Nama : La Ode Muhammad Gazali
NIM : 222212696
Kelas : 2KS2

MODUL 14 PEMROGRAMAN BERORIENTASI OBJEK

(Networking Bagian 2)

Penugasan

Laporkan hasil praktikum berikut dengan hasil penugasan dan penjelasannya ke Dosen dalam bentuk file pdf dengan format nama <<nim>>_modul14.

1. Lengkapi kode semua objek
2. Tangkapan layar hasil kode yang dilengkapi
3. Tangkapan layar hasil running

Penyelesaian

Pada modul 14 ini kita akan mengembangkan kode pada modul 13 supaya Server dapat menerima koneksi dari banyak client dengan menerapkan concurrency. Selain itu, pesan yang dikirimkan antara client ke server adalah object dari kelas Pesan yang menerapkan interface Serializable. (Latihan Server saya ubah menjadi Latihan Server2)

Pada project Latihan Server, buatlah kelas baru bernama Pesan dengan kode sebagai berikut (**Pesan.java**)

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this
template
 */
package latihan.server.pkg2;

/**
 *
 * @author U53R
 */
import java.io.Serializable;

public class pesan implements Serializable{
    private String nama;
    private String pesan;
    public pesan(String nama, String pesan) {
        this.nama = nama;
        this.pesan = pesan;
    }

    @Override
    public String toString(){
        return "["+nama+"] "+ pesan;
    }

    /**
    * @return the nama
    */
    public String getNama() {
        return nama;
    }

    /**
    * @param nama the nama to set
    */
    public void setNama(String nama) {
        this.nama = nama;
    }
    /**
    * @return the pesan
    */
    public String getPesan() {
        return pesan;
    }

    /**
    * @param pesan the pesan to set
    */
    public void setPesan(String pesan) {
        this.pesan = pesan;
    }
}
```

Kemudian modifikasilah **LatihanClient.java** menyesuaikan kode berikut ini

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this
template
```

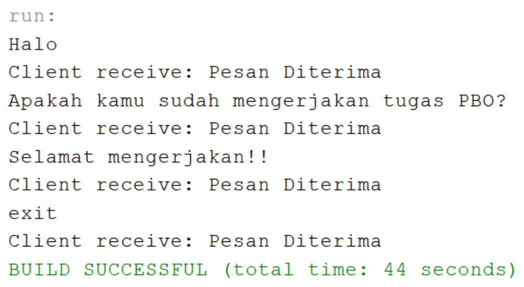
```
 */
package latihan.server.pkg2;

/**
 *
 * @author U53R
 */

import java.io.BufferedOutputStream;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.ObjectOutputStream;
import java.io.PrintWriter;
import java.net.Socket;
import java.util.logging.Level;
import java.util.logging.Logger;

public class LatihanClient {
    public static void main(String args[]){
        String hostName = "localhost";
        int portNumber = 4444;
        try (
            Socket echoSocket = new Socket(hostName, portNumber);
            BufferedOutputStream bos = new
BufferedOutputStream(echoSocket.getOutputStream());
            ObjectOutputStream oos = new ObjectOutputStream(bos);
            BufferedReader in = new BufferedReader(new
InputStreamReader(echoSocket.getInputStream()));
            BufferedReader stdIn = new BufferedReader(new
InputStreamReader(System.in));){

            String msg;
            while((msg = stdIn.readLine()) != null){
                oos.writeObject(new pesan("Andi", msg));
                oos.flush();
                System.out.println("Client receive: "+ in.readLine());
                if(msg.equalsIgnoreCase("exit")) break;
            }
        }catch (IOException ex) {
            Logger.getLogger(LatihanClient.class.getName()).log(Level.SEVERE,null
, ex);
        }
    }
}
```

Kemudian modifikasilah **LatihanServer2.java** menyesuaikan kode di bawah ini

```
/*
 * 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 latihan.server.pkg2;

import java.io.BufferedInputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author U53R
 */
public class LatihanServer2 {
   /**
    * @param args the command line arguments
    */
    public static void main(String[] args) {
        // TODO code application logic here
        int portNumber = 4444;
        try (
            ServerSocket serverSocket = new ServerSocket(portNumber);
            Socket clientSocket = serverSocket.accept();
            PrintWriter out = new PrintWriter(clientSocket.getOutputStream(),
true);
            BufferedInputStream bis = new
BufferedInputStream(clientSocket.getInputStream());
            ObjectInputStream ois = new ObjectInputStream(bis);
        )
        {
            pesan pesan;
            while((pesan = (pesan) ois.readObject())!=null){
                System.out.println("Server Receive: " +
                pesan.toString());
                out.println("Pesan Diterima");
                if(pesan.getPesan().equalsIgnoreCase("exit"))
                    break;
            }
        } catch (IOException ex) {
            Logger.getLogger(LatihanServer2.class.getName()).log(Level.SEVERE, 
null, ex);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(LatihanServer2.class.getName()).log(Level.SEVERE, 
null, ex);
        }
    }
}
```

