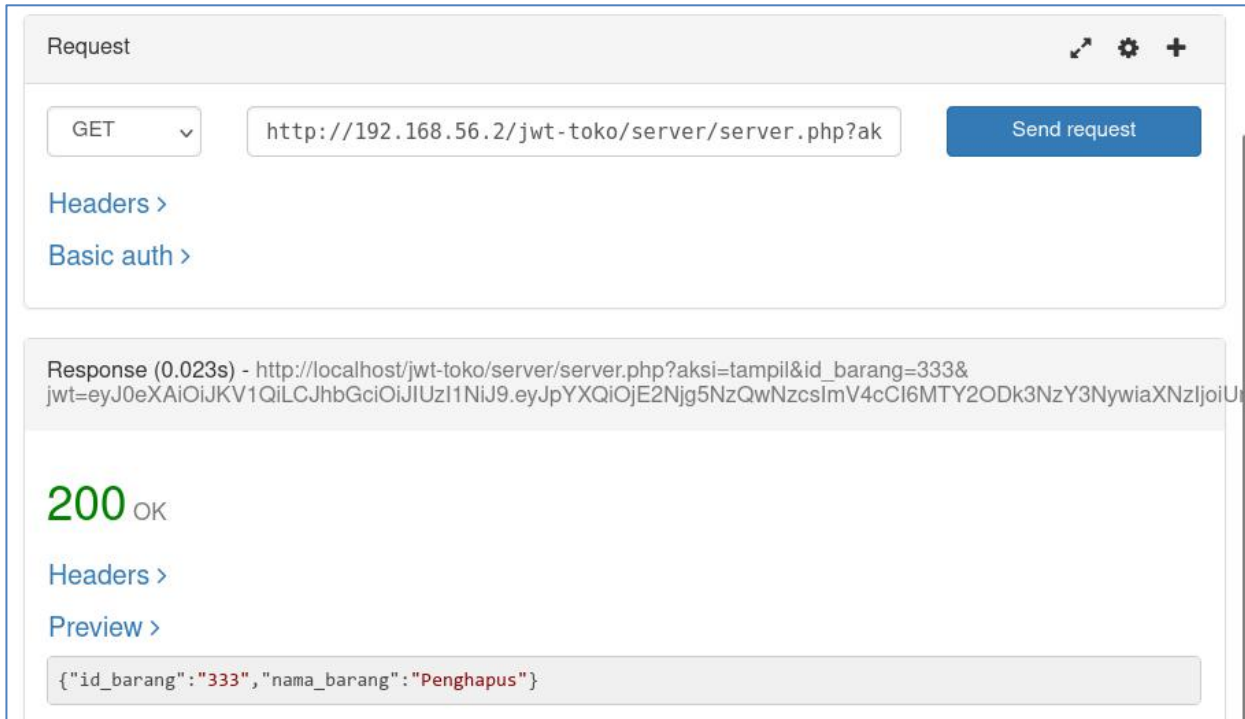
PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

9.2.9 Tampil salah satu data di *RESTED*

Cobalah untuk menampilkan salah satu data di *RESTED* dengan memasukkan beberapa item sebagai berikut:

Request: GET

url: http://192.168.56.xx/
toko/server/server.php?aksi=tampil&id_barang=333&jwt=xxxxxxxxxxxxxx (output hasil login)



Gambar 9. Tampil salah data data di *RESTED*

MODUL 10

JSON Web Token (JWT) - II

10.1 Aplikasi

Aplikasi Toko menggunakan *JWT - CRUD* sederhana, dimana **data akan dikirim dari Client (Windows dengan IP Address 192.168.56.1) ke Server (Debian Server dengan IP Address 192.168.56.xx) dan data hanya disimpan pada database Debian Server.** Database tetap menggunakan database “*toko*”, tabel “*barang*” dan tabel “*pengguna*” pada *Debian Server*. Langkah selanjutnya adalah sebagai berikut:

1. Pada tabel “*pengguna*”, masukkan data pengguna dan pin dienkripsi menggunakan *md5*, misalnya:

```
INSERT INTO pengguna (id_pengguna,nama,pin) VALUES ('111','Muhammad',MD5('222'));
```
2. Untuk server menggunakan folder */opt/lampp/htdocs/jwt-toko/server/* pada *Debian Server* lalu ubah *source code Database.php, core.php* dan *server.php*. Pada aplikasi ini masih menggunakan library *firebase/php-jwt v5.5.1* yang berada di */opt/lampp/htdocs/jwt-toko/server/lib/php-jwt*, seperti pada modul sebelumnya. Library *firebase/php-jwt v5.5.1* mendukung *PHP v5.6* yang kita gunakan. Untuk mendukung *PHP* yang terbaru, gunakan library *firebase/php-jwt* yang terbaru.
3. Buat folder */opt/lampp/htdocs/jwt-toko/client/* pada *Debian Server* lalu buat file *Client.php, proses.php* dan *index.php*.

