

Laporan Praktikum

Sinkronisasi Proses/Thread

Interprocess Communication



Nama : Josep Phyto Napitupulu

NIM : 11421039

Program Studi: DIV TEKNOLOGI REKAYASA
PERANGKAT LUNAK

INSTITUT TEKNOLOGI DEL
FAKULTAS VOKASI

A. Pemograman

A.1. Socket

1. Kode program

Input

```
// Nama : Josep Phyto Napitupulu
// NIM : 11421039
// Kelas : TRPL 2

import java.net.*;
import java.io.*;

public class LowPortScanner{

    public static void main(String[] args){

        String host = "localhost";

        if(args.length > 0){
            host = args[0];
        }

        for (int i = 1; i < 1024; i++){
            try{
                Socket socket = new Socket(host, i);
                System.out.println("There is a server on port" + i + "of" + host);
            }
            catch(UnknownHostException ex){
                System.err.println(ex);
                break;
            }
            catch(IOException ex){
                System.err.println(ex);
                break;
            }
        }
    }
}
```

Output

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

E:\Kuliah\SEMESTER 2\Praktikum\Sisop\Week 6>java LowPortScanner.java
java.net.ConnectException: Connection refused: connect

E:\Kuliah\SEMESTER 2\Praktikum\Sisop\Week 6>java IPFinder.class
```

2. Dari kode program di atas jelaskan apa yang dimaksud dengan:

a. `Import java.net.*;`

Jawab :

Merupakan perintah Program Java berisi kelas-kelas yang memperbolehkan program berkomunikasi melewati jaringan. Digunakan dalam Membuat suatu Komunikasi Jaringan dalam sebuah program yang dibuat dengan bahasa pemrograman Java.

b. `Import java.io.*;`

Jawab :

Pengertian IO sendiri merupakan singkatan dari Input Output.

Input merupakan perintah untuk memasukan data dari peralatan input sebagai contoh keyboar, sedangkan output merupakan perintah atau berfungsi menampilkan data kepada pengguna.

c. `Socket socket = new Socket (host, i);`

Jawab :

Berfungsi untuk membuka server socket, dimana socket merupakan sebuah interface untuk pemograman jaringan yang digunakan pada transport layer. Penggunaan socket mirip dengan pemograman file I/O.

d. `UnknownHostException`

Jawab :

memanggil nama komputer dengan parameter variable host yang berisi alamat IP.

e. `IOException`

Jawab :

Suatu Method yang Membaca Input Data String.

3. Kode program

Input

```
// Nama : Josep Phyto Napitupulu
// NIM : 11421039
// Kelas : TRPL 2

import java.net.*;
import java.util.*;

public class IPFinder{

    public static void main(String[] args){

        String host;
        Scanner input = new Scanner(System.in);

        System.out.print("\n\nEnter the hostname: ");
        host = input.next();

        try{
            InetAddress address = InetAddress.getByName(host);
            System.out.println("IP Address: " + address.toString());
        }catch(UnknownHostException ex){
            System.out.println("Can not find host");
        }
    }
}
```

Output

```
C:\Windows\System32\cmd.exe - java IPFinder.java
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

E:\Kuliah\SEMESTER 2\Praktikum\Sisop\Week 6>java LowPortScanner.java
java.net.ConnectException: Connection refused: connect

E:\Kuliah\SEMESTER 2\Praktikum\Sisop\Week 6>java IPFinder.java

Enter the hostname:
```

4. Dari kode program di atas jelaskan apa yang dimaksud dengan:

a. InetAddress

Jawab :

Sintaks berikut digunakan untuk menerima nilai ip address dari localhost.

b. Method getByName()

Jawab :

Metode getByName digunakan untuk mendapatkan IP address dari suatu host Bernama nama_host.

5. Modifikasilah program pada no.1 untuk menampilkan IP Address lokal komputer anda. Contoh keluaran yang diharapkan seperti gambar di bawah.

TCP Socket

1. Code program.

Input

