

PEMOGRAMAN SOCKET DI JAVA

Rachmat Selamat

Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI
Jl. Ir. H. Juanda 96 Bandung 40132

E-mail : rachmatselametskom@gmail.com

ABSTRAK

Perkembangan jaringan sudah sangat pesat yang ditandai dengan adanya internet. Java versi JDK menyediakan fasilitas untuk pemograman jaringan dengan beberapa metode, salah satunya adalah socket. Fasilitas jaringan tersebut tersedia dalam paket net. Untuk membuat program menggunakan socket, program dipecah menjadi 2 bagian, yaitu untuk server dan untuk client.

Kata-kata kunci : socket, Java, jaringan.

1 PENDAHULUAN

Sistem jaringan dalam dunia komputer sudah dikenal sejak lama. Dahulu transfer data hanya dikenal untuk 1 ruangan atau yang disebut dengan LAN(Local Area Network). Kemudian sistem ini berkembang pesat hingga dapat mentransfer data antar dunia atau dikenal dengan nama internet.

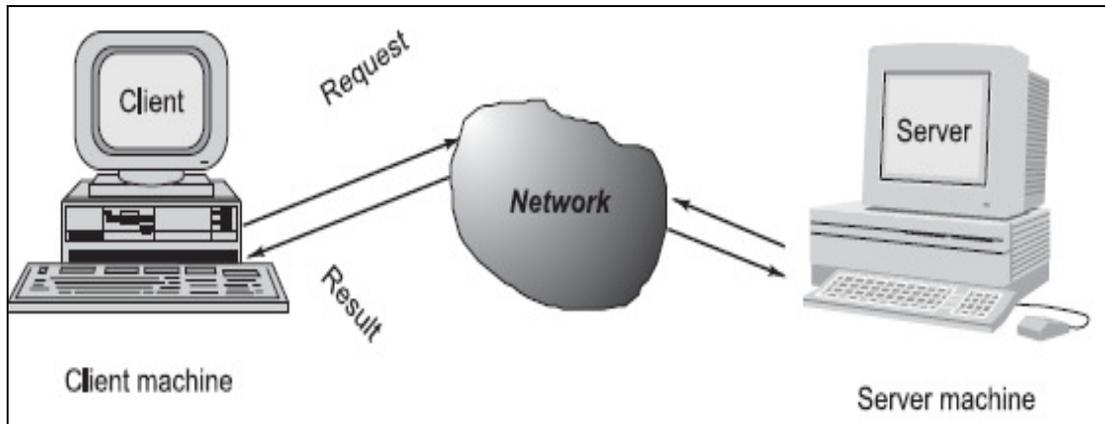
Perkembangan teknologi jaringan ini disediakan oleh pemograman java dalam versi JDK (*Java Development Kit*). Semua kelas yang dapat digunakan terbungkus dalam paket net. Semua fungsi-fungsi umum yang digunakan dalam jaringan terdapat dalam paket ini.

Dalam paper ini akan dibahas bagaimana cara menggunakan kelas yang berhubungan dengan socket. Program yang dibuat terdiri dari 2 bagian, yaitu untuk server dan untuk client.

2 SISTEM JARINGAN DI JAVA

Sistem jaringan secara umum memiliki sebuah server yang terhubung dengan beberapa client. Client akan melakukan request ke server, kemudian server memproses

request tersebut dan akan mengirimkan hasilnya ke client. Proses ini dapat dilihat pada Gambar 1.



