

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

TUGAS PRA-PERTEMUAN 13 PEMROGRAMAN BERORIENTASI OBJEK

(Networking Bagian 1)

Penugasan

100 points

Buat chatbot sederhana menggunakan Socket Class.

client: Halo

server: Halo juga

client: apa kabar?

server: kabar baik

dst...

 Class comments

Penyelesaian

Socket adalah sebuah class yang digunakan baik oleh client/server, instance dari class ini adalah nilai balik dari metode `accept()` yang dimiliki oleh class `ServerSocket`. Ketika metode `accept()` sudah dipanggil berapa sudah melakukan binding ke port tertentu dan siap untuk menerima request dari client.

Untuk membuat chatbot, kita perlu membuat program dari sisi client untuk mengirimkan pesan dan juga dari sisi server untuk menanggapi pesan. Berikut program lengkapnya beserta output:

Server.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 chatbot;

/**
 *
 * @author U53R
 */
```

```

*/
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 Server {
    public static void main(String[] args) {
        try {
            ServerSocket serverSocket = new ServerSocket(1234);

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

            Socket clientSocket = serverSocket.accept();
            System.out.println("Klien terhubung: " + clientSocket);

            BufferedReader in = new BufferedReader(new
                InputStreamReader(clientSocket.getInputStream()));
            PrintWriter out = new PrintWriter(clientSocket.getOutputStream(),
true);

            String message;
            while ((message = in.readLine()) != null) {
                System.out.println("Pesan dari klien: " + message);
                String response = generateResponse(message);
                out.println(response);
            }

            in.close();
            out.close();
            clientSocket.close();
            serverSocket.close();

        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    private static String generateResponse(String message){
        message = message.toLowerCase();
        String respons;
        if(message.contains("halo") ) respons = "Halo Juga ";
        else if (message.contains("kabar")) respons = "Kabar Baik";
    }
}

```

```

        else if (message.contains("terima kasih")) respons = "Sama-sama";
        else respons ="Bye,Senang berbincang dengan anda";
        return "Chatbot: "+respons;
    }
}

```

Client.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 chatbot;

/**
 *
 * @author U53R
 */
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;

public class Client {

    public static void main(String[] args) {
        try {
            Socket socket = new Socket("localhost", 1234);

            BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

            BufferedReader userInput = new BufferedReader(new
InputStreamReader(System.in));
            String message;

            while (true) {
                System.out.print("Anda: ");
                message = userInput.readLine();
                out.println(message);
                String response = in.readLine();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

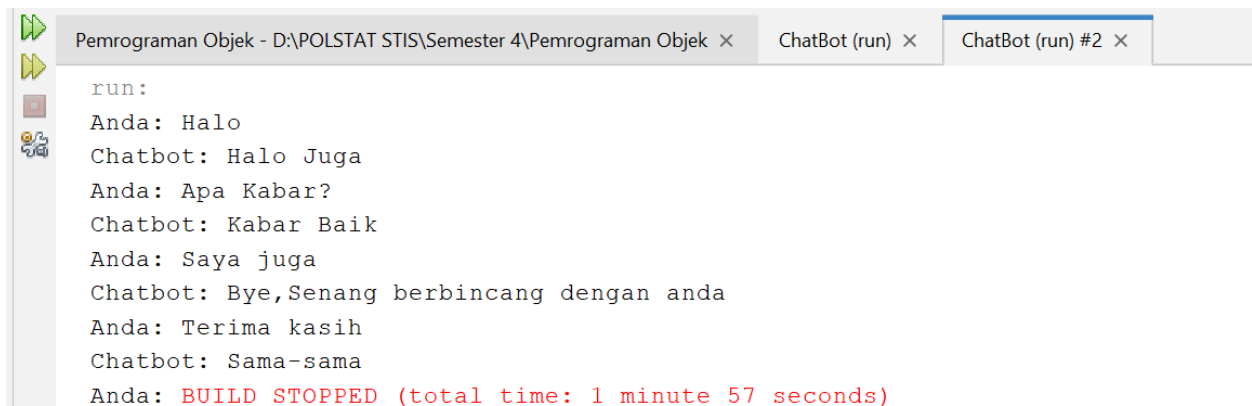
```

        System.out.println(response);
    }

    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

Simulasi percakapan pendek dengan Chat Bot



Pada dasarnya Class Server digunakan untuk merepresentasikan sebuah fungsi server, skenario dari class tersebut adalah ketika client mengirimkan pesan “hello server” akan dibalas oleh server dengan pesan “hello client”, dst. `PrintWriter` digunakan untuk mengirimkan/menulis pesan ke client, sedangkan `BufferedReader` digunakan untuk menerima/membaca pesan dari client.