```
// Nama : Josep Phyto Napitupulu
// NIM : 11421039
// Kelas : TRPL 2

import java.net.*;
import java.io.*;
import java.util.*;

public class TCPEchoServer{

    private static ServerSocket servSocket;
    private static final int PORT = 1234;

    public static void main(String[] args){

        try{
            servSocket = new ServerSocket(PORT);
        }

        catch(IOException ioEx){
            System.out.println("Unable to attach to port!");
            System.exit(1);
        }

        do{
            handleClient();
        }while(true);
    }

    private static void handleClient(){
        Socket link = null;

        try{

            link = servSocket.accept();

            Scanner input = new Scanner(link.getInputStream());
            PrintWriter output = new PrintWriter(link.getOutputStream(), true);

            int numMessages = 0;
            String message = input.nextLine();

            while(!message.equals("CLOSE")){
                System.out.println("Message received.");
                numMessages++;
                output.println("Message " + numMessages + ": " + message);
                message = input.nextLine();
            }
        }
    }
}
```

```

47         message = input.nextLine();
48     }
49     output.println(numMessages + " Messages received.");
50 }
51
52 catch(IOException ioEx){
53     ioEx.printStackTrace();
54 }
55 finally{
56
57     try{
58         System.out.println("Closing Connection...");
59         link.close();
60     }
61
62     catch(IOException ioEx){
63         System.out.println("Unable to disconnect");
64         System.exit(1);
65     }
66 }
67 }
68 }
69

```

Output

```

C:\Windows\System32\cmd.exe - java TCPEchoServer.java
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

E:\Kuliah\SEMESTER 2\Praktikum\Sisop\Week 6>java LowPortScanner.java
java.net.ConnectException: Connection refused: connect

E:\Kuliah\SEMESTER 2\Praktikum\Sisop\Week 6>java IPfinder.java

Enter the hostname: msi
IP Address: msi/192.168.56.1

E:\Kuliah\SEMESTER 2\Praktikum\Sisop\Week 6>java TCPEchoServer.java

```

2. Dari kode program di atas, jelaskan langkah-langkah untuk membentuk server.

Jawab :

Langkah-langkah untuk membentuk server yaitu :

1. Mengetikkan kode program dari TCPEchoServer
2. Memanggil java.net, java.io, dan java.util tersebut
3. Selanjutnya, buat sebuah statement handleClient dan melakukan query dalam statement handleClient.
4. Dari melakukan query sebelumnya, dan menghasilkan sebuah proses hasil nya serta menutup koneksi di dalam finally nya.

3. Kode program client

Input

```
> Kuliah > SEMESTER 2 > Praktikum > Sisop > Week 6 > TCPEchoClient.java
1 // Nama : Josep Phyto Napitupulu
2 // NIM : 11421039
3 // Kelas : TRPL 2
4
5 import java.net.*;
6 import java.io.*;
7 import java.util.*;
8
9 public class TCPEchoClient{
10     private static InetAddress host;
11     private static final int PORT = 1234;
12
13     public static void main(String[] args){
14
15         try{
16             host = InetAddress.getLocalHost();
17         }
18
19         catch(UnknownHostException unEx){
20             System.out.println("Host ID not found");
21             System.exit(1);
22         }
23
24         accessServer();
25     }
26
27     private static void accessServer(){
28         Socket link = null;
29
30         try{
31             link = new Socket(host, PORT);
32
33             Scanner input = new Scanner(link.getInputStream());
34             PrintWriter output = new PrintWriter(link.getOutputStream(), true);
35             Scanner userEntry = new Scanner(System.in);
36
37             String message, response;
38             do{
39                 System.out.print("Enter Message: ");
40                 message = userEntry.nextLine();
41                 output.println(message);
42                 response = input.nextLine();
43                 System.out.println("\nSERVER> " + response);
44             }while(!message.equals("CLOSE"));
45         }
46
47         catch(IOException ioEx){
48             ioEx.printStackTrace();
49         }
50         finally{
51             try{
52                 System.out.println("closing connection..");
53                 link.close();
54             }
55
56             catch(IOException ioEx){
57                 System.out.println("Unable to disconnect");
58                 System.exit(1);
59             }
60         }
61     }
62 }
```

Output