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

10.1.1 *Source code Database.php di Debian Server*

```
1 <?php
2 class Database
3 {
4     private $host="localhost";
5     private $dbname="toko";
6     private $user="root";
7     private $password="root";
8     private $port="3306";
9     private $conn;
10
11 // function yang pertama kali di-load saat class dipanggil
12 public function __construct()
13 {
14     try
15     {
16         $this->conn = new PDO("mysql:host=$this->host;port=$this->port;dbname=$this->dbname;charset=utf8",$this->user,$this->password);
17     }
18     catch (PDOException $e)
19     {
20         echo "Koneksi gagal";
21     }
22 }
23
24 public function login($data)
25 {
26     $query = $this->conn->prepare("SELECT id_pengguna,nama FROM pengguna WHERE id_pengguna=? AND pin=MD5(?)");
27     $query->execute(array($data['id_pengguna'],$data['pin']));
28     $data = $query->fetch(PDO::FETCH_ASSOC);
29     return $data;
30     $query->closeCursor();
31     unset($data);
32 }
33
34 public function tampil semua data()
35 {
36     $query = $this->conn->prepare("select id_barang,nama_barang from barang order by id_barang");
37     $query->execute();
38     // mengambil banyak data dengan fetchAll
39     $data = $query->fetchAll(PDO::FETCH_ASSOC);
40     // mengembalikan data
41     return $data;
42     // hapus variable dari memory
43     $query->closeCursor();
44     unset($data);
45 }
46
47 public function tampil data($id_barang)
48 {
49     $query = $this->conn->prepare("select id_barang,nama_barang from barang where id_barang=?");
50     $query->execute(array($id_barang));
51     // mengambil satu data dengan fetch
52     $data = $query->fetch(PDO::FETCH_ASSOC);
53     return $data;
54     $query->closeCursor();
55     unset($id_barang,$data);
56 }
57
58 public function tambah data($data)
59 {
60     $query = $this->conn->prepare("insert ignore into barang (id_barang,nama_barang) values (?,?)");
61     $query->execute(array($data['id_barang'],$data['nama_barang']));
62     $query->closeCursor();
63     unset($data);
64 }
65
66 public function ubah data($data)
67 {
68     $query = $this->conn->prepare("update barang set nama_barang=? where id_barang=?");
69     $query->execute(array($data['nama_barang'],$data['id_barang']));
70     $query->closeCursor();
71     unset($data);
72 }
73
74 public function hapus data($id_barang)
75 {
76     $query = $this->conn->prepare("delete from barang where id_barang=?");
77     $query->execute(array($id_barang));
78     $query->closeCursor();
79     unset($id_barang);
80 }
81 }
82 }>
```

