PEMROGRAMAN JARINGAN KOMPUTER

Tugas Chat Client Server Socket

Dosen Pengampu : Noprianto S.Kom, M.Eng

Oleh:

Kresniawati 1731710118

MI - 3A

PROGRAM STUDI D-III MANAJEMEN INFORMATIKA

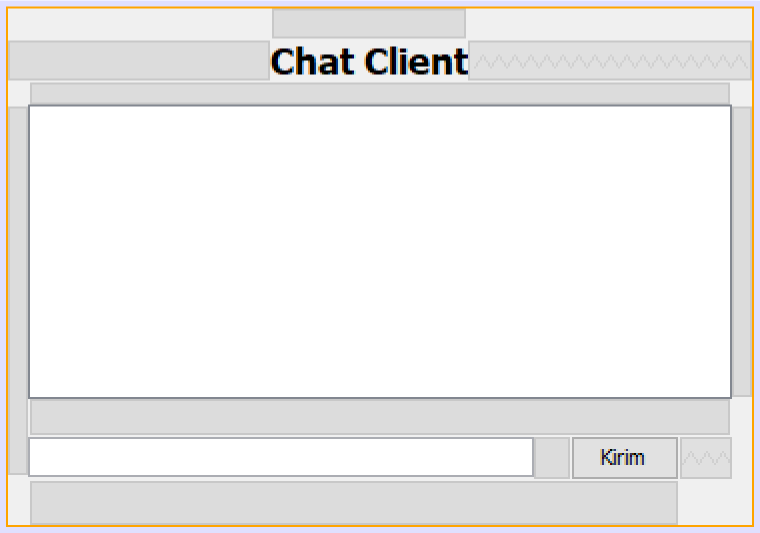
JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

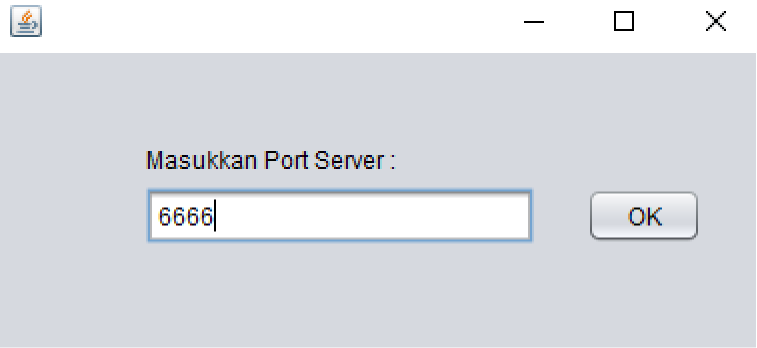
2019

Tugas

1. Buat desain form untuk chat client server seperti di bawah ini :



2. Buat desain form untuk port server seperti di bawah ini jangan lupa tambahkan getter untuk text field dan button ok.



Masukkan Port Server :

6666

3. Buat Class Server untuk menjalankan server dengan mengisi nomor port 6666

```
public class Server {
    private ServerSocket serverSocket;
    private Socket clieSocket;
    private PrintWriter out;
    private BufferedReader in;
    public void start(int port){
        try {
            serverSocket = new ServerSocket(port);
            clieSocket = serverSocket.accept();
            out = new PrintWriter(clieSocket.getOutputStream(), true);
            in = new BufferedReader(new InputStreamReader(clieSocket.getInputStream()));

            String inputLine;

            while((inputLine = in.readLine()) != null){
                out.println(inputLine);
            }

        } catch (IOException ex) {
            Logger.getLogger(Client.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    public void stop(){
        try {
            in.close();
            out.close();
            serverSocket.close();
        } catch (IOException ex) {
            Logger.getLogger(Client.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    public static void main(String[] args) {
        EchoServer server = new EchoServer();
        server.start(6666);
    }
}
```

4. Buat Class Client

```
public class Client {
    private Socket clientSocket;
    private PrintWriter out;
    private BufferedReader in;

    public void startConnection(String ip, int port){
        try {
```

```
      clientSocket = new Socket(ip, port);
      out = new PrintWriter(clientSocket.getOutputStream(), true);
      in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
    } catch (IOException ex) {
      Logger.getLogger(Client.class.getName()).log(Level.SEVERE, null, ex);
    }
  }

  public String sendMessage(String msg){
    String resp = null;
    try {
      out.println(msg);
      resp = in.readLine();
    } catch (IOException ex) {
      Logger.getLogger(Client.class.getName()).log(Level.SEVERE, null, ex);
    }
    return resp;
  }

  public void stopConnection(){
    try {
      in.close();
      out.close();
      clientSocket.close();
    } catch (IOException ex) {
      Logger.getLogger(Client.class.getName()).log(Level.SEVERE, null, ex);
    }
  }

}
```

