LAPORAN

PRAKTIKUM KONSEP JARINGAN

SOCKET PROGRAMMING



Nama : Daegal Prayoga

NRP : 2110171001

KELAS **:** D4 A TEKNIK INFORMATIKA

**POLITEKNIK ELEKTRONIKA NEGERI SURABAYA**

**TAHUN PELAJARAN 2018/2019**

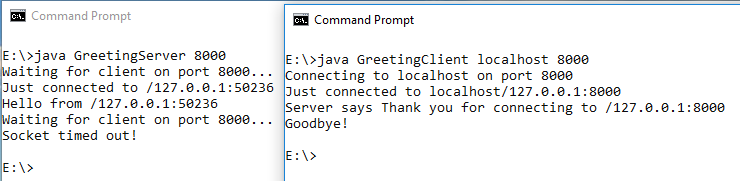
**PERCOBAAN**

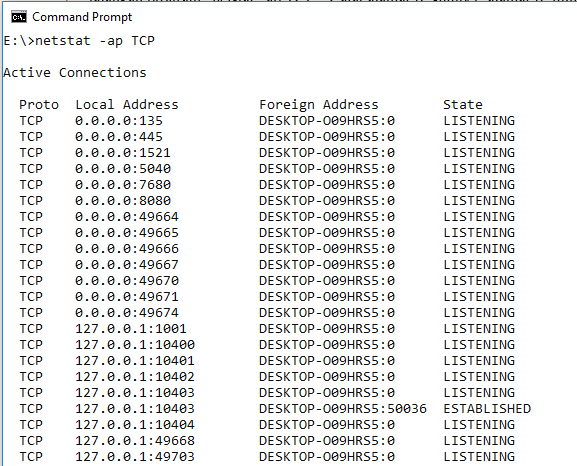
1. Tulis kembali program C1 dalam editor yang anda sukai ! Program Socket Server Sederhana

|  |
| --- |
| // File Name GreetingServer.java  import java.net.\*;  import java.io.\*;  public class GreetingServer extends Thread  {  private ServerSocket serverSocket;  public GreetingServer(int port) throws IOException  {  serverSocket = new ServerSocket(port);  serverSocket.setSoTimeout(10000);  }  public void run()  {  while(true)  {  try  {  System.out.println("Waiting for client on port " +  serverSocket.getLocalPort() + "...");  Socket server = serverSocket.accept();  System.out.println("Just connected to "  + server.getRemoteSocketAddress());  DataInputStream in =  new DataInputStream(server.getInputStream());  System.out.println(in.readUTF());  DataOutputStream out =  new DataOutputStream(server.getOutputStream());  out.writeUTF("Thank you for connecting to "  + server.getLocalSocketAddress() + "\nGoodbye!");  server.close();  }catch(SocketTimeoutException s)  {  System.out.println("Socket timed out!");  break;  }catch(IOException e)  {  e.printStackTrace();  break;  }  }  }  public static void main(String [] args)  {  int port = Integer.parseInt(args[0]);  try  {  Thread t = new GreetingServer(port);  t.start();  }catch(IOException e)  {  e.printStackTrace();  }  }  } |

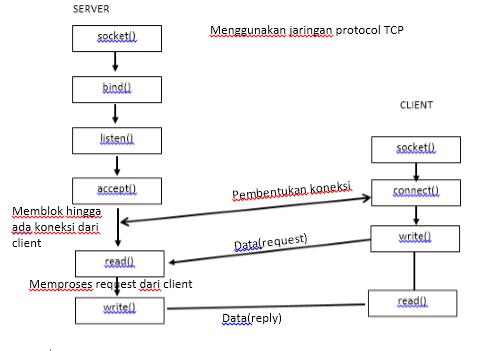
Program Socket Client Sederhana

|  |
| --- |
| // File Name GreetingClient.java  import java.net.\*;  import java.io.\*;  public class GreetingClient  {  public static void main(String [] args)  {  String serverName = args[0];  int port = Integer.parseInt(args[1]);  try  {  System.out.println("Connecting to " + serverName +  " on port " + port);  Socket client = new Socket(serverName, port);  System.out.println("Just connected to "  + client.getRemoteSocketAddress());  OutputStream outToServer = client.getOutputStream();  DataOutputStream out = new DataOutputStream(outToServer);  out.writeUTF("Hello from "  + client.getLocalSocketAddress());  InputStream inFromServer = client.getInputStream();  DataInputStream in =  new DataInputStream(inFromServer);  System.out.println("Server says " + in.readUTF());  client.close();  }catch(IOException e)  {  e.printStackTrace();  }  }  } |

Output kedua program diatas :

1. Jalankan program “netstat –ap TCP”. Catat alamat IP sumber, alamat IP tujuan dan nomor port yang sedang dalam keadaan menunggu