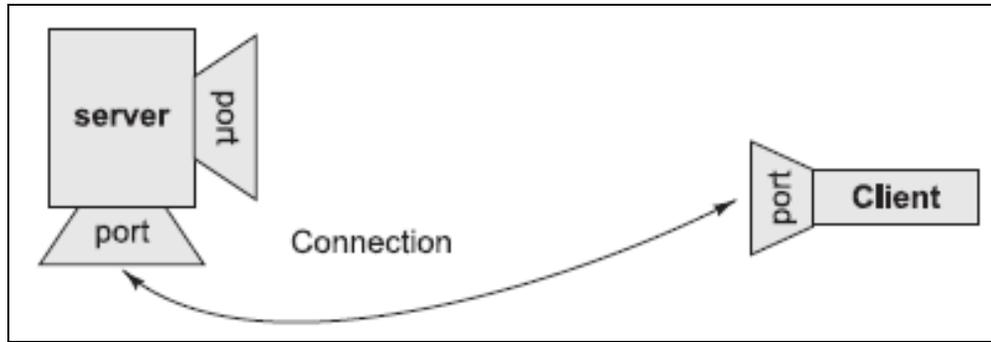
Gambar 1
Gambaran proses Client-Server

Proses yang dikerjakan di server akan dikerjakan oleh program yang berada di mesin server. Proses ini melibatkan juga service yang sudah disediakan oleh transport layer dalam jaringan, yang salah satu bagiannya disebut TCP/IP. Bagian ini merupakan protocol dalam jaringan yang menghubungkan antar client dan server. Metode yang banyak digunakan untuk mengatur protocol ini adalah socket.

Socket adalah sebuah interface untuk pemrograman jaringan yang digunakan pada transport layer. Penggunaan socket mirip dengan pemrograman file I/O. Metode stream yang digunakan pada file I/O juga dapat diterapkan pada pemrograman socket. Pemrograman server membutuhkan sebuah socket yang terikat dengan suatu port. Server akan menunggu (listen) melalui socket sampai sebuah client melakukan request (gambar 2.2). Jika berhasil, maka server akan menerima koneksi dengan mengambil socket yang mengikat port yang lain sehingga dapat melanjutkan penungguan request dari client sambil mengerjakan proses. Proses ini dapat dilihat pada Gambar 2.

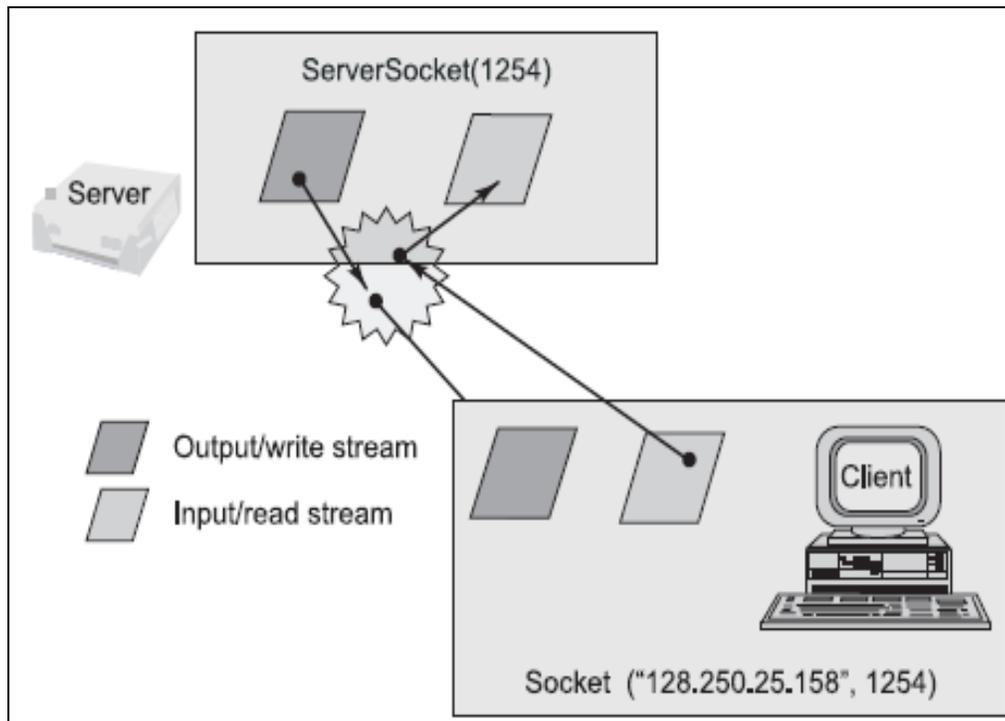


Gambar 2
Proses server menunggu request dari client



Gambar 3
Proses server menunggu dan mengerjakan request

Dalam paket net, ada 2 kelas yang digunakan untuk pemrograman socket, yaitu `ServerSocket` dan `Socket`. Program server digunakan untuk menunggu request dari client. Apabila request sedang dikerjakan, program akan membuat socket baru untuk menukar data dengan client menggunakan input dan output stream. Konsepnya hampir sama dengan file I/O. Cara kerjanya dapat dilihat pada gambar 2.4. Penulisan IP address (128.250.25.158) pada socket dapat diganti dengan nama host seperti `likmi.co.id`. Sedangkan 1254 adalah nomor port yang digunakan untuk menghubungkan client dan server.