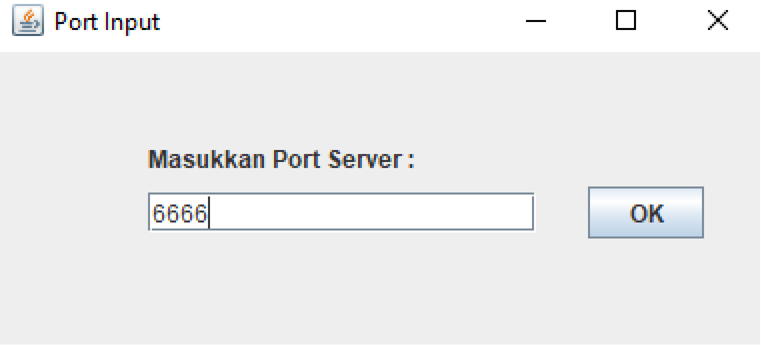
5. ClientRun.java yang mana akan digunakan sebagai class utama

```
public class ClientRun implements Runnable{
  private Client client;
  private portGui portView;
  private PesanGui view;
  private String mess = "";

  public ClientRun(){
    this.client = new Client();
```

```
        this.portView = new portGui();
        this.view = new PesanGui();
        this.portView.setTitle("Port Input");
        this.portView.setVisible(true);

        this.portView.getPortOk().addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                if(String.valueOf(portView.getPortText().getText()).equals("6666")){
                    client.startConnection("127.0.0.1",
Integer.valueOf(portView.getPortText().getText()));
                    portView.setVisible(false);
                    view.setTitle("Client Chat");
                    view.setVisible(true);
                }else{
                    portView.getPortText().setText("");
                }
            }
        });

        this.view.getKirimButton().addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                mess += time()+"-> "+String.valueOf(view.getTextField().getText()+"\n");
                String response = client.sendMessage(time()+"<- "+
String.valueOf(view.getTextField().getText()));
                mess += response + "\n";
            }
        });
    }

    public String time(){
        Date date = new Date();
        SimpleDateFormat simpleDateFormat = new SimpleDateFormat("HH:mm:ss.SSS");
        String time = simpleDateFormat.format(date);
        return time;
    }

    @Override
    public void run() {
        do{
            if(this.view.getTextArea().getText().equals(mess)==false){
                this.view.getTextArea().setText(mess);
            }
        }while(true);
    }

    public static void main(String[] args) {
        new Thread(new ClientRun()).start();
    }


}
```

6. jalankan class EchoServer lalu jalankan client run

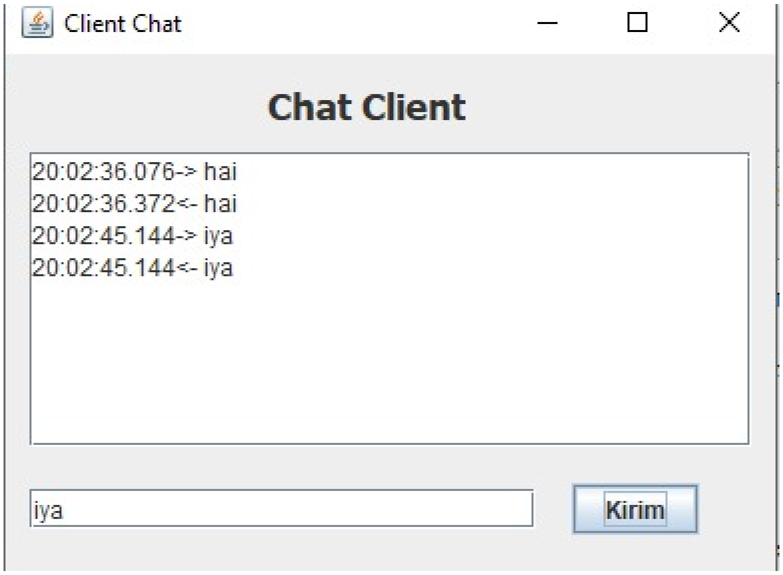
   - Masukkan port 666



Masukkan Port Server :

6666

   - Output



```
20:02:36.076-> hai
20:02:36.372<- hai
20:02:45.144-> iya
20:02:45.144<- iya
```

iya