```
C:\Windows\System32\cmd.exe - java TCPEchoClient.java
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

E:\Kuliah\SEMESTER 2\Praktikum\Sisop\Week 6>java TCPEchoClient.java
Enter Message:
[
[
```

4. Dari kode program di halaman sebelumnya, jelaskan langkah-langkah untuk membentuk client.

Jawab :

Langkah-langkah membentuk client yaitu :

5. Buatlah koneksi ke server nya tersebut. Buat obyek Socketnya. Tentukan lah IP Address atau Hostname mesin server dan nomor port yang dimana aplikasi server tersebut berjalan. 2
6. 2. Socket link = new Socket(InetAddress,No_Port); Socket link = new Socket(InetAddress.getLocalHost(),1234);. Set up stream input dan output. Panggil metode `getInputStream()` & `getOutputStream()` dari obyek Socket nya. Scanner input = new Scanner(link.getInputStream()); PrintWriter output = new PrintWriter(link.getOutputStream(),true);
7. Kirim dan terima data. Obyek Scanner akan menerima data dari server (input), obyek PrintWriter juga akan mengirimkan data nya ke server (output). `output.println(message); response = input.nextLine();`
8. Selanjutnya, tutup koneksi dan panggil metode `close()` dari Obyek Socket `link.close();` Lalu liat di bagian cmd dari TCPEchoServer tersebut.
9. Eksekusilah kedua program client dan server kemudian cetaklah hasil yang Anda dapatkan

Client

```
C:\Windows\System32\cmd.exe - java TCPEchoClient.java
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

E:\Kuliah\SEMESTER 2\Praktikum\Sisop\Week 6>java TCPEchoClient.java
Enter Message: Josep Phyto Napitupulu

SERVER> Message 1: Josep Phyto Napitupulu
Enter Message:
```


Server

```
C:\Windows\System32\cmd.exe - java TCPEchoServer.java
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

E:\Kuliah\SEMESTER 2\Praktikum\Sisop\Week 6>java LowPortScanner.java
java.net.ConnectException: Connection refused: connect

