**Sistem Tersebar**

**Ulangan Tengah Semester**

Diajukan Untuk Memenuhi Ulangan Tengah Semester

Sistem Tersebar



Disusun Oleh

Nama Antum (NPM antum)

D3 Teknik Informatika 2A

**PROGRAM STUDI DIPLOMA III TEKNIK INFORMATIKA**

**POLITEKNIK POS INDONESIA**

**2021**

|  |  |  |
| --- | --- | --- |
| **POLITEKNIK POS INDONESIA**  UJIAN TENGAH SEMESTER GANJIL TAHUN 2021 / 2022 | | |
| **MATA KULIAH** | **:** | Sistem Tersebar |
| **PRODI & KELAS** | **:** | D3 Teknik Informatika |
| **HARI & TANGGAL** | **:** | Rabu 15 Desember 2021 |
| **WAKTU** | **:** | 24 Jam |
| **SIFAT** | **:** | *Open Book* |
| **JENIS** | **:** | Praktikum Daring |

# PETUNJUK SOAL :

1. Buatlah application project untuk soal dibawah ini menggunakan bahasa pemrograman.
2. Buatlah laporan praktikum dari project yang dikerjakan, isi dari laporan adalah : cover, soal ujian, script program dan screen shoot running program (setiap output program pada soal dibawah).
3. Penamaan Laporan Praktikum adalah 20211-UTSSISTER-Nama Masing-masing
4. Segala bentuk plagiat akan menyebabkan hasil pekerjaan tidak akan diperiksa nilai UTS = 0.

# SOAL :

1. Buatlah program penyimpanan data mahasiswa client-server dengan mengimplemantasikan Object Serialization. Data mahasiswa terdiri dari: nim, nama, asal, dan kelas praktikum. Data diinputkan dari sisi client melalui keyboard. Data object dari client dikirim ke server kemudian server menyimpan dalam suatu file. (**bobot 50**)

Jawaban :

* Program
* Clintmhs

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template

\*/

package Soal\_no\_1;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.net.ServerSocket;

import java.net.Socket;

import java.net.InetAddress;

/\*\*

\*

\* @author Raihan

\*/

public class Serverrai {

public static void main(String args[]) {

Mahasiswacl mhs = null;

try {

ServerSocket socketConnection = new ServerSocket(11111);

System.out.println("Server Menyala....");

Socket pipe = socketConnection.accept();

ObjectInputStream serverInputStream = new

ObjectInputStream(pipe.getInputStream());

ObjectOutputStream serverOutputStream = new

ObjectOutputStream(pipe.getOutputStream());

mhs = (Mahasiswacl)serverInputStream.readObject();

serverOutputStream.writeObject(mhs);

System.out.println("Packet Received from " + pipe.getInetAddress());

System.out.println("Packet Sending to e:/uts\_sisbar\_nomer\_1.txt from " + pipe.getLocalAddress());

serverInputStream.close();

serverOutputStream.close();

} catch(Exception e) {System.out.println(e);

}

}

}

* Mahasiswacl

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template

\*/

package Soal\_no\_1;

import java.io.\*;

import java.util.\*;

/\*\*

\*

\* @author Raihan

\*/

public class Mahasiswacl implements Serializable{

private String mhsNim;

private String mhsNama;

private String mhsAsal;

private String mhsKelasPraktikum;

private String mhsumur;

private String mhsasal\_sekolah;

Mahasiswacl(String nim, String nama, String asal, String kelasPraktikum, String umur, String asal\_sekolah) {

mhsNim = nim;

mhsNama= nama;

mhsAsal = asal;

mhsKelasPraktikum = kelasPraktikum;

mhsumur = umur;

mhsasal\_sekolah = asal\_sekolah ;

}

public String getMhsNim() {

return mhsNim ;

}

public void setMhsNim(String nim) {

mhsNim = nim;

}

public String getMhsNama() {

return mhsNama ;

}

public void setMhsNama(String nama) {

mhsNama = nama;

}

public String getMhsAsal() {

return mhsAsal ;

}

public void setMhsAsal(String asal) {

mhsAsal = asal;

}

public String getMhsKelasPraktikum() {

return mhsKelasPraktikum ;

}

public String getMhsumur() {

return mhsumur ;

}

public void setMhsumur(String umur) {

mhsumur = umur;

}

public String getasal\_sekolah() {

return mhsasal\_sekolah ;

}

public void setMhsasal\_sekolah(String asal\_sekolah) {

mhsasal\_sekolah = asal\_sekolah;

