

PEMROGRAMAN JARINGAN KOMPUTER

TUGAS 12

DOSEN PENGAJAR : NOPRIANTO, S. Kom., M.Eng.



OLEH:
NABILAH ROISUL AMINI (1731710116)
MI – 3A

D3 MANAJEMEN INFORMATIKA
TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2019

TUGAS 12

1. Pertama, buat class **EchoServer** untuk menjalankan server

```
6      package pemrograman.jaringan.tugas12;
7
8      import java.io.BufferedReader;
9      import java.io.IOException;
10     import java.io.InputStreamReader;
11     import java.io.PrintWriter;
12     import java.net.ServerSocket;
13     import java.net.Socket;
14     import java.util.logging.Level;
15     import java.util.logging.Logger;
16
17     /**
18      *
19      * @author asus
20      */
21     public class EchoServer {
22
23         private ServerSocket serverSocket;
24         private Socket clieSocket;
25         private PrintWriter out;
26         private BufferedReader in;
27
28         public void start(int port) {
29             try {
30                 serverSocket = new ServerSocket(port);
31                 clieSocket = serverSocket.accept();
32                 out = new PrintWriter(clieSocket.getOutputStream(), true);
33                 in = new BufferedReader(new InputStreamReader(clieSocket.getInputStream()));
34                 String inputLine;
35                 while ((inputLine = in.readLine()) != null) {
36                     out.println(inputLine);
37                 }
38             } catch (IOException ex) {
39                 Logger.getLogger(Client.class.getName()).log(Level.SEVERE, null, ex);
40             }
41         }
42
43         public void stop() {
44             try {
45                 in.close();
46                 out.close();
47                 serverSocket.close();
48             } catch (IOException ex) {
49                 Logger.getLogger(Client.class.getName()).log(Level.SEVERE, null, ex);
50             }
51         }
52
53         public static void main(String[] args) {
54             EchoServer server = new EchoServer();
55             server.start(6666);
56         }
57     }
```

2. Membuat class Client

```
6 package pemrograman.jaringan.tugas12;
7
8 import java.io.BufferedReader;
9 import java.io.IOException;
10 import java.io.InputStreamReader;
11 import java.io.PrintWriter;
12 import java.net.Socket;
13 import java.util.logging.Level;
14 import java.util.logging.Logger;
15
16 /**
17  *
18  * @author asus
19  */
20 public class Client {
21
22     private Socket clientSocket;
23     private PrintWriter out;
24     private BufferedReader in;
25
26     public void startConnection(String ip, int port) {
27         try {
28             clientSocket = new Socket(ip, port);
29             out = new PrintWriter(clientSocket.getOutputStream(), true);
30             in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
31         } catch (IOException ex) {
32             Logger.getLogger(Client.class.getName()).log(Level.SEVERE, null, ex);
33         }
34     }
35
36     public String sendMessage(String msg) {
37         String resp = null;
38         try {
39             out.println(msg);
40             resp = in.readLine();
41         } catch (IOException ex) {
42             Logger.getLogger(Client.class.getName()).log(Level.SEVERE, null, ex);
43         }
44         return resp;
45     }
46
47     public void stopConnection() {
48         try {
49             in.close();
50             out.close();
51             clientSocket.close();
52         } catch (IOException ex) {
53             Logger.getLogger(Client.class.getName()).log(Level.SEVERE, null, ex);
54         }
55     }
56 }
```

3. Buat class untuk tampilan port gui.

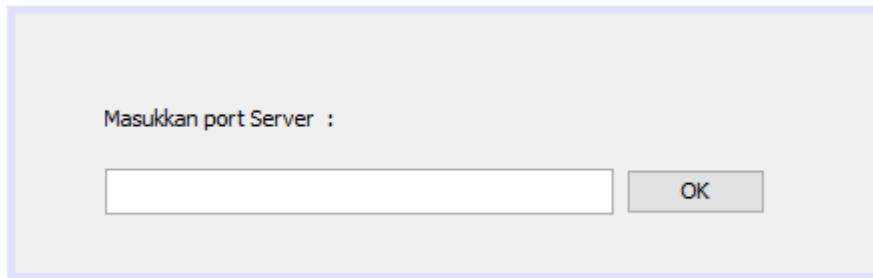
```
6 package pemrograman.jaringan.tugas12;
7
8 import javax.swing.JButton;
9 import javax.swing.JTextField;
10
```

```

110 public JButton getOk() {
111     return Ok;
112 }
113
114 public JTextField getTextPort() {
115     return TextPort;
116 }

