

LAPORAN PRAKTIKUM

953633205  
  
PRAKTIKUM SISTEM TERDISTRIBUSI

MODUL: 2

“PEMROGRAMAN SOCKET DENGAN PAKET UDP”

Eska Smara Nofiansi M0516018

Fabian Gilanggi M0517014

Henny Nurcahyaning Tyas M0517022

KELOMPOK : 6

HARI : JUMAT

TANGGAL : 3 APRIL 2020

WAKTU : 12.45 – 15.00 WIB

ASISTEN : HAIDAR HENDRI SETYAWAN M0516023

TAUFIQ ODHI DWI PUTRA M0516043

PROGRAM STUDI INFORMATIKA

UNIVERSITAS SEBELAS MARET

2020

Modul 2

PEMROGRAMAN SOCKET DENGAN PAKET UDP

Eska Smara Nofiansi M0516018, Fabian Gilanggi M0517014, Henny Nurcahyaning Tyas M0517022/ Kelompok 6/ Jumat, 3 April 2020

[fgilanggi@student.uns.ac.id,](mailto:fgilanggi@student.uns.ac.id,) [hnytyas72@student.uns.ac.id,](mailto:hnytyas72@student.uns.ac.id,) [rarasmara@student.uns.ac.id](mailto:rarasmara@student.uns.ac.id)

Asisten : Haidar Hendri Setyawan

**Taufiq Odhi Dwi Putra**

***Abstraksi – Pengertian socket adalah interface pada jaringan yang menjadi titik komunikasi antarmesin pada Internet Protocol. Jika tidak terjadi komunikasi maka tidak ada pertukaran data dan informasi jaringan. Oleh karena itu dilakukan pemrograman untuk menggunakan mengatur lalu lintas data pada socket tersebut. Pada perccobaa ini, dibuat dua aplikasi socket sederhana yaitu aplikasi open socket untuk server yang akan menerima dan mengirimkan output ke client dan aplikasi open socket untuk client. Komunikasi yang dilakukan menggunakan model paket UDP.***

***Kata Kunci - Socket, UDP, Komunikasi***

***Client-Server.***

# Pendahuluan

Socket adalah sebuah abstaksi perangkat lunak yang digunakan suatu “terminal” dari suatu hubugan antara dua mesin atau proses yang saling berinterkoneksi. Socket dapat melakukan beberapa operasi yang meliputi koneksi ke main remote, mengirim data (Write), menerima data (Read), menutup koneksi (Close), bind to port, listen pada data yang masuk dan menerima koneksi dari mesin remote pada port tertentu

Dalam laporan ini kami membuat socket programming client server untuk proses penjumlahan, suatu socket server yang di dalamnya terdapat suatu method untuk melakukan penjumlahan suatu data. Socket client akan mengirimkan data yang akan dijumlahkan dan socket server melakukan proses penjumlahan dan mengirimkan hasilnya ke socket client. Dalam socket programming dapat dibangun dengan menggunakan dua model paket yaitu TCP (untuk connection oriented) dan UDP (untuk connectionless).

# Dasar Teori

* 1. *UDP (User Datagram Protocol)*

UDP merupakan protocol transport yang sederhana. Berbeda dengan TCP yang *connection oriented*, UDP bersifat *connectionless*. Dalam UDP tidak ada *sequencing* (pengurutan kembali) paket yang datang, *acknowledgement* terhadap paket yang datang, atau retransmisi jika paket mengalami masalah di tengah jalan.

Kemiripan TCP dengan UDP adalah pada penggunaan port number, sebagaimana digunakan pada TCP, UDP menggunakan port number ini membedakan pengiriman datagram ke beberapa aplikasi berbeda yang terletak pada komputer yang berbeda. Karena sifatnya yang *connectionless* dan *ureliable*, UDP digunakan oleh aplikasi aplikasi yang secara periodic melakukan aktivitas tertentu.

Permasalahan dalam komunikasi UDP, pertama ukuran pesan, proses penerimaan perlu menyediakan sejumlah elemen array untuk menerima sebuah pesan. Jika pesan terlalu besar akan dipotong sesuai dengan ukuran array. Kedua blocking, socket menyediakan non-blocking send dan blocking receive untuk komunikasi datagram. Ketiga timeout, operasi blocking untuk receive cocok untuk menunggu permintaan client. Namun sangat tidak baik jika berada dalam keadaan blocking terus pada saat menunggu balasan, maka diperlukan timeout.

* 1. *UDP Socket (Datagram)*

UDP socket diciptakan dengan menggunakan class DatagramSocket. Sebuah class DatagramSocket melakukan send dan receive data menggunakan packet, yang direpresentasikan dalam object DatagramPacket. Dalam suatu kasus dua buah program yang berkomunikasi dengan menggunakan koneksi UDP, kedua program tersebut harus mempunyai koneksi DatagramSocket pada suatu port dalam suatu komputer. Ini dapat diselesaikan dengan membuat objek DatagramSocket seperti berikut

*DatagramSocket coonect = new DatagramSocket(5555);*

Dalam contoh tersebut kita sedang membuat suatu koneksi UDP Socket dengan spesifik portnya 5555 pada local komputer. Jika kita tidak tahu port mana yang tidak sedang digunakan/tidak yakin bahwa port tersebut tidak sedang digunakan, maka dapat dilakukan dengan cara :

*DatagramSocket ds = new DatagramSocket();*

Dalm kasus ini kita tidak menggunakan port pada komputer local. Kita dapat mencari port yang sedang digunakan dengan cara