Gambar 1. *Source code Database.php di Debian Server*

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

10.1.2 *Source code core.php* di *Debian Server*

```
1 <?php
2 date_default_timezone_set('Asia/Jakarta');
3
4 // Buat RSA Key 1024 bit atau 2048 bit di Linux/FreeBSD
5 // $ openssl genrsa 1024
6 // $ openssl genrsa 2048
7 $key = "MIICWwIBAAKBgQCZDcDCKRS/ml0muH29w+1vgegJpK3eUrA0yx1KU1vSLABIQwoiNaLWv7RwCagdL+wIUy1lvvWA0Y
r+Twr+tjy3jiWY2xLLA3+8g8+kryylf4o3gh5JVycxAe3VsLWmYZUMHE4CG2ZMKiryup4Zseol8o8pLtS/BNGL+V3MCM1SKQID
AQABAoGAAsR15MzbXC+NwAliWykMxQRjTrFUL32FRB8cE3j4Yw96ndPgfgfcPuf0emwiRwzTxr7CM93DCjOAKOMC/
uIKrPvTa8l0eQa1jGP6MeVPazcwPT5XHbkunfF+jVIAzSvsnf3m7W+PLp1KikgiGFyvn0jR5Sdfdn/uZr79zHGUAwECQQD
HdLqxSUJfecUeam3LNTjas6pTINLqLByEQxR6IqTltvxzm26JCKV5pNm50occk4FVVAJ6yaB0qMe1Mmi9UU/
JAKEAxHFwC3yNyUXkdjUQ+bFMBW9ugoR4Tjr/ukP2dB4YArfXYMaYOCPR4ASR1ie2RrUJHdA4XDNuazeg4+4n/47fYQJBA
MYhCTcEy97lqk7NcCU0yDZP1Ljg2ULu8KDdtaChe2YJ0Htiggm1X1A31mQFYlublXT478Gj0hAJDtD14y90bykCQBjJ78ejSgk
SBrs80w4oAVjWP02/ZacJjuekV99DR0hi5ozRwrei3YMVUInjrP6zLZLTgykvWQHGNujl7qozD0ECP2z9qx2S7YC/2EbIFHzKH
tCeK5Ms/FcYvN7syprpHUGRjuc5i/TGRk0QIYerWxIIbcfwqVth29U9tEw8tu1dg==";
8 $issued_at = time();
9 $expiration_time = $issued_at+(60*60); // valid selama 1 jam
10 $issuer = "RestApiAuthJWT";
11 ?>
```

Gambar 2. *Source code core.php* di *Debian Server*

Buat *RSA Key 1024 bit* atau *2048 bit* di *Debian Server* dengan perintah di *console/terminal* sebagai berikut dan masukkan ke *variable \$key*:

\$ openssl genrsa 1024

atau

\$ openssl genrsa 2048

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

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

```
1 <?php
2 error_reporting(1);
3
4 include_once 'core.php';
5 include_once 'lib/php-jwt/src/BeforeValidException.php';
6 include_once 'lib/php-jwt/src/ExpiredException.php';
7 include_once 'lib/php-jwt/src/SignatureInvalidException.php';
8 include_once 'lib/php-jwt/src/JWT.php';
9 use \Firebase\JWT\JWT;
10
11 include_once "Database.php";
12 $abc = new Database();
13
14 if (isset($_SERVER['HTTP_ORIGIN'])) {
15     header("Access-Control-Allow-Origin: {$_SERVER['HTTP_ORIGIN']}");
16     header("Content-Type: application/json; charset=UTF-8");
17     header('Access-Control-Allow-Credentials: true');
18     header('Access-Control-Max-Age: 3600'); // cache selama 1 jam
19 }
20
21 if ($_SERVER['REQUEST_METHOD'] == 'OPTIONS') {
22     if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_METHOD']))
23         header("Access-Control-Allow-Methods: GET, POST, OPTIONS");
24     if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']))
25         header("Access-Control-Allow-Headers: {$_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']}");
26     exit(0);
27 }
28
29 $postdata = file_get_contents("php://input");
30 $data = json_decode($postdata);
31
32 function filter($data)
33 { $data = preg_replace('/[^a-zA-Z0-9]/', '', $data);
34   return $data;
35   unset($data);
36 }
37
38 if ($_SERVER['REQUEST_METHOD']=='POST' and isset($data->id_pengguna) and isset($data->pin))
39 { $data2['id_pengguna'] = $data->id_pengguna;
40   $data2['pin'] = $data->pin;
41
42   // cek login pengguna
43   $data3 = $abc->login($data2);
44   if ($data3){
45       // generate json web token (jwt)
46       $token = array(
47           "iat" => $issued_at,
48           "exp" => $expiration_time,
49           "iss" => $issuer,
50           "data" => array(
51               "id_pengguna" => $data3['id_pengguna'],
52               "nama" => $data3['nama']
53           )
54       );
55       // set response code
56       http_response_code(200);
57       // generate jwt
58       $jwt = JWT::encode($token,$key);
59       echo json_encode(
60           array(
61               "message" => "Login sukses",
62               "id_pengguna" => $data3['id_pengguna'],
63               "nama" => $data3['nama'],
64               "jwt" => $jwt
65           )
66       );
67   } else { // login gagal
68       // set response code
69       http_response_code(401);
70       echo json_encode(array("message" => "Login gagal"));
71   }
72 } elseif ($_SERVER['REQUEST_METHOD']=='POST')
73 { $jwt = $data->jwt;
74   $aksi = $data->aksi;
```