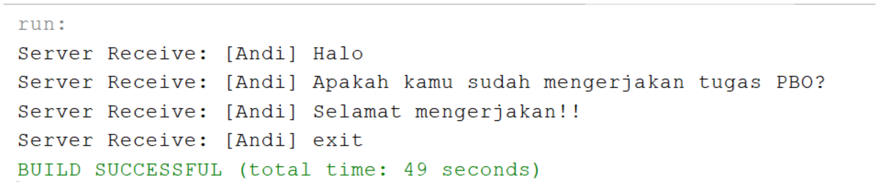
Kemudian jalankan LatihanServer2.java kemudian setelah itu jalankan LatihanClient.java. kita sudah menerapkan object serialization pada pertukaran pesan client dan server. Pesan yang ditukarkan berupa object dari kelas Pesan.

Client(Andi) mengirimkan pesan ke server



```
run:
Halo
Client receive: Pesan Diterima
Apakah kamu sudah mengerjakan tugas PBO?
Client receive: Pesan Diterima
Selamat mengerjakan!!
Client receive: Pesan Diterima
exit
Client receive: Pesan Diterima
BUILD SUCCESSFUL (total time: 44 seconds)
```

Server menerima pesan dari client(Andi)



```
run:
Server Receive: [Andi] Halo
Server Receive: [Andi] Apakah kamu sudah mengerjakan tugas PBO?
Server Receive: [Andi] Selamat mengerjakan!!
Server Receive: [Andi] exit
BUILD SUCCESSFUL (total time: 49 seconds)
```

Supaya server dapat menerima banyak client, maka kita perlu melakukan penyesuaian. Pertama tambahkan new Class bernama **ServerThread.java** sebagai berikut

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to 
change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this 
template
 */
package latihan.server.pkg2;
```

```
/**
 *
 * @author U53R
 */
import java.io.BufferedInputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.PrintWriter;
import java.net.Socket;
import java.util.logging.Level;
import java.util.logging.Logger;

public class ServerThread extends Thread{
    private Socket clientSocket = null;
    public ServerThread(Socket clientSocket) {
        super();
        this.clientSocket = clientSocket;
    }

    @Override
    public void run(){
        try (
            PrintWriter out = new
            PrintWriter(clientSocket.getOutputStream(), true);
            BufferedInputStream bis = new
            BufferedInputStream(clientSocket.getInputStream());
            ObjectInputStream ois = new ObjectInputStream(bis);){
                pesan pesan;
                while((pesan = (pesan) ois.readObject())!=null){
                    System.out.println("Server Receive: " + pesan.toString());
                    out.println("Pesan Diterima");
                    if(pesan.getPesan().equalsIgnoreCase("exit"))
                        break;
                }
        } catch (IOException ex) {
            Logger.getLogger(ServerThread.class.getName()).log(Level.SEVERE,null,
ex);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(ServerThread.class.getName()).log(Level.SEVERE,null,
ex);
        } finally {
            if(clientSocket!=null){
                try {
                    clientSocket.close();
                } catch (IOException ex) {
                    Logger.getLogger(ServerThread.class.getName()).log(Level.SEVE
RE,null, ex);
                }
            }
        }
    }
}
```

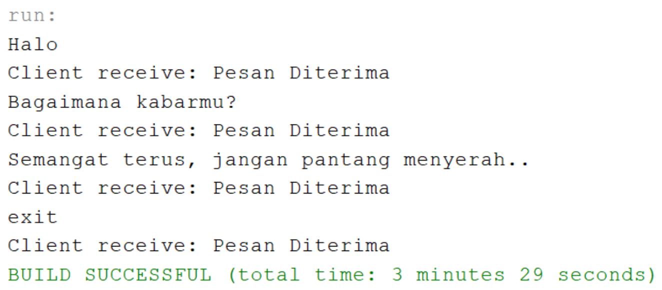
Kemudian modifikasi **LatihanServer2.java** sebagai berikut:

```
/*
 * 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 latihan.server.pkg2;

import java.io.BufferedInputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author U53R
 */
public class LatihanServer2 {
   /**
    * @param args the command line arguments
    */
   public static void main(String[] args) {
        // TODO code application logic here
        int portNumber = 4444;
        boolean listening = true;
        try (
            ServerSocket serverSocket = new
            ServerSocket(portNumber);){
                while(listening){
                    Socket clientSocket = serverSocket.accept();
                    new ServerThread(clientSocket).start();
                }
        } catch (IOException ex) {
            Logger.getLogger(LatihanServer2.class.getName()).log(Level.SEVERE, 
null, ex);
        }
    }
}
```

Tidak ada perubahan pada kode LatihanClient.java namun ketika mau melakukan run file LatihanClient yang kedua, ubahlah nama Andi menjadi Budi supaya nama pengirimnya berbeda.

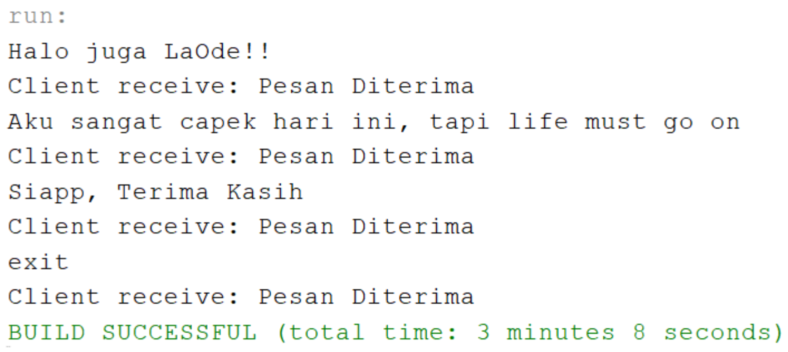
Jalankan LatihanServer.java kemudian Jalankan LatihanClient.java dengan nama LaOde, kemudian jalankan latihanClient.java dengan nama Gazali. Client akan berhenti jika mengetikkan exit sedangkan server masih akan terus menunggu client baru terkoneksi.

Clien (LaOde) mengirim pesan ke server



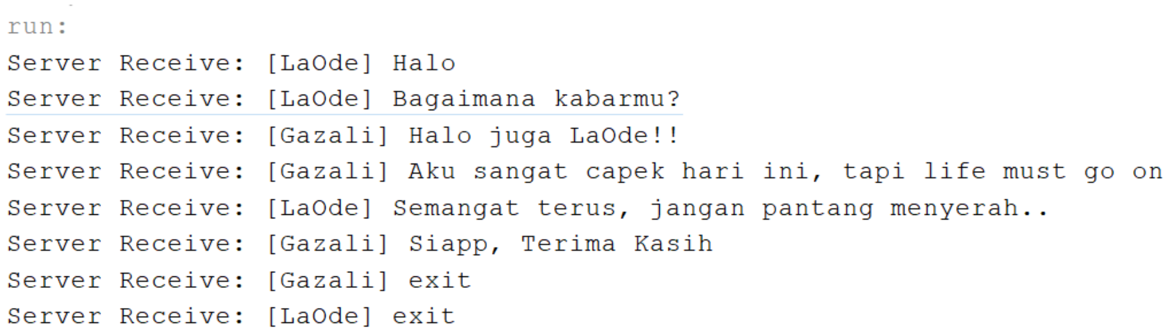
```
run:
Halo
Client receive: Pesan Diterima
Bagaimana kabarmu?
Client receive: Pesan Diterima
Semangat terus, jangan pantang menyerah..
Client receive: Pesan Diterima
exit
Client receive: Pesan Diterima
BUILD SUCCESSFUL (total time: 3 minutes 29 seconds)
```

Clien (Gazali) juga mengirim pesan ke server



```
run:
Halo juga LaOde!!
Client receive: Pesan Diterima
Aku sangat capek hari ini, tapi life must go on
Client receive: Pesan Diterima
Siapp, Terima Kasih
Client receive: Pesan Diterima
exit
Client receive: Pesan Diterima
BUILD SUCCESSFUL (total time: 3 minutes 8 seconds)
```

Selanjutnya server menerima pesan kedua client tersebut



```
run:
Server Receive: [LaOde] Halo
Server Receive: [LaOde] Bagaimana kabarmu?
Server Receive: [Gazali] Halo juga LaOde!!
Server Receive: [Gazali] Aku sangat capek hari ini, tapi life must go on
Server Receive: [LaOde] Semangat terus, jangan pantang menyerah..
Server Receive: [Gazali] Siapp, Terima Kasih
Server Receive: [Gazali] exit
Server Receive: [LaOde] exit
```