E:\Kuliah\SEMESTER 2\Praktikum\Sisop\Week 6>java IPFinder.java

Enter the hostname: msi
IP Address: msi/192.168.56.1

E:\Kuliah\SEMESTER 2\Praktikum\Sisop\Week 6>java TCPEchoServer.java
Message received.
```

B. Tugas Teori

Jelaskanlah ketiga tipe Socket berikut:

- a. Connection Oriented (TCP)

Jawab :

TCP adalah suatu protokol yang berada di lapisan transport (baik itu dalam tujuh lapis model referensi OSI atau model DARPA) yang berorientasi sambungan (Connection-oriented) dan dapat diandalkan (reliable). TCP bertugas untuk pengiriman data dari sumber ke tujuan dengan benar, bukan hanya itu TCP juga dapat mendeteksi kesalahan atau hilangnya data dan melakukan pengiriman kembali sampai data diterima dengan lengkap.

- b. Connectionless (UDP)

Jawab :

UDP merupakan singkatan dari User Datagram Protocol, merupakan salah satu protokol lapisan transport TCP/IP yang mendukung komunikasi yang tidak andal (unreliable), tanpa koneksi (connectionless) selang host-host dalam jaringan yang menggunakan TCP. Connectionless merupakan salah satu karakteristik UDP yang mana pesan-pesan UDP akan dikirim tanpa harus diterapkan bagian negosiasi koneksi selang dua host yang ingin berubah informasi.

- c. MulticastSocket

Jawab :

MulticastSocket bermanfaat untuk aplikasi yang menerapkan komunikasi secara berkelompok. Selain itu MulticastSocket berfungsi untuk mengirim pesan kepada grup.

C. Tugas Pemograman

1. Tuliskan langkah-langkah untuk mengimplementasikan server dan client dengan menggunakan UDP Socket.

Jawab :

- Langkah-langkah untuk mengimplementasikan server menggunakan UDP Socket yaitu:
 1. Buatlah class dengan nama JavaSenderUDP.
 2. Mengkonversi data yang mau dikirim ke dalam bentuk array byte.
 3. Memasukkan array byte tadi, panjang data dalam array, dan InetAddress serta nomor port tujuan ke dalam DatagramPacket lewat konstruktornya.
 4. Buatlah DatagramSocket menggunakan konstruktornya (client DatagramSocket), dan memanggil method send() dengan memasukkan DatagramPacket yang telah dibuat.
 5. Setelah itu, jalankan program nya.
- Langkah-langkah untuk mengimplementasikan client menggunakan UDP Socket yaitu:
 1. Buatlah sebuah class dengan nama JavaReceiveUDP.
 2. Buatlah sebuah DatagramPacket kosong (DatagramPacket untuk menerima data).
 3. Buatlah objek DatagramSocket melalui konstruktornya (DatagramSocket untuk server).
 4. Lalu panggil method receive() dari datagramSocket dengan memasukkan DatagramPacket kosong yang telah dibuat tadi untuk menampung datagram yang datang.ESS/SISOP/2 3-3- 9.
 5. Ambil datanya dalam bentuk array byte pada DatagramPacket yang telah terisi. Setelah itu, konversikan ke bentuk data yang diinginkan.

2. Buatlah sebuah program client-server yang mengimplementasikan UDP Sockets. Server diberi nama UDPEchoServer.java, sedangkan client diberinama UDPEchoClient.java.

Jawab :

Input Server

```
// Nama      : Josep Phyto Napitupulu
// NIM       : 11421039
// Kelas     : TRPL 2

import java.net.*;
import java.io.*;

public class UDPEchoServer{

    public static void main(String args[]){

        int port = 8000;
        DatagramSocket serverDatagramSocket = null;

        try{
            serverDatagramSocket = new DatagramSocket(port);
            System.out.println("Create UDP Echo Server on port "+port);
        }
        catch(IOException e){
            System.out.println(e);
            System.exit(1);
        }try{
            byte buffer[] = new byte[1024];
            DatagramPacket datagramPacket = new
            DatagramPacket (buffer, buffer.length);
            String input;
            while(true){
                serverDatagramSocket.receive(datagramPacket);
                input = new String(datagramPacket.getData(), 0, datagramPacket.getLength());
                System.out.println("Received from server: "+input);
                serverDatagramSocket.send(datagramPacket);
            }
        }
        catch(IOException e){
            System.out.println(e);
        }
    }
}
```

Input Client

```
// Nama      : Josep Phyto Napitupulu
// NIM       : 11421039
// Kelas     : TRPL 2

import java.net.*;
import java.io.*;

public class UDPEchoClient{

    public static class UDPEchoReader extends Thread{

        public UDPEchoReader(DatagramSocket socket){

            datagramSocket = socket;
            active = true;
        }

        public void run(){
            byte[] buffer = new byte[1024];
            DatagramPacket incoming = new DatagramPacket(buffer, buffer.length);
            String receivedString;
            while(active){
                try{
                    datagramSocket.receive(incoming);
                    receivedString = new String(incoming.getData(), 0, incoming.getLength());
                    System.out.println("Received from server: "+receivedString);
                }
                catch(IOException e){
                    System.out.println(e);
                    active = false;
                }
            }
        }

        public boolean active;
        public DatagramSocket datagramSocket;
    }

    public static void main(String[] args){
        InetAddress address = null;
        int port = 8000;
        DatagramSocket datagramSocket = null;
        BufferedReader keyboardReader = null;

