|  |
| --- |
| **LAPORAN PRAKTIKUM**  **SISTEM OPERASI**  **“INTERPROCESS”** |
|  |
| **Oleh:**    13323008 Stevi Greis Sianipar  13323015 Darwin Teguh Silaen  13323033 Agus Pranata Marpaung |
| **INSTITUT TEKNOLOGI DEL**  **FAKULTAS VOKASI** |

**1031202 – SISTEM OPERASI**

**INTERPROCESS**

**COMMUNICATION**

|  |  |  |
| --- | --- | --- |
| **Minggu** | **:** | 6 |
| **Setoran** | **:** | Laporan praktikum dalam bentuk *softcopy* yang berisikan hasil observasi program dan jawaban pada nomor yang memberikan pertanyaan. |
| **Batas Waktu Setoran** | **:** | **H+3** |
| **Tujuan** | **:** | Mahasiswa mampu menulis program untuk penerapan komunikasi antar proses melalui jaringan komputer menggunakan konsep Socket dengan menggunakan Bahasa pemrograman Java. |

**Petunjuk Praktikum:**

1. Anda dapat mengerjakan tugas secara berkelompok, sesuai dengan kelompok yang telah ditentukan pada minggu pertama perkuliahan.
2. Setiap kelompok harus menyerahkan setoran pada waktu yang telah ditentukan. Pada laporan harus dicantumkan identitas anggota kelompok seperti NIM, Nama dan kontribusi dalam mengerjakan tugas ini, seperti contoh.



1. Anda hanya diperbolehkan diskusi dengan sesama anggota kelompok.
2. Sebelum memberikan pertanyaan silahkan baca buku yang ada pada referensi.
3. Mencontoh pekerjaan dari orang lain akan dianggap plagiarisme dan anda akan ditindak sesuai dengan sanksi akademik yang berlaku di IT Del atau sesuai dengan kebijakan saya dengan memberikan nilai 0.
4. Jawaban diketik menggunakan word processor kemudian dikonversi ke file berekstensi .pdf
5. Penamaan file **HARUS** sesuai dengan format NoKelompok\_Tugas-X\_NamaTugas.pdf (contoh: 01\_Tugas-2\_MapReduce.pdf).
6. Keterlambatan menyerahkan laporan tidak ditolerir dengan alasan apapun. Oleh karena itu, laporan harus dikumpul tepat waktu.
7. Gunakan Sistem Operasi Linux boleh menggunakan Distro apapun namun disarankan untuk mempermudah praktikum gunakan Ubuntu.

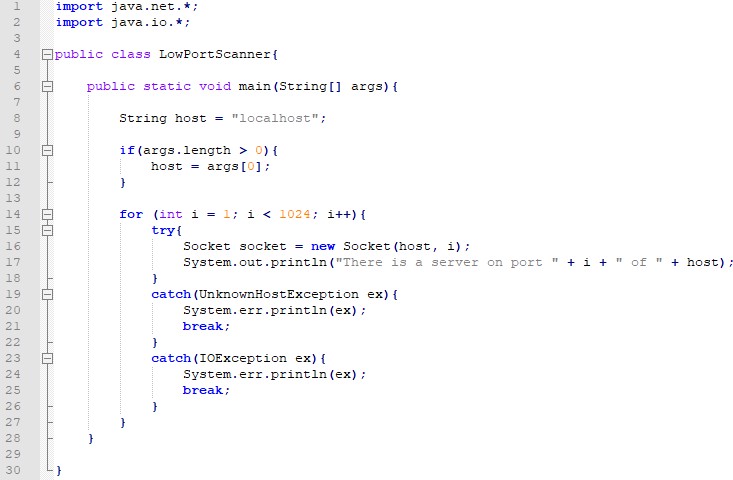
**Referensi:**

1. A. Silberschatz, P.B. Galvin, and G. Gagne, Operating System Concepts, 9th edition, Chapter 1 and 2, John Wiley & Sons, Inc., 2013.