}

public void setMhsKelasPraktikum(String kelasPraktikum) {

mhsKelasPraktikum = kelasPraktikum;

}

}

* Serverrai

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template

\*/

package Soal\_no\_1;

import java.io.BufferedReader;

import java.io.FileWriter;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.net.Socket;

/\*\*

\*

\* @author Raihan

\*/

public class Clientmhs {

public static void main(String args[]) {

try {

String fileName = "e:/uts\_sisbar\_nomer\_1.txt";

InputStream input = System.in;

InputStreamReader reader = new InputStreamReader(input);

BufferedReader bufReader = new BufferedReader (reader);

String nim = "";

String nama = "";

String asal = "";

String kelas = "";

String umur = "";

String asal\_sekolah ="";

int i = 0;

System.out.println("\nMasukkan Data anda : ");

System.out.print("Masukkan NIM : ");

nim = bufReader.readLine();

System.out.print("Masukkan Nama : ");

nama = bufReader.readLine();

System.out.print("Masukkan Asal : ");

asal = bufReader.readLine();

System.out.print("Masukkan Kelas : ");

kelas = bufReader.readLine();

System.out.print("umur : ");

umur = bufReader.readLine();

System.out.print("asal sekolah : ");

asal\_sekolah = bufReader.readLine();

Mahasiswacl mhs = new Mahasiswacl(nim, nama, asal,kelas,umur,asal\_sekolah);

System.out.println("=======================================");

System.out.println("Review Data :");

System.out.println("Nim = "

+ mhs .getMhsNim());

System.out.println("Nama = "

+ mhs .getMhsNama());

System.out.println("Asal = "

+ mhs .getMhsAsal());

System.out.println("Kelas Praktikum = "

+ mhs .getMhsKelasPraktikum());

System.out.println("umur = "

+ mhs .getMhsumur());

System.out.println("asal sekolah = "

+ mhs .getasal\_sekolah());

Socket socketConnection = new Socket("127.0.0.1", 11111);

ObjectOutputStream clientOutputStream = new

ObjectOutputStream(socketConnection.getOutputStream());

ObjectInputStream clientInputStream = new

ObjectInputStream(socketConnection.getInputStream());

clientOutputStream.writeObject(mhs);

mhs = (Mahasiswacl)clientInputStream.readObject();

FileWriter tulisFile= new FileWriter(fileName);

tulisFile.write((nim)+ "\n");

tulisFile.write((nama)+ "\n");

tulisFile.write((asal)+ "\n");

tulisFile.write((kelas)+ "\n");

tulisFile.write((umur)+ "\n");

tulisFile.write(asal\_sekolah);

tulisFile.close();

System.out.println("=======================================");

System.out.println("Nim = "

+ mhs .getMhsNim());

System.out.println("Nama = "

+ mhs .getMhsNama());

System.out.println("Asal = "

+ mhs .getMhsAsal());

System.out.println("Kelas Praktikum = "

+ mhs .getMhsKelasPraktikum());

System.out.println("Umur = "

+ mhs .getMhsumur());

System.out.println("asal sekolah = "

+ mhs .getasal\_sekolah());

System.out.println("\nData anda sudah di kirimkan pada file uts\_sisbar\_nomer\_1.txt");

clientOutputStream.close();

clientInputStream.close();

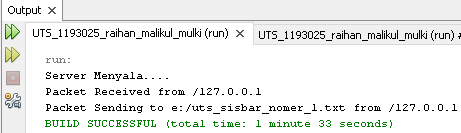
} catch (Exception e) {

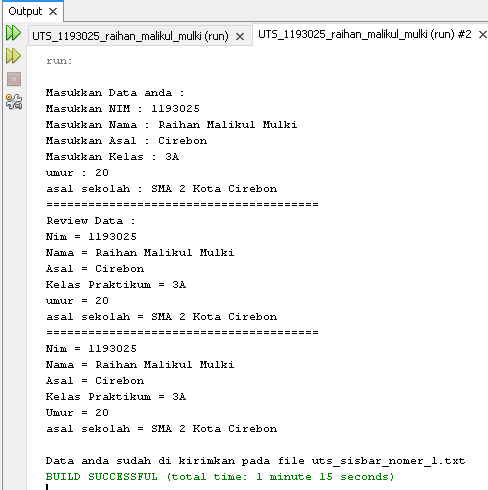
System.err.println(e); }

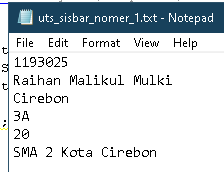
}

}

* Hasil







1. Buatlah program dengan GUI untuk mengirim dan menerima pesan teks dan gambar yang mengimplementasikan TCP. (**bobot 50**)