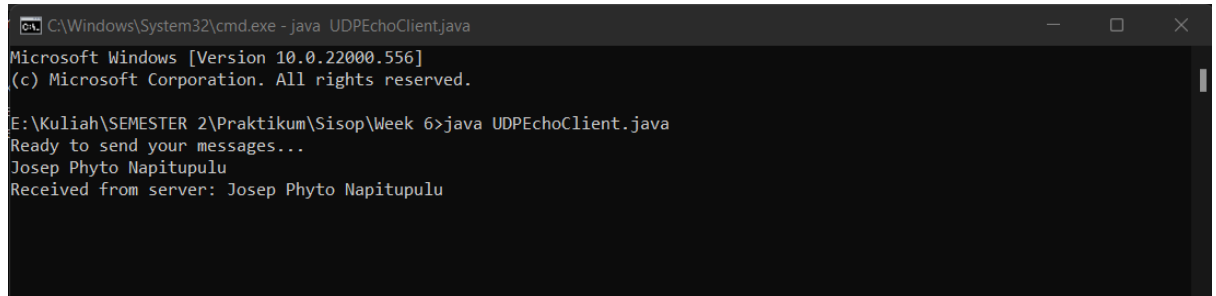
        try{
            address = InetAddress.getByName("127.0.0.1");
            datagramSocket = new DatagramSocket();
            keyboardReader = new BufferedReader(new InputStreamReader(System.in));
        }
        catch(IOException e){
            System.out.println(e);
            System.exit(1);
        }
    }
}

45 keyboardReader = new BufferedReader(new InputStreamReader(System.in));
46 }
47 catch(IOException e){
48     System.out.println(e);
49     System.exit(1);
50 }
51 UDPEchoReader reader = new UDPEchoReader(datagramSocket);
52 reader.setDaemon(true);
53 reader.start();
54 System.out.println("Ready to send your messages...");
55
56 try{
57     String input;
58     while(true){
59         input = keyboardReader.readLine();
60         DatagramPacket datagramPacket = new DatagramPacket(input.getBytes(), input.length(), address, port);
61         datagramSocket.send(datagramPacket);
62     }
63 }
64 catch(IOException e){
65     System.out.println(e);
66 }
67 }
68 }
```

3. Tampilkan hasil eksekusi dari program UDPSocket yang telah anda kerjakan.

Jawab :

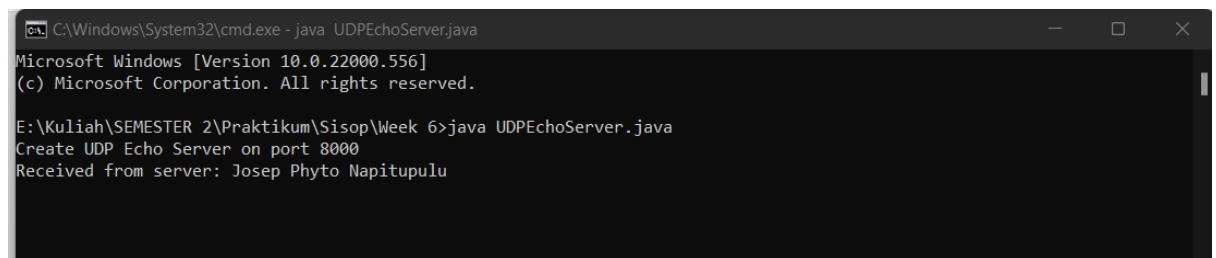
Output Client



```
C:\Windows\System32\cmd.exe - java UDPEchoClient.java
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

E:\Kuliah\SEMESTER 2\Praktikum\Sisop\Week 6>java UDPEchoClient.java
Ready to send your messages...
Josep Phyto Napitupulu
Received from server: Josep Phyto Napitupulu
```

Output Server



```
C:\Windows\System32\cmd.exe - java UDPEchoServer.java
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

E:\Kuliah\SEMESTER 2\Praktikum\Sisop\Week 6>java UDPEchoServer.java
Create UDP Echo Server on port 8000
Received from server: Josep Phyto Napitupulu
```