



## MODUL PRAKTIKUM

# Jaringan

Versi 1.3

## Modul Praktikum Jaringan

### 1. Tujuan

- Mengerti konsep dasar jaringan
  - IP address
  - protokol
  - ports
  - client/server
  - socket
- Membuat aplikasi menggunakan paket jaringan Java
  - ↳ *ServerSocket*
  - ↳ *Socket*
  - ↳ *MulticastSocket*
  - ↳ *DatagramPacket*

### 2. Latar Belakang

Java memungkinkan anda untuk mempermudah mengembangkan aplikasi yang mengerjakan berbagai pekerjaan melalui jaringan. Ini adalah suatu cita-cita pembuatan Java yang menjadi salah satu kekuatan Java sejak dibuat dengan internet di dalam cita-cita. Sebelum mempelajari tentang jaringan dalam Java. Pertama-tama anda akan diperkenalkan kepada beberapa konsep dasar jaringan.

Jika sebelumnya anda suda mengetahui, internet adalah jaringan global dengan berbagai jenis komputer yang berbeda yang tersambung dalam berbagai jalan. Walaupun terdapat perbedaan dalam software dan hardware yang disambung bersama, hal tersebut sangatlah bagus dimana internet masih berfungsi. Hal ini memungkinkan karena standar komunikasi memiliki ketetapan dan juga keselarasan. Standar ini menjamin kesesuaian dan kekuatan komunikasi diantara luasnya sistem pada internet.

### 3. Percobaan

#### Percobaan 1 Koneksi Server:

```
import java.net.*;

import java.io.*;

public class Networking01 {
```

## Modul Praktikum Jaringan

```
public static void main(String [] args) {  
    ServerSocket server = null;  
    Socket client;  
    try {  
        server = new ServerSocket(1234);  
        //1234 is an unused port number  
    } catch (IOException ie) {  
        System.out.println("Cannot open socket.");  
        System.exit(1);  
    }  
    while(true) {  
        try {  
            client = server.accept();  
            OutputStream clientOut = client.getOutputStream();  
            PrintWriter pw = new PrintWriter(clientOut, true);  
            InputStream clientIn = client.getInputStream();  
            BufferedReader br = new BufferedReader(new  
                InputStreamReader(clientIn));  
            pw.println(br.readLine());  
        } catch (IOException ie) {  
        }  
    }  
}
```

## Modul Praktikum Jaringan

### Percobaan 2 Koneksi Client:

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

public class Networking02 {
    public static void main(String args[]) {
        try {
            //Socket client = new Socket("133.0.0.1", 1234);
            Socket client = new Socket(InetAddress.getLocalHost(),
            1234);

            InputStream clientIn = client.getInputStream();
            OutputStream clientOut = client.getOutputStream();
            PrintWriter pw = new PrintWriter(clientOut, true);
            BufferedReader br = new BufferedReader(new
            InputStreamReader(clientIn));
            BufferedReader stdIn = new BufferedReader(new
            InputStreamReader(System.in));

            System.out.println("Type a message for the server: ");
            pw.println(stdIn.readLine());

            System.out.println("Server message: ");
            System.out.println(br.readLine());

            pw.close();
            br.close();
            client.close();

        } catch (ConnectException ce) {
            System.out.println("Cannot connect to the server.");
        } catch (IOException ie) {
```

## Modul Praktikum Jaringan

```
System.out.println("I/O Error.");
```

```
}
```

```
}
```

```
}
```

## 4. Latihan

### 4.1 Trivia Server

Buatlah sebuah server yang berisi satu set pertanyaan yang mudah. Secara sederhana, akan ada sekitar 5-10 pertanyaan.

Client yang terhubung ke server mengirim sebuah permintaan untuk sebuah pertanyaan atau jawaban sebuah pertanyaan, Client mengirim pesan "permintaan". Untuk jawaban dari sebuah pertanyaan, client mengirim pesan "jawaban". Ketika menerima pesan "permintaan", secara acak server akan memilih satu pertanyaan dari koleksi yang ada. Dia mengirimkan pertanyaan yang dipilih sesuai dengan nomor yang bersangkutan kepada client.

Ketika server menerima pesan "jawaban" dari client, dia menginformasikan user bahwa user perlu mengirimkan jawaban sesuai dengan nomor pertanyaan kepada server. Jawaban itu harus dalam format <no pertanyaan>#<jawaban Anda>.

Berikut adalah contoh skenario :

Client: "permintaan"

Server: "3#Siapa pembuat Java?"

Client: "jawaban"

Server: "Berikan jawabanmu dengan format: <nomor pertanyaan>#<jawaban Anda>"

Client: "3#James Gosling"

Server: Kerja yang bagus!

...



> > > Java Education Network Indonesia

## Modul Praktikum Jaringan