# A. Pemrograman A.1 Socket

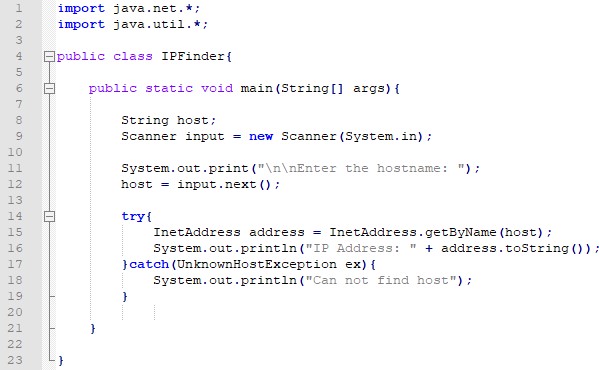
Antar proses dapat saling berkomunikasi satu dengan yang lain melalui jaringan komputer dengan menggunakan Socket. Socket dapat diimplementasikan dengan menggunakan tipe komunikasi TCP Socket dan datagram socket (UDP Socket). Pada umumnya, dua proses yang saling berkomunikasi akan mempunya relasi client/server. Sebelum mengimplementasikan client dan server, selesaikanlah kode program berikut.

1. Tulislah dan eksekusi program berikut.



1. Dari kode program di atas jelaskan apa yang dimaksud dengan:
   1. Import java.net.\*;
   2. Import java.io.\*;
   3. Socket socket = new Socket (host, i);
   4. UnknownHostexception
   5. IOException

1. Tulislah dan eksekusi program berikut.



1. Dari kode program di atas jelaskan apa yang dimaksud dengan:
   1. InetAddress
   2. Method getByName()

1. Modifikasilah program pada no.1 untuk menampilkan IP Address lokal komputer anda.

Contoh keluaran yang diharapkan seperti gambar di bawah.



Berikutnya obervasilah mengenai penerapan konsep client-server dengan menggunakan TCP Socket.

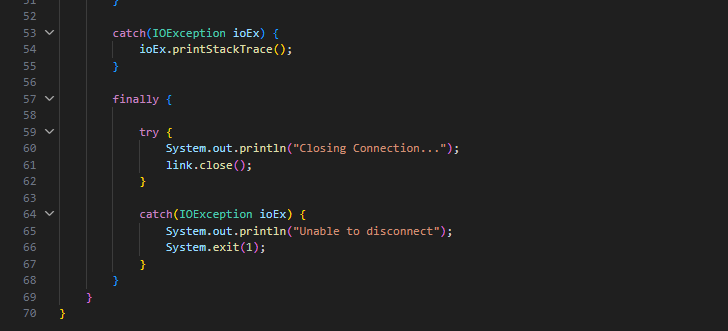
# TCP Sockets

Jalur komunikasi yang dibangun dengan menggunakan TCP/IP socket adalah jenis komunikasi yang berorientasi terhadap koneksi (connection-oriented). Dengan demikian, koneksi antara client dan server tetap terjalin selama terjadi komunikasi antara client dan server dan akan diterminasi jika dan hanya jika adanya pemutusan koneksi dari client atau server. Langkah pertama server akan diimplementasikan kemudian berikutnya adalah client.