Gambar 4
Cara kerja pemrograman socket

3 PEMOGRAMAN SOCKET SEBAGAI SERVER

Algoritma untuk pemograman server menggunakan socket :

- a. Buka server socket, bentuk penulisannya :

```
ServerSocket server = new ServerSocket(PORT);
```

- b. Menunggu request dari client, bentuk penulisannya :

```
Socket client = server.accept();
```

- c. Membuat I/O stream untuk komunikasi dengan client, bentuk penulisannya :

```
DataInputStream is = new DataInputStream(client.getInputStream());
```

```
DataOutputStream os = new DataOutputStream(client.getOutputStream());
```

- d. Mengerjakan komunikasi dengan client

```
String line=is.readLine();
```

```
Os.writeByte("Halo\n");
```

- e. Menutup socket

```
client.close();
```

Contoh lengkap SimpleServer.java:

```
import java.net.*;
import java.io.*;
public class SimpleServer {
    public static void main(String args[]) throws IOException {

        ServerSocket s = new ServerSocket(1254);
        Socket s1=s.accept();

        OutputStream s1out = s1.getOutputStream();
        DataOutputStream dos = new DataOutputStream (s1out);

        dos.writeUTF("Halo apa kabar");

        dos.close();
        s1out.close();
        s1.close();
    }
}
```

Pada program SimpleServer, server akan meunggu client masuk/request. Apabila ada client masuk, maka akan dikirim pesan "Halo apa kabar".

4 PEMOGRAMAN SOCKET SEBAGAI CLIENT

Algoritma untuk pemograman client menggunakan socket :

- a. Membuat object socket, bentuk penulisannya :

```
Socket client = new Socket(server, PORT);
```

- b. Membuat I/O stream untuk komunikasi ke server, bentuk penulisannya:

```
DataInputStream is = new DataInputStream(client.getInputStream());
```

```
DataOutputStream os = new DataOutputStream(client.getOutputStream());
```

- c. Mengerjakan komunikasi dengan server

```
String line=is.readLine();
```

```
Os.writeByte("Halo\n");
```

- d. Menutup socket

```
client.close();
```

Contoh lengkap SimpleClient.java:

```
import java.net.*;
import java.io.*;
public class SimpleClient {
    public static void main(String args[]) throws IOException {

        Socket s1 = new Socket("localhost",1254);

        InputStream s1In = s1.getInputStream();
        DataInputStream dis = new DataInputStream(s1In);
        String st = new String (dis.readUTF());
        System.out.println(st);

        dis.close();
        s1In.close();
        s1.close();
    }
}
```

Pada program SimpleClient, client akan membuat koneksi dan langsung komunikasi dengan server. Setelah terhubung, maka akan muncul pesan "Halo apa kabar", yang merupakan hasil proses dari server.

5 KESIMPULAN

Kesimpulan yang dapat ditarik dari pemograman socket ini adalah sebagai berikut:

- a. Dengan menggunakan java versi JDK, membuat program untuk jaringan menjadi lebih mudah karena sudah tersedia kelas-kelas yang dibutuhkan di dalam paket net.
- b. Program untuk jaringan terpecah menjadi 2 bagian, yaitu program server dan program client.
- c. Cara kerja pemograman socket mirip dengan konsep file I/O.

DAFTAR PUSTAKA

- [1] <https://docs.oracle.com>.
- [2] <http://www.javatpoint.com>.
- [3] <http://beginnersbook.com>.
- [4] <http://www.tutorialspoint.com>.
- [5] Buyya, R, Selvi, S. T., Chu X. Object-Oriented Programming with Java Essentials and Applications. 2007. McGraw Hills.