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

```
75 $id_barang = $data->id_barang;
76 $nama_barang = $data->nama_barang;
77
78 // decode jwt
79 try { JWT::decode($jwt,$key,array('HS256'));
80
81     if ($aksi == 'tambah')
82     { $data2=array( 'aksi' => $aksi,
83                   'id_barang' => $id_barang,
84                   'nama_barang' => $nama_barang
85                   );
86       $abc->tambah_data($data2);
87     } elseif ($aksi == 'ubah')
88     { $data2=array( 'aksi' => $aksi,
89                   'id_barang' => $id_barang,
90                   'nama_barang' => $nama_barang
91                   );
92       $abc->ubah_data($data2);
93     } elseif ($aksi == 'hapus')
94     { $data2=array( 'aksi' => $aksi,
95                   'id_barang' => $id_barang
96                   );
97       $abc->hapus_data($id_barang);
98     }
99
100     // set response code
101     http_response_code(200);
102     echo json_encode($data2);
103
104     // jika decode gagal, berarti jwt tidak valid
105 } catch (Exception $e)
106 { // set response code
107   http_response_code(401);
108   echo json_encode(array(
109     "message" => "Access denied"
110   ));
111 }
112 } elseif ($ _SERVER['REQUEST_METHOD']=='GET')
113 { $jwt = $_GET['jwt'];
114   // decode jwt
115   try { JWT::decode($jwt,$key,array('HS256'));
116
117       if ( ($ _GET['aksi']=='tampil') and (isset($_GET['id_barang'])) )
118       { $id_barang = filter($_GET['id_barang']);
119         $data=$abc->tampil_data($id_barang);
120       } else //menampilkan semua data
121       { $data = $abc->tampil_semua_data();
122       }
123
124       // set response code
125       http_response_code(200);
126       echo json_encode($data);
127
128       // jika decode gagal, berarti jwt tidak valid
129 } catch (Exception $e)
130 { // set response code
131   http_response_code(401);
132   echo json_encode(array(
133     "message" => "Access denied"
134   ));
135 }
136 } else { // error jika tanpa jwt
137   // set response code
138   http_response_code(401);
139   echo json_encode(array("message" => "Access denied"));
140 }
141
142 unset($abc,$postdata,$data,$data2,$data3,$token,$key,$issued_at,$expiration_time,$issuer,$jwt,$id_barang,$nama_barang,$aksi,$e);
143 ?>
```

Gambar 3. Source code server.php di Debian Server

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

10.1.4 *Source code Client.php* di *Windows*

```
1 <?php
2 error_reporting(1); // error ditampilkan
3 class Client
4 { private $url;
5
6 // function yang pertama kali di-load saat class dipanggil
7 public function __construct($url)
8 { $this->url=$url;
9   unset($url);
10 }
11
12 // function untuk menghapus selain huruf dan angka
13 public function filter($data)
14 { $data = preg_replace('/[^a-zA-Z0-9]/','',$data);
15   return $data;
16   unset($data);
17 }
18
19 public function login($data)
20 { $data = '{ "id_pengguna":"' . $data['id_pengguna'] . '",
21           "pin":"' . $data['pin'] . '",
22           "aksi":"' . $data['aksi'] . '"
23         }';
24   $c = curl_init();
25   curl_setopt($c,CURLOPT_URL,$this->url);
26   curl_setopt($c,CURLOPT_RETURNTRANSFER,true);
27   curl_setopt($c,CURLOPT_POST,true);
28   curl_setopt($c,CURLOPT_POSTFIELDS,$data);
29   $response = curl_exec($c);
30   curl_close($c);
31   $data2 = json_decode($response);
32   return $data2;
33   unset($data,$data2,$c,$response);
34 }
35
36 public function tampil_semua_data($jwt)
37 { $client = curl_init($this->url."?jwt=".$jwt);
38   curl_setopt($client,CURLOPT_RETURNTRANSFER,1);
39   $response = curl_exec($client);
40   curl_close($client);
41   $data = json_decode($response);
42   // mengembalikan data
43   return $data;
44   // menghapus variabel dari memory
45   unset($jwt,$client,$response,$data);
46 }
47
48 public function tampil_data($data)
49 { $id_barang = $this->filter($data['id_barang']);
50   $client = curl_init($this->url."?aksi=tampil&id_barang=".$data['id_barang']."&jwt=".$data['jwt']);
51   curl_setopt($client,CURLOPT_RETURNTRANSFER,1);
52   $response = curl_exec($client);
53   curl_close($client);
54   $data = json_decode($response);
55   return $data;
56   unset($id_barang,$client,$response,$data);
57 }
58
59 public function tambah_data($data)
60 { $data = '{ "id_barang":"' . $data['id_barang'] . '",
61           "nama_barang":"' . $data['nama_barang'] . '",
62           "jwt":"' . $data['jwt'] . '",
63           "aksi":"' . $data['aksi'] . '"
64         }';
65   $c = curl_init();
66   curl_setopt($c,CURLOPT_URL,$this->url);
```

**PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG**

```
67     curl_setopt($c,CURLOPT_RETURNTRANSFER,true);
68     curl_setopt($c,CURLOPT_POST,true);
69     curl_setopt($c,CURLOPT_POSTFIELDS,$data);
70     $response = curl_exec($c);
71     curl_close($c);
72     unset($data,$c,$response);
73 }
74
75 public function ubah_data($data)
76 { $data = '{ "id_barang":"' . $data['id_barang'] . '",
77           "nama_barang":"' . $data['nama_barang'] . '",
78           "jwt":"' . $data['jwt'] . '",
79           "aksi":"' . $data['aksi'] . '"
80         }';
81     $c = curl_init();
82     curl_setopt($c,CURLOPT_URL,$this->url);
83     curl_setopt($c,CURLOPT_RETURNTRANSFER,true);
84     curl_setopt($c,CURLOPT_POST,true);
85     curl_setopt($c,CURLOPT_POSTFIELDS,$data);
86     $response = curl_exec($c);
87     curl_close($c);
88     unset($data,$c,$response);
89 }
90
91 public function hapus_data($data)
92 { $id_barang = $this->filter($data['id_barang']);
93   $data = '{ "id_barang":"' . $id_barang . '",
94           "jwt":"' . $data['jwt'] . '",
95           "aksi":"' . $data['aksi'] . '"
96         }';
97     $c = curl_init();
98     curl_setopt($c,CURLOPT_URL,$this->url);
99     curl_setopt($c,CURLOPT_RETURNTRANSFER,true);
100
101     curl_setopt($c,CURLOPT_POST,true);
102     curl_setopt($c,CURLOPT_POSTFIELDS,$data);
103     $response = curl_exec($c);
104     curl_close($c);
105     unset($id_barang,$data,$c,$response);
106 }
107
108 // function yang terakhir kali di-load saat class dipanggil
109 public function __destruct()
110 { // hapus variable dari memory
111   unset($this->url);
112 }
113
114 $url = 'http://192.168.56.2/jwt-toko/server/server.php';
115 // buat objek baru dari class Client
116 $abc = new Client($url);
117 ?>
```

Gambar 4. Source code Client.php di Windows

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

10.1.5 Source code proses.php di Windows

```
1 <?php
2 include "Client.php";
3
4 if ($_POST['aksi']=='login')
5 {   $data = array("id_pengguna"=>$_POST['id_pengguna'],
6                 "pin"=>$_POST['pin'],
7                 "aksi"=>$_POST['login']);
8     $data2 = $abc->login($data);
9     //echo "<pre>";
10    //print_r($data2); //cek $data2
11    //echo "</pre>";
12
13    if ($data2) {
14        setcookie('jwt',$data2->jwt,time()+3600); // hilang dalam 1 jam
15        setcookie('id_pengguna',$data2->id_pengguna,time()+3600);
16        setcookie('nama',$data2->nama,time()+3600);
17        header('location:index.php?page=daftar-data');
18    } else {
19        header('location:index.php?page=login');
20    }
21 } else if ($_POST['aksi']=='tambah')
22 {   $data = array("id_barang"=>$_POST['id_barang'],
23                 "nama_barang"=>$_POST['nama_barang'],
24                 "jwt"=>$_POST['jwt'],
25                 "aksi"=>$_POST['aksi']);
26     $abc->tambah_data($data);
27     header('location:index.php?page=daftar-data');
28 } else if ($_POST['aksi']=='ubah')
29 {   $data = array("id_barang"=>$_POST['id_barang'],
30                 "nama_barang"=>$_POST['nama_barang'],
31                 "jwt"=>$_POST['jwt'],
32                 "aksi"=>$_POST['aksi']);
33     $abc->ubah_data($data);
34     header('location:index.php?page=daftar-data');
35 } else if ($_GET['aksi']=='hapus')
36 {   $data = array("id_barang"=>$_GET['id_barang'],
37                 "jwt"=>$_GET['jwt'],
38                 "aksi"=>$_GET['aksi']);
39     $abc->hapus_data($data);
40     header('location:index.php?page=daftar-data');
41 } else if ($_GET['aksi']=='logout')
42 {   setcookie('jwt','',time()-3600);
43     setcookie('id_pengguna','',time()-3600);
44     setcookie('nama','',time()-3600);
45     header('location:index.php?page=login');
46 }
47 unset($abc,$data,$data2);
48 ?>
```