1. Untuk mengimplentasikan server, tuliskanlah kode program berikut dengan nama TCPEchoServer.java.

**{Stevi Greis Sianipar\_13323008}**





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

**Jawab:**

**{Stevi Greis Sianipar\_13323008}**

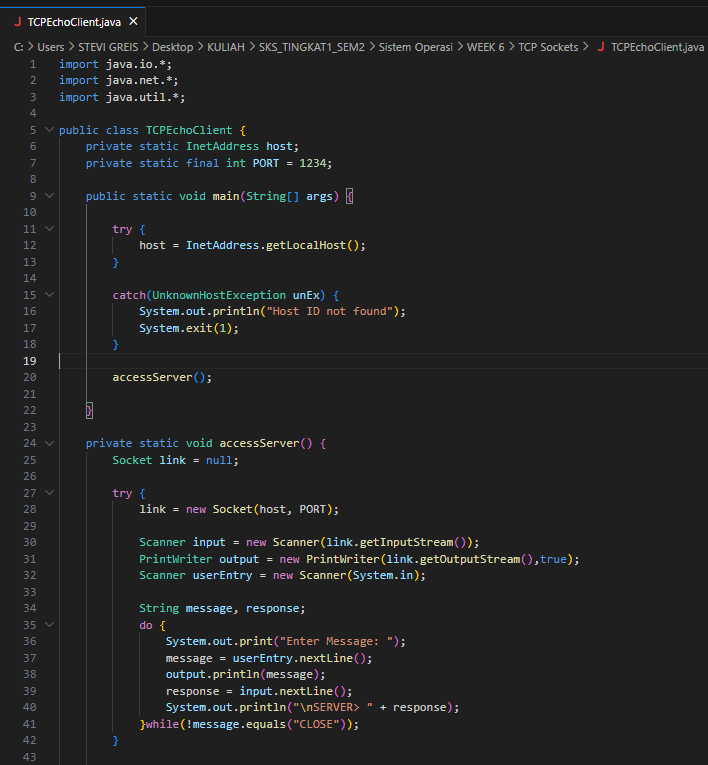
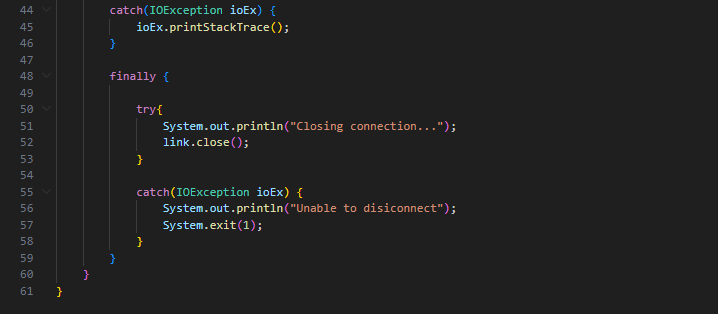
Berikut adalah langkah-langkah untuk membuat sebuah server:

* Siapkan Server: Buatlah sesuatu yang disebut ServerSocket. Ini seperti telinga server yang siap mendengarkan panggilan dari klien. Misalnya, kita bisa membuat ServerSocket yang mendengarkan panggilan di nomor telepon 1234.
* Terima Panggilan dari Klien: Begitu ServerSocket siap, ia akan mendengarkan dan menunggu panggilan dari klien. Ketika ada panggilan masuk, ServerSocket akan menghubungkan klien dengan server dan membentuk koneksi yang baru.
* Baca Pesan dari Klien: Setelah koneksi terbentuk, server akan membaca pesan yang dikirim oleh klien. Ini seperti membaca pesan teks yang dikirimkan melalui telepon.
* Kirim Balasan ke Klien: Server akan merespons pesan dari klien dengan mengirimkan pesan balik. Ini seperti membalas pesan teks yang diterima.
* Tutup Koneksi: Setelah semua pesan telah diproses, server akan menutup koneksi dengan klien. Ini seperti menutup telepon setelah selesai berbicara.
* Ulangi Proses untuk Panggilan Selanjutnya: Server akan terus mendengarkan panggilan berikutnya dan mengulangi proses di atas untuk setiap panggilan yang masuk.

Untuk menghentikan server, klien bisa mengirimkan pesan "TUTUP". Ini akan membuat server menutup koneksi dan menghentikan prosesnya.Untuk menjalankan server, Anda hanya perlu menjalankan program TCPEchoServer. Server akan mulai mendengarkan panggilan pada nomor telepon 1234.