```

Tampilan port gui

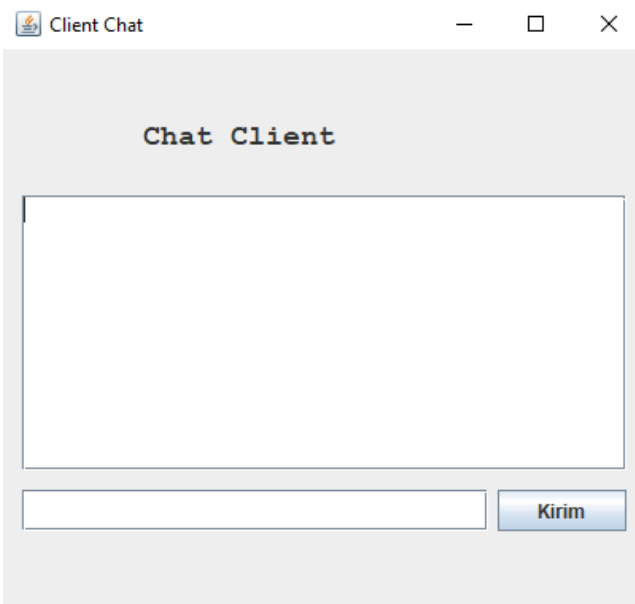


4. Buat class untuk tampilan pesan di gui

```

6 package pemrograman.jaringan.tugas12;
7
8 import javax.swing.JButton;
9 import javax.swing.JTextArea;
10 import javax.swing.JTextField;
11
120 public JButton getKirimButton() {
121     return Kirim;
122 }
123
124 public JTextArea getTextArea() {
125     return Text;
126 }
127
128 public JTextField getTextField() {
129     return textField;
130 }

```



5. Buat class ClientRun sebagai class utamanya

```
6 package pemrograman.jaringan.tugas12;
7
8 import java.awt.event.ActionEvent;
9 import java.awt.event.ActionListener;
10 import java.text.SimpleDateFormat;
11 import java.util.Date;
12
13 /**
14  *
15  * @author asus
16  */
17 public class ClientRun implements Runnable {
18
19     private Client client;
20     private port portView;
21     private pesan view;
22     private String mess = "";
23
24     public ClientRun() {
25         this.client = new Client();
26         this.portView = new port();
27         this.view = new pesan();
28         this.portView.setTitle("Port Input");
29         this.portView.setVisible(true);
30
31         this.portView.getOk().addActionListener(new ActionListener() {
32             @Override
33             public void actionPerformed(ActionEvent e) {
34                 if (String.valueOf(portView.getTextPort().getText()).equals("6666")) {
35                     client.startConnection("127.0.0.1", Integer.valueOf(portView.getTextPort().getText()));
36                     portView.setVisible(false);
37                     view.setTitle("Client Chat");
38                     view.setVisible(true);
39                 } else {
40                     portView.getTextPort().setText("");
41                 }
42             }
43         });
44         this.view.getKirimButton().addActionListener(new ActionListener() {
45             @Override
46             public void actionPerformed(ActionEvent e) {
47                 mess += time() + "-> " + String.valueOf(view.getTextField().getText()) + "\n";
48                 String response = client.sendMessage(time() + "<- " + String.valueOf(view.getTextField().getText()));
49                 mess += response + "\n";
50             }
51         });
52     }
53 }
```

```

53     public String time() {
54         Date date = new Date();
55         SimpleDateFormat simpleDateFormat = new SimpleDateFormat("HH:mm:ss.SSS");
56         String time = simpleDateFormat.format(date);
57         return time;
58     }
59
60     @Override
61     public void run() {
62         do {
63             if (this.view.getTextArea().getText().equals(mess) == false) {
64                 this.view.getTextArea().setText(mess);
65             }
66         } while (true);
67     }
68
69     public static void main(String[] args) {
70         new Thread(new ClientRun()).start();
71     }
72 }

```

6. Jalankan class EchoServer, kemudian jalankan class ClientRun