Jawaban :

* Program
* Clent

/\*\*

\* @author Raihan

\*/

import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

import java.net.\*;

import javax.swing.\*;

import javax.swing.text.DefaultCaret;

public class JavaClient extends JFrame implements ActionListener

{

// Extra variables

static String message = "";

static String userName = "";

static String iPAddress = null;

// Networking Variables

static Socket socket = null;

static PrintWriter writer = null;

// // Graphics Variables

static JTextArea msgRec = new JTextArea(100, 50);

static JTextArea msgSend = new JTextArea(100, 50);

JButton send = new JButton("Send");

JScrollPane pane2, pane1;

JMenuBar bar = new JMenuBar();

JMenu messanger = new JMenu("ChatBox V1.0");

JMenuItem logOut = new JMenuItem("Log Out");

JMenuItem about = new JMenuItem("about");

public JavaClient() {

super("Java Client");

setBounds(0, 0, 407, 495);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setResizable(false);

setLayout(null);

msgRec.setEditable(false);

msgRec.setBackground(Color.BLACK);

msgRec.setForeground(Color.WHITE);

msgRec.setText("");

msgRec.setWrapStyleWord(true);

msgRec.setLineWrap(true);

pane2 = new JScrollPane(msgRec);

pane2.setBounds(0, 0, 400, 200);

pane2.setVerticalScrollBarPolicy(JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS);

add(pane2);

msgSend.setBackground(Color.WHITE);

msgSend.setForeground(Color.BLACK);

msgSend.setLineWrap(true);

msgSend.setWrapStyleWord(true);

msgSend.setText("Tuliskan Pesan ");

pane1 = new JScrollPane(msgSend);

pane1.setBounds(0, 200, 400, 200);

pane1.setVerticalScrollBarPolicy(JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS);

add(pane1);

send.setBounds(0, 400, 400, 40);

add(send);

send.addActionListener(this);

bar.add(messanger);

messanger.add(logOut);

logOut.addActionListener(this);

bar.add(about);

about.addActionListener(this);

setJMenuBar(bar);

addWindowFocusListener(new WindowFocusListener() {

@Override

public void windowGainedFocus(WindowEvent e) {

if (!msgRec.getText().equals("")) {

System.out.println("Yes Focus");

writer.flush();

}

}

@Override

public void windowLostFocus(WindowEvent e) {

if (!msgRec.getText().equals("")) {

writer.flush();

}

}

});

if ((userName) != null) {

setVisible(true);

} else {

System.exit(0);

}

}

/\*\*

\* @param args

\*/

public static void main(String[] args) throws Exception {

// TODO Auto-generated method stub

userName = JOptionPane.showInputDialog("User Name (Client)");

iPAddress = JOptionPane.showInputDialog("Masukan IP Address");

// swing thread

(new Thread(new Runnable() {

public void run() {

new JavaClient();

}

})).start();

socket = new Socket(iPAddress, 8888);

msgRec.setText("Berhasil Terkoneksi!");

// listening port thread

(new Thread(new Runnable() {

public void run() {

try {

BufferedReader reader = new BufferedReader(

new InputStreamReader(socket.getInputStream()));

String line = null;

boolean testFlag = true;

while ((line = reader.readLine()) != null) {

msgRec.append("\n" + line);

cursorUpdate();

if (!reader.ready()) {

testFlag = true;

}

}

} catch (IOException ee) {

try {

socket.close();

} catch (IOException eee) {

eee.printStackTrace();

}

ee.printStackTrace();

}

}

})).start();

try {

writer = new PrintWriter(socket.getOutputStream(), true);

} catch (IOException e) {

try {

socket.close();

} catch (IOException eee) {

}

}

}

// ActionEvents

@Override

public void actionPerformed(ActionEvent e) {

Object scr = e.getSource();

if (scr == send) {

sendMessage();

} else if (scr == logOut) {

System.exit(0);

}

else if (scr == about) {

JOptionPane.showMessageDialog(this,

"UTS\_Raihan\_Malikul\_Mulki\n");

}

}

private void sendMessage() {

writer.println(userName + " :" + msgSend.getText());

msgRec.append("\nMe: " + msgSend.getText());

writer.flush();

cursorUpdate();

msgSend.setText("");

msgSend.setCaretPosition(0);

}

private static void cursorUpdate() {

// Update cursor position

DefaultCaret caret = (DefaultCaret) msgRec.getCaret();

caret.setDot(msgRec.getDocument().getLength());