1. Untuk mengimplentasikan client, tuliskanlah kode program di halaman selanjutnya dengan nama TCPEchoClient.java.

**{Stevi Greis Sianipar\_13323008}**

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

**Jawab:**

**{Stevi Greis Sianipar\_13323008}**

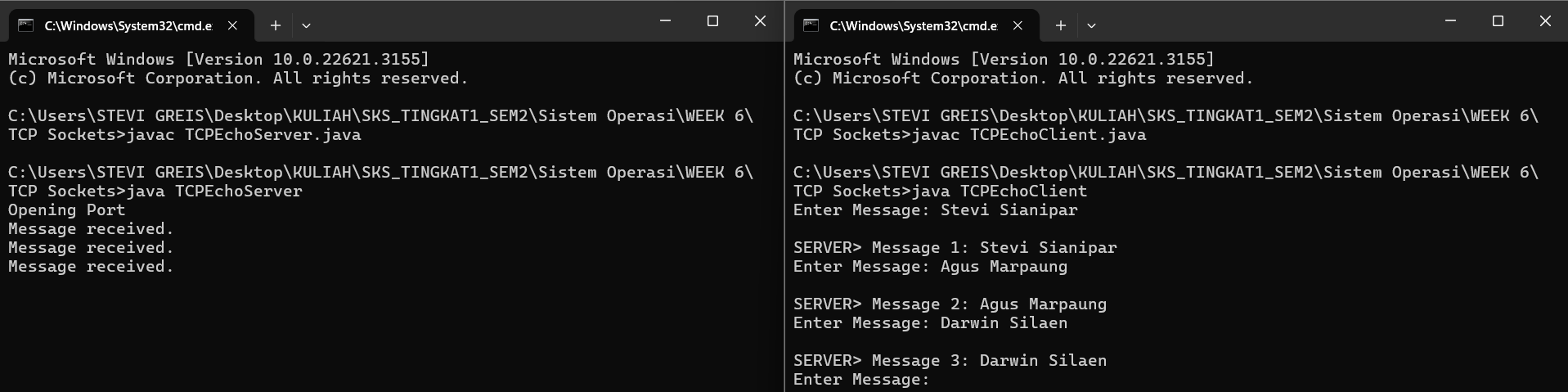
Berikut adalah langkah-langkah untuk membuat sebuah klien dari kode yang diberikan:

* Impor pustaka Java yang diperlukan, yaitu java.io, java.net, dan java.util.
* Buat sebuah kelas publik dengan nama TCPEchoClient.
* Tentukan sebuah variabel statis privat dengan nama host bertipe InetAddress dan sebuah variabel konstan statis final dengan nama PORT dan nilai 1234.
* Pada metode utama, buat blok try untuk mendapatkan host lokal menggunakan InetAddress.getLocalHost(). Jika alamat host tidak dapat ditemukan, cetak pesan kesalahan dan keluar dari program.
* Panggil metode accessServer.
* Dalam metode accessServer, buat sebuah objek Socket dengan nama link dan inisialisasikan dengan melewatkan variabel host dan konstanta PORT ke konstruktor.
* Buat sebuah objek Scanner dengan nama input untuk membaca data dari server menggunakan aliran masukan soket. Buat sebuah objek PrintWriter dengan nama output untuk menulis data ke server menggunakan aliran keluaran soket.
* Buat sebuah objek Scanner dengan nama userEntry untuk membaca masukan pengguna.
* Buat sebuah loop do-while untuk secara berulang mengirim pesan ke server dan mencetak balasannya sampai pengguna memasukkan "TUTUP".
* Di dalam loop, minta pengguna untuk memasukkan sebuah pesan dan baca menggunakan userEntry.nextLine(). Tulis pesan ke server menggunakan output.println().
* Baca respon dari server menggunakan input.nextLine() dan cetak ke konsol dengan awalan "SERVER> ".
* Dalam blok try-catch, tutup koneksi soket menggunakan link.close(). Jika terjadi pengecualian saat menutup koneksi, cetak pesan kesalahan dan keluar dari program.
* Terakhir, tutup objek input, output, dan userEntry untuk membebaskan sumber daya sistem yang digunakan.

1. Eksekusilah kedua program client dan server kemudian cetaklah hasil yang Anda dapatkan.

**Jawab:**

**{Stevi Greis Sianipar\_13323008}**

****

# B. Tugas Teori

Jelaskanlah ketiga tipe Socket berikut:

1. Connection Oriented (TCP)

**Jawab:**

**{ Agus Pranata Marpaung}**

Connected oriented adalah istilah yang digunakan dalam bidang jaringan komputer untuk menggambarkan protokol komunikasi yang memerlukan pembentukan koneksi yang stabil antara dua titik (misalnya, server dan klien) sebelum data dapat dikirimkan. Protokol ini, seperti TCP (Transmission Control Protocol), menjamin bahwa semua data yang dikirimkan akan sampai ke tujuan dalam urutan yang benar dan tanpa kesalahan.

1. Connectionless (UDP)

**Jawab:**

**{Stevi Greis Sianipar\_13323008}**

UDP (User Datagram Protocol) adalah salah satu protokol transport yang digunakan di kalangan jaringan komputer. Differencenya dengan TCP (Transmission Control Protocol), UDP adalah protokol yang tidak terikat (connectionless), yang berarti tidak ada koneksi yang terbentuk antara klien dan server.