*int portNumber = ds.getLocalPort();*

Data dalam komunikasi paket UDP dikirimkan diatas sebuah DatagramSocket dengan menggunakan DatagramPacket, dan setiap DatagramSocket mempunyai sebuah buffer data, alamat komputer remote yang akan dikirimi data dan nomer port yang digunakan server untuk listening. Untuk suatu client yang mengirimkan data ke server yang sedang listening pada port 5555 dan pada komputer/host “maswi”, kita dapat menuliskan listing programnya sebagai berikut :

*byte buf[] ={“h”,”e”,”l”,”l”,”o”};*

*InetAddress address = InetAddress.getByName(“maswi”);*

*DatagramPacket paket = new DatagramPacket(buf,buf.length,address,5555);*

*ds.send(paket);*

Sedangkan untuk remote servernya adalah

*byte buf[] = new byte[256];*

*DatagramPacket paket = new DatagramPacket(buf,buf.length);*

*sd.receive(paket);*

# ALAT DAN LANGKAH PERCOBAAN

Alat yang digunakan adalah sebagai berikut:

1. Laptop
2. IDE Java Eclipse

Langkah dalam melakukan pemrograman socket dengan packet UDP antara lain

- Open socket

- Menciptakan sebuah input data stream

- Menciptakan sebuah output data stream

- Close socket

Program ini dijalankan dengan Java.

# Hasil dan Analisis Percobaan

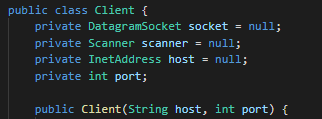
Program terdiri dari 2 file utama yaitu Server.java dan Client.java

1. Server.java



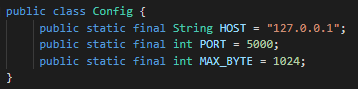
Server menginisiasi socket dan host dengan nilai null, serta menginisiasi harga *shirt* dan *pants* masing – masing 55000 dan 50000. Kemudian saat server dijalankan maka akan diciptakan koneksi dengan client pada suatu port di dalam komputer dengan host tertentu. Lalu server akan menunjukkan inputan dari client (menerima inputan dari client) yang kemudian dijumlah lalu akan diminta oleh client. Saat server diberhentikan maka akan menutup semua socket, inputStream, dan outputStream.

2. Client.java



Client menginisiasi object socket, scanner, dan host dari Server.java agar bisa dipakai di client.

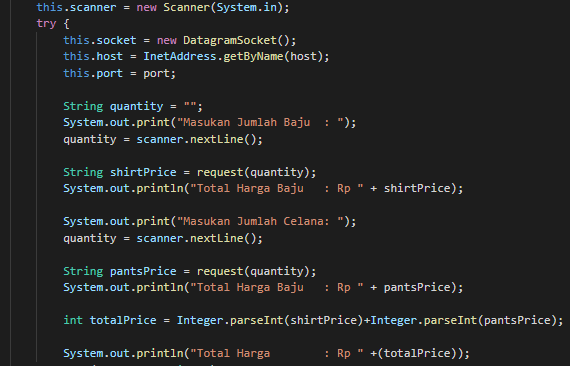
Host dan Port yang digunakan ada pada config.java, yatu 127.0.0.1 dan 5000.



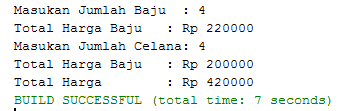


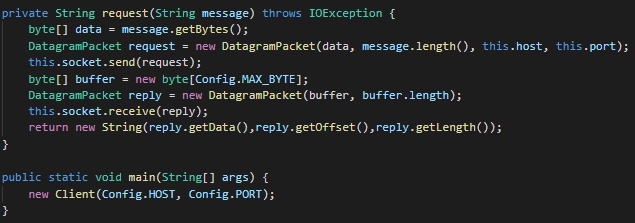


Melakukan pengecekan server dan host dengan mencoba mengambil informasi dari server.java



Saat client dijalankan maka akan muncul “Masukkan Jumlah Baju : ” dan “Masukan Jumlah Celana: ”. User diminta untuk menginput angka yang dibawa ke server sebagai variabel *quantity* untuk dikalkulasikan di dalam server. Setelah menekan enter maka akan ditampilkan hasil perhitungan dari dalam server. Berikut hasil running





Data inputan dibawa dari client ke server dengan menggunakan DatagramPacket, dan demikian juga sebaliknya dari server akan membalas ke client dengan menggunakan DatagramPacket.

# Kesimpulan

Dalam model paket UDP, komunikasi yang berlangsung harus dengan kesepakatan antara clint dengan server. Sehingga paket yang dikirim akan benar-benar sampai pada penerima dan sesuai dengan kriteria yang dibutuhkan penerima. Hal ini dikarenakan jika terjadi masalah maka pengirim akan melakukan retransmisi. Pada percobaan yang dilakukan, server berhasil menerima input angka yang diinginkan user untuk dijumlahkan. Server juga berhasil memberikan hasil penjumlahan tersebut sehingga client dapat menampilkannya ke layar.

# DaftarPustaka

Winarno. 2020. *Modul Praktikum Sistem Terdistribusi”*.

Oracle. 2017. “*All About Sockets”*. *<https://docs.oracle.com/javase/tutorial/networking/sockets/>*

**[Eska Smara Nofiansi]** Mahasiswa Informatika     Angkatan 2016

**[Fabian Gilanggi]** Mahasiswa Informatika         Angkatan 2017

**[Henny Nurcahyaning Tyas]** Mahasiswa Informatika     Angkatan 2017