3. Buat flowchart untuk menggambarkan interaksi program C1 dan browser anda!



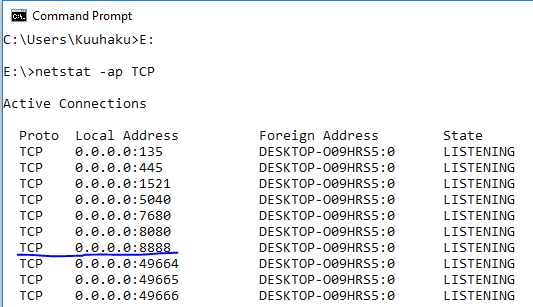
1. Tulis kembali program C2, C3 dalam editor yang anda sukai! Program Aplikasi HTTP Server Sederhana

|  |
| --- |
| ///A Simple Web Server (WebServer.java)  import java.io.BufferedReader;  import java.io.InputStreamReader;  import java.io.PrintWriter;  import java.net.ServerSocket;  import java.net.Socket;  public class webServer {  /\*\*  \* WebServer constructor.  \*/  protected void start() { ServerSocket s; int port = 8888;  System.out.println("Webserver starting up on port " +port);  System.out.println("(press ctrl-c to exit)"); try {  // tahap bind(),membuat socket  s = new ServerSocket(port);  } catch (Exception e) {  System.out.println("Error: " + e);  return;  }  System.out.println("Waiting for connection"); for (;;) {  try {  // tahap listen(), menunggu koneksi  Socket remote = s.accept();  // tahap accept()  System.out.println("Connection, sending data.");  BufferedReader in = new BufferedReader(new  InputStreamReader(remote.getInputStream())); PrintWriter out = new  PrintWriter(remote.getOutputStream());  // membaca request  String str = ".";  while (!str.equals(""))  str = in.readLine();  // Mengirim response dan mengirim HTTP headers  out.println("HTTP/1.0 200 OK");  out.println("Content-Type: text/html");  out.println("Server: Bot");  // Batas pengiriman header ditandai dengan baris kosong  out.println("");  // Mengirim halam HTML  out.println("<H1>Selamat datang di web server percobaan</H1>");  out.println("<blink>Selamat datang di web server percobaan</blink>");  out.flush();  remote.close();  } catch (Exception e) { System.out.println("Error: " + e);  }  }  }  /\*\*  \* Menjalankan server  \*  \* @param args  \* Command line parameters are not used.  \*/  public static void main(String args[]) {  webServer ws = new webServer();  ws.start();  }  } |

1. Jalankan program Aplikasi Server C2! Amati perubahan yang sedang terjadi pada komputer anda dengan menggunakan perintah pada nomor 2!

Output program tersebut adalah sebagai berikut

Perubahan yang terjadi : (terdapat local address baru yaitu 8888)



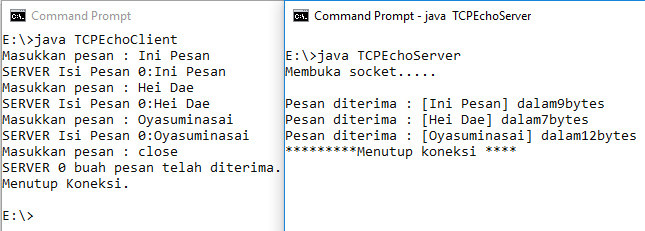
Program Aplikasi Chat Server dengan menggunakan TCP

|  |
| --- |
| import java.io.BufferedReader;  import java.io.IOException;  import java.io.InputStreamReader;  import java.io.PrintWriter;  import java.net.ServerSocket;  import java.net.Socket;  public class TCPEchoServer {  private static ServerSocket servSock;  private static final int PORT = 1234;  public static void main(String args[]) {  System.out.println("Membuka socket.....\n");  try {  servSock = new ServerSocket(PORT);  } catch (IOException e) { System.out.println("Gagal membuka port !!!");  System.exit(1);  }  do {  run();  } while (true);  }  private static void run() {  Socket link = null;  try {  link = servSock.accept();  BufferedReader in = new BufferedReader(new InputStreamReader(link.getInputStream()));  PrintWriter out = new  PrintWriter(link.getOutputStream(), true);  int numMessages = 0;  String message = in.readLine();  while(!message.equals("close")) {  System.out.println("Pesan diterima : [" +message.toString() + "] dalam" + message.length() + "bytes");  //System.out.println("Message received"); numMessages++;  out.println("Isi Pesan " + numMessages + ":" + message);  message = in.readLine();  }  out.println(numMessages + " buah pesan telah diterima.");  } catch (IOException e) { } finally {  try {  System.out.println("\*\*\*\*\*\*\*\*\*Menutup koneksi \*\*\*\*");  link.close();  } catch (IOException e) {  System.out.println("Tidak dapat memustukan koneksi");  System.exit(1);  }  }  }  } |