Dalam UDP, data dikirim dalam bentuk paket kecil yang disebut datagram. Setiap datagram terdiri dari beberapa bagian, yaitu header dan payload. Header berisi informasi tentang paket, seperti alamat pengirim dan port tujuan, sedangkan payload berisi data yang akan dikirim.

Ketika klien mengirimkan datagram ke server, klien tidak akan mengetahui apakah datagram terkirim atau tidak. Selain itu, datagram juga tidak selalu sampai ke tujuan, atau mungkin sampai dalam urutan yang tidak benar. Namun, kelemahan ini menjadi kekuatan ketika digunakan dalam jaringan yang membutuhkan kinerja tinggi, seperti streaming video atau audio.

1. MulticastSocket

# Tugas Pemrograman

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

**Jawab:**

**{ Agus Pranata Marpaung}**

* Server

Membuat Server (UDPEchoServer.java):

* Import library yang diperlukan.
* Inisialisasi DatagramSocket untuk menerima paket dari klien.
* Buat buffer untuk menerima data.
* Terima paket dari klien.
* Kirim balik paket yang diterima ke klien.
* Tutup socket.
* Client
* Import library yang diperlukan.
* Inisialisasi DatagramSocket untuk mengirim dan menerima paket.
* Baca input dari pengguna.
* Kirim paket ke server.
* Terima balasan dari server.
* Tampilkan balasan dari server.

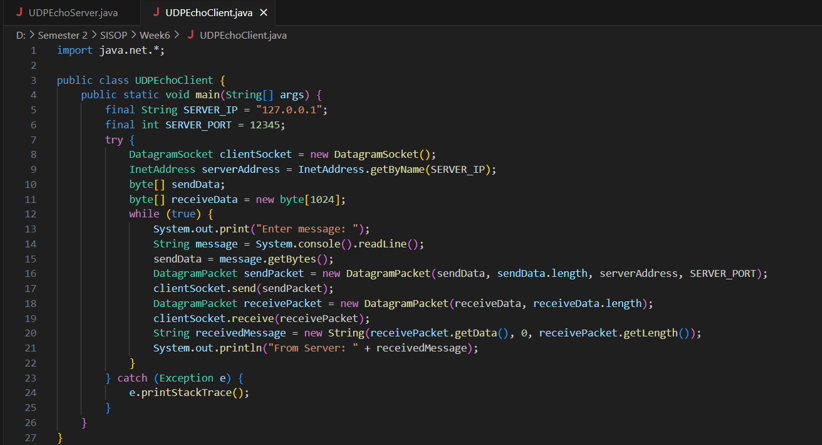
- Tutup socket.

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

**Jawab:**

**{ Agus Pranata Marpaung}**

* UDPEchoClient.java

****

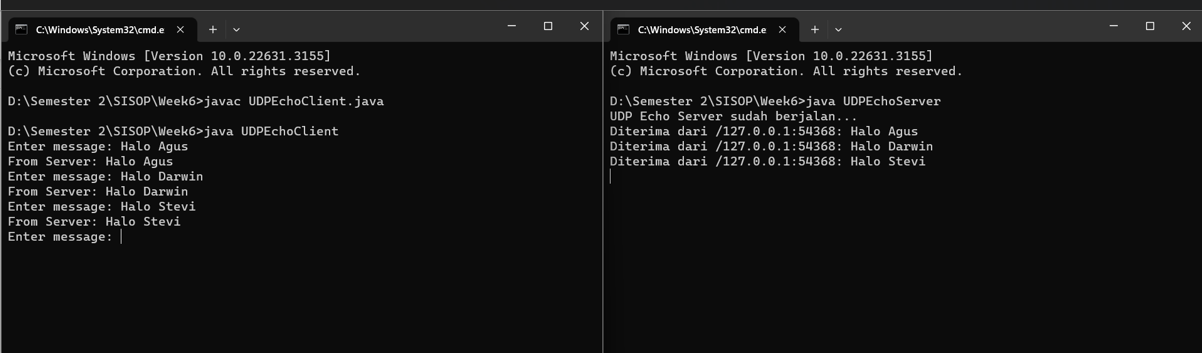
* UDPEchoServer.java

****

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

**Jawab:**

**{ Agus Pranata Marpaung}**

******