DefaultCaret caret2 = (DefaultCaret) msgSend.getCaret();

caret2.setDot(msgSend.getDocument().getLength());

}

}

// A Java Client program will take a Server Ip address

* Server

/\*\*

\* @author Raihan

\*/

import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

import java.net.\*;

import javax.swing.\*;

import javax.swing.text.DefaultCaret;

public class JavaServer extends JFrame implements ActionListener{

static String message= "";

static String username= "";

static ServerSocket server=null;

static Socket socket=null;

static PrintWriter writer=null;

static JTextArea msgRec=new JTextArea(100,50);

static JTextArea msgSend=new JTextArea(100,50);

JButton send= new JButton("Send");

JScrollPane pane2, pane1;

JMenuBar bar=new JMenuBar();

JMenu messanger=new JMenu("ChatBox V1.0");

JMenuItem logOut=new JMenuItem("Log Out");

JMenuItem about=new JMenuItem("about");

public JavaServer() {

super("Java Server");

setBounds(0, 0, 407, 495);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setResizable(false);

setLayout(null);

msgRec.setEditable(false);

msgRec.setBackground(Color.BLACK);

msgRec.setForeground(Color.WHITE);

msgRec.setText("");

msgRec.setWrapStyleWord(true);

msgRec.setLineWrap(true);

pane2=new JScrollPane(msgRec);

pane2.setBounds(0, 0, 400, 200);

pane2.setVerticalScrollBarPolicy(JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS);

add(pane2);

msgSend.setBackground(Color.WHITE);

msgSend.setForeground(Color.BLACK);

msgSend.setLineWrap(true);

msgSend.setWrapStyleWord(true);

msgSend.setText("Tuliskan Pesan Disini!");

pane1=new JScrollPane(msgSend);

pane1.setBounds(0, 200, 400, 200);

pane1.setVerticalScrollBarPolicy(JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS);

add(pane1);

send.setBounds(0, 400, 400, 40);

add(send);

send.addActionListener(this);

bar.add(messanger);

messanger.add(logOut);

logOut.addActionListener(this);

bar.add(about);

about.addActionListener(this);

setJMenuBar(bar);

addWindowFocusListener(new WindowFocusListener() {

@Override

public void windowGainedFocus(WindowEvent e) {

if( !msgRec.getText().equals("")) {

System.out.println("Yes Focus");

writer.flush();

}

}

@Override

public void windowLostFocus(WindowEvent e) {

if(!msgRec.getText().equals("")) {

writer.flush();

}

}

});

if((username)!=null) {

setVisible(true);

}

else {

System.exit(0);

}

}

public static void main(String[] args) {

username=JOptionPane.showInputDialog("User Name (Server)");

(new Thread(new Runnable() {

public void run() {

new JavaServer();

}

})).start();

try {

server=new ServerSocket(8888);

System.out.println(server.getInetAddress().getLocalHost());

socket=server.accept();

}

catch(Exception e)

{

System.out.println(e);

}

msgRec.setText("Terkoneksi");

(new Thread(new Runnable() {

public void run() {

try {

BufferedReader reader=new BufferedReader(new InputStreamReader(socket.getInputStream()));

String line=null;

boolean testFlag=true;

while((line=reader.readLine())!=null) {

msgRec.append("\n"+line);

cursorUpdate();

}

}

catch(IOException ee)

{

try {

server.close();

socket.close();

}

catch(IOException eee)

{

eee.printStackTrace();

}

}

}

})).start();

try {

writer=new PrintWriter(socket.getOutputStream(), true);

}

catch(IOException e) {

try {

server.close();

socket.close();

}

catch(IOException eee) {

}

}

}

public void actionPerformed(ActionEvent e) {

Object scr=e.getSource();

if(scr==send) {

sendMessage();

}

else if(scr==logOut) {

System.exit(0);

}

else if(scr==about) {

JOptionPane.showMessageDialog(this, "UTS\_Raihan\_Malikul\_Mulki\n");

}

}

public void sendMessage() {

writer.println(username+":"+msgSend.getText());

writer.flush();

msgRec.append("\nVena: "+msgSend.getText());

cursorUpdate();

msgSend.setText("");

msgSend.setCaretPosition(0);

}

public static void cursorUpdate() {

DefaultCaret caret=(DefaultCaret) msgRec.getCaret();

caret.setDot(msgRec.getDocument().getLength());

DefaultCaret caret2=(DefaultCaret) msgSend.getCaret();

caret2.setDot(msgSend.getDocument().getLength());

}

}

* Hasil