Program Aplikasi Chat Client dengan menggunakan TCP

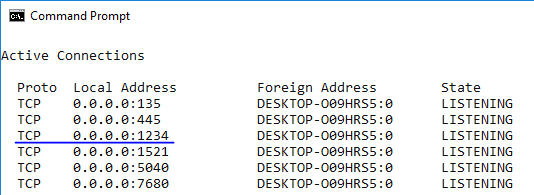
|  |
| --- |
| import java.io.BufferedReader;  import java.io.IOException;  import java.io.InputStreamReader;  import java.io.PrintWriter;  import java.net.InetAddress;  import java.net.Socket;  import java.net.UnknownHostException;  public class TCPEchoClient {  private static String strHost;  private static InetAddress host;  private static final int PORT = 1234;  public static void main(String args[]) {  try {  // host = InetAddress.getLocalHost(); strHost = "localhost" ; // <- Masukan sesuai dengan tujuan    host = InetAddress.getByName(strHost);    } catch (UnknownHostException e) {  System.out.println("Alamat tidak ditemukan");  System.exit(1);  }  run();  }  private static void run() {  Socket link = null;  try {  link = new Socket(host, PORT);  BufferedReader in = new BufferedReader(new InputStreamReader(link.getInputStream()));  PrintWriter out = new PrintWriter(link.getOutputStream(), true);    BufferedReader userEntry = new BufferedReader(new InputStreamReader(System.in));  String message, response;  do {  System.out.print("Masukkan pesan : ");  message = userEntry.readLine();  out.println(message);  response = in.readLine();  System.out.println("SERVER " + response);  } while (!message.equals("close"));  } catch (IOException e) { e.printStackTrace();  } finally {  try {  System.out.println("Menutup Koneksi."); link.close();  } catch (IOException e) { System.out.println("Tidak dapat memutuskan koneksi!");  System.exit(1);  }  }  }  } |

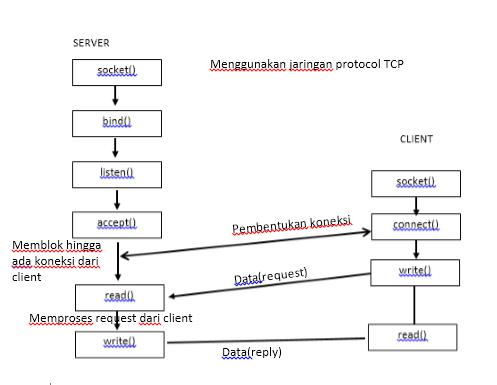
1. Jalankan program C3! Amati perubahan yang sedang terjadi pada komputer anda dengan menggunakan perintah pada nomor 2 !

Output kedua program tersebut adalah sebagai berikut :

Ketika TCPEchoClient dijalankan maka kita bias mengirimkan pesan ke Server, yang nantinya di server akan muncul pesan yang dikirim beserta ukurannya. Kemudian untuk mengakhiri bias mengetikkan close, sehingga koneksi akan terputus.

Perbedaan dalam port address 1234 :



7. Buatlah flowchart untuk menggambarkan interaksi program C2 dan program C3

8. Program Utama (main program) Aplikasi Server dengan Multithread

Server

|  |
| --- |
| import java.io.IOException;  import java.net.ServerSocket;  public class TCPEchoServerThread {  private static ServerSocket servSock;  private static final int PORT = 12345;  public TCPEchoServerThread() {  }  public void start() {  try {  servSock = new ServerSocket(PORT);  while (true) {  Thread clientThread = new Thread(new clientHandler(servSock.accept()));  clientThread.start();  }  } catch (IOException e) { e.printStackTrace();  } finally {  try {  System.out.println("Menutup koneksi....");  servSock.close();  } catch (IOException e) {  System.out.println("Tidak dapat memustukan koneksi");  e.printStackTrace();  System.exit(1);  }  }  }  public static void main(String[] args) {  TCPEchoServerThread es = new TCPEchoServerThread();  System.out.println("Server telah berjalan di komputer ini pada port " +PORT);  es.start();  }  } |

Client

|  |
| --- |
| import java.io.BufferedReader;  import java.io.IOException;  import java.io.InputStreamReader;  import java.io.PrintWriter;  import java.net.\*;  public class clientHandler implements Runnable {  private static int numConnections;  private int connectionId = 0;  Socket link;  public clientHandler(Socket s) {  connectionId = numConnections++;  System.out.println("Melayani koneksi ke-"+ connectionId);  link = s;  }  public void run() {  PrintWriter out = null;  BufferedReader in = null;  int numMessages = 0;  try {  out = new PrintWriter(link.getOutputStream(), true);  in = new BufferedReader(new InputStreamReader(link.getInputStream()));  String message=in.readLine();  while (!message.equals("close") ) {  System.out.println("Pesan diterima : [" +message.toString() + "] dari client " +connectionId +" dalam " + message.length() + " bytes");  numMessages++;  out.println("Isi Pesan " + numMessages + ":" + message);  message = in.readLine();  }  } catch (Exception e) {  e.printStackTrace();  } finally {  out.close();  try {  in.close();  link.close();  System.out.println("Menutup koneksi, #" + connectionId);  } catch (IOException e) {  e.printStackTrace();  }  }  }  } |

Output Program diatas :