Gambar 5. Source code proses.php di Windows

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

10.1.6 *Source code index.php* di *Windows*

```
1 <?php
2 include "Client.php";
3 ?>
4 <!doctype html>
5 <html>
6 <head>
7     <title></title>
8 </head>
9 <body>
10 <?php if ($_COOKIE['jwt']) { ?>
11     <a href="?page=home">Home</a> | <a href="?page=tambah">Tambah Data</a> | <a href="?page=daftar-data">Data Server</a>
12     | <a href="proses.php?aksi=logout" onclick="return confirm('Apakah Anda ingin Logout?')">Logout</a>
13 <br/>
14 <?php echo '<strong>' . $_COOKIE['nama'] . ' (' . $_COOKIE['id_pengguna'] . ')</strong>'; ?>
15 <?php } else { ?> <a href="?page=home">Home</a> | <a href="?page=login">Login</a> <?php ?>
16 <br/><br/>
17 <fieldset>
18 <? if ($_GET['page']=='login' and !isset($_COOKIE['jwt'])) { ?>
19 <legend>Login</legend>
20 <form name="form" method="POST" action="proses.php">
21     <input type="hidden" name="aksi" value="login"/>
22     <label>ID Pengguna</label>
23     <input type="text" name="id_pengguna"/>
24 <br/>
25     <label>PIN</label>
26     <input type="password" name="pin"/>
27 <br/>
28     <button type="submit" name="login">Login</button>
29 </form>
30
31 <? } elseif ($_GET['page']=='tambah' and isset($_COOKIE['jwt'])) { ?>
32 <legend>Tambah Data</legend>
33 <form name="form" method="POST" action="proses.php">
34     <input type="hidden" name="aksi" value="tambah"/>
35     <input type="hidden" name="jwt" value="<?=$_COOKIE['jwt']?>"/>
36     <label>ID Barang</label>
37     <input type="text" name="id_barang"/>
38 <br/>
39     <label>Nama Barang</label>
40     <input type="text" name="nama_barang"/>
41 <br/>
42     <button type="submit" name="simpan">Simpan</button>
43 </form>
44
45 <? } elseif ($_GET['page']=='ubah' and isset($_COOKIE['jwt'])) {
46     $data = array("jwt"=>$_COOKIE['jwt'],
47                 "id_barang"=>$_GET['id_barang']);
48     $r = $abc->tampil_data($data);
49 ?>
50 <legend>Ubah Data</legend>
51 <form name="form" method="post" action="proses.php">
52     <input type="hidden" name="aksi" value="ubah"/>
53     <input type="hidden" name="id_barang" value="<?=$r->id_barang?" />
54     <input type="hidden" name="jwt" value="<?=$_COOKIE['jwt']?>"/>
55     <label>ID Barang</label>
56     <input type="text" value="<?=$r->id_barang?" disabled>
57 <br/>
58     <label>Nama Barang</label>
59     <input type="text" name="nama_barang" value="<?=$r->nama_barang?">
60 <br/>
61     <button type="submit" name="ubah">Ubah</button>
62 </form>
63
64 <? unset($data,$r,$abc);
65 } else if ($_GET['page']=='daftar-data' and isset($_COOKIE['jwt'])) {
```

PRAKTIKUM SISTEM TERDISTRIBUSI
JURUSAN TEKNIK INFORMATIKA
UIN MAULANA MALIK IBRAHIM MALANG

```

66 ?>
67 <legend>Daftar Data Server</legend>
68 <table border="1">
69 <tr><th width='5%'>No</th>
70 <th width='10%'>ID Barang</th>
71 <th width='75%'>Nama</th>
72 <th width='5%' colspan="2">Aksi</th>
73 </tr>
74 <? $no = 1;
75 $data = $abc->tampil_semua_data($_COOKIE['jwt']);
76 foreach ($data as $r) {
77 ?> <tr><td><?=$no?></td>
78 <td><?=$r->id_barang?></td>
79 <td><?=$r->nama_barang?></td>
80 <td><a href="?page=ubah&id_barang=<?=$r->id_barang?&jwt=<?=$_COOKIE['jwt']?>">Ubah</a></td>
81 <td><a href="proses.php?aksi=hapus&id_barang=<?=$r->id_barang?&jwt=<?=$_COOKIE['jwt']?>" onclick="return confirm('
    Apakah Anda ingin menghapus data ini?')">Hapus</a></td>
82 </tr>
83 <? $no++;
84 }
85 unset($no,$data,$r,$abc);
86 ?>
87 </table>
88
89 <? } else { ?>
90 <legend>Home</legend>
91 Aplikasi sederhana ini menggunakan JWT dan RESTful dengan format data JSON (JavaScript Object Notation).
92 </fieldset>
93 <? } ?>
94 </body>
95 </html>

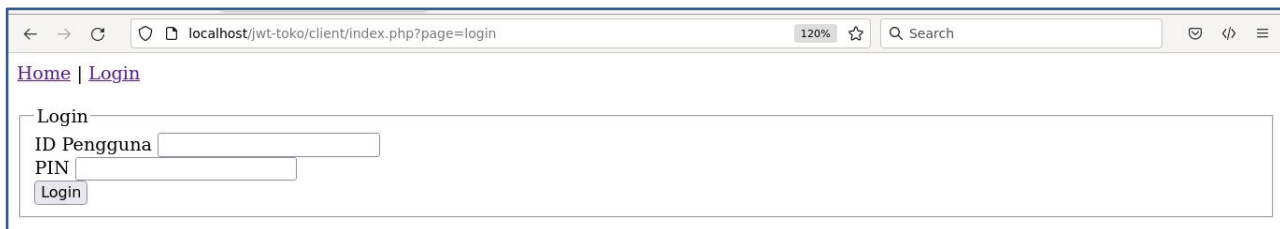
```

Gambar 6. Source code *index.php* di *Windows*

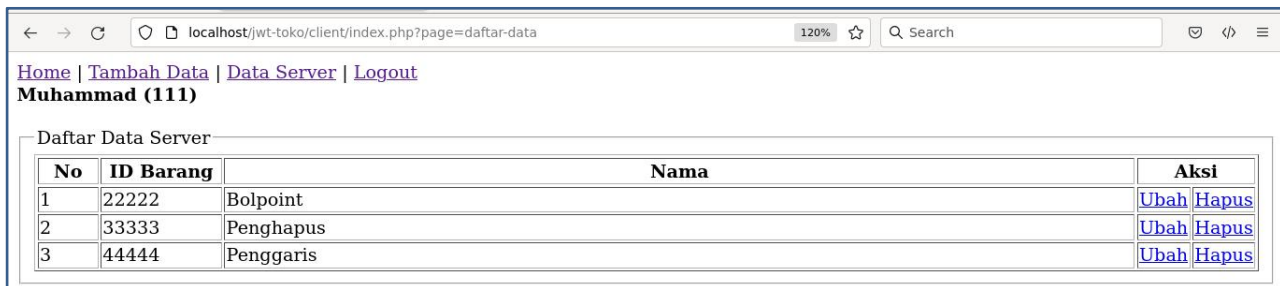
Akses melalui *web browser* <http://localhost/jwt-toko/client/index.php> di *Windows*. Coba jalankan aplikasi sederhana ini dengan *login* pengguna, menambah, menampilkan, mengubah dan menghapus datanya. Data akan dikirim dari *Windows* ke *Debian Server* dan data hanya disimpan di *database Debian Server*. Data yang tersimpan tersebut akan ditampilkan ke *Client* di *Windows*.



Gambar 7. Laman Utama *Client* sebelum pengguna *login*



Gambar 8. Laman *Login* pada *Client*



Gambar 9. Laman *Client* setelah pengguna *login*