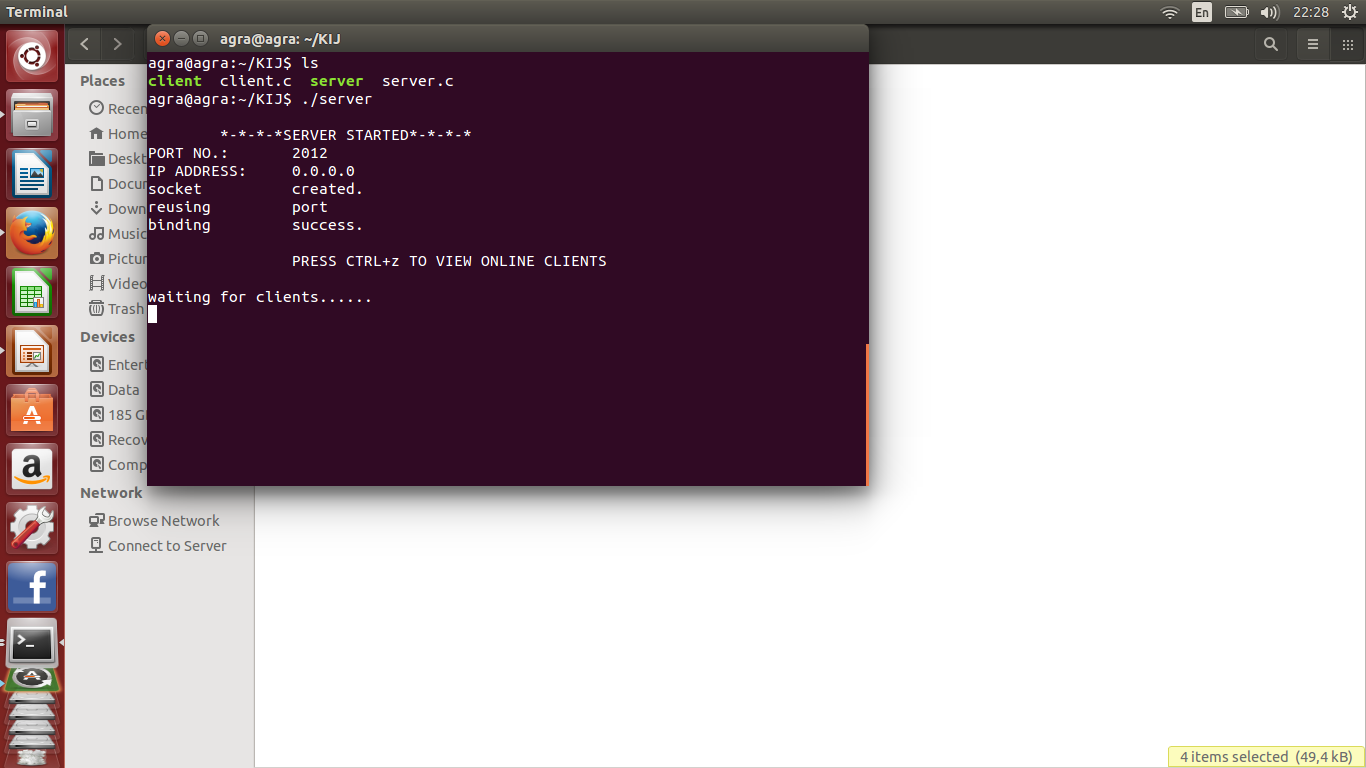


Client memasukkan usernamenya

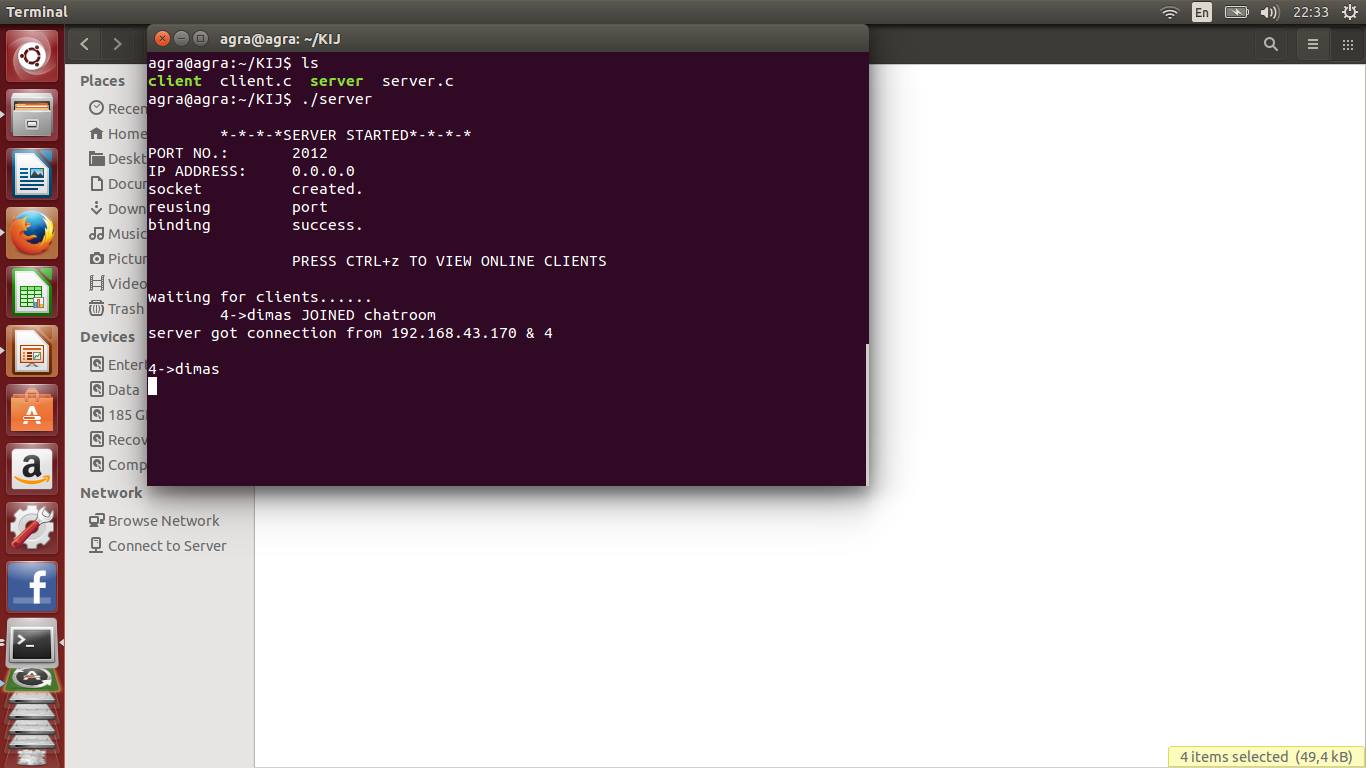
Kode client

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package kijchat;  import java.net.\*;  import java.io.\*;  import java.util.\*;  /\*\*  \*  \* @author dimas  \*/  public class FrameKIJChat extends javax.swing.JFrame {  boolean isConnected = false;  String username, computername, serverIP = "10.151.34.156"; // set server IP  Socket sock;  int Port = 2012; // set client port  BufferedReader reader;  PrintWriter writer;  ArrayList<String> userList = new ArrayList();  /\*\*  \* Creates new form FrameKIJChat  \*/  public FrameKIJChat() {  initComponents();  }  public class IncomingReader implements Runnable{    public void run(){  String data[];  String stream, connect = "Con", done = "Done";  try{  while((stream = reader.readLine())!=null){  data = stream.split(":");  if(data[0].equals(connect)){  chatTextArea.removeAll();  userAdd(data[1]);  }else if(data[0].equals(done)){  onlineTextArea.setText("");  writeUsers();  userList.clear();  }  }  }catch(Exception ex){    }  }    }      public void ListenThread(){  Thread IncomingReader = new Thread(new IncomingReader());  IncomingReader.start();  }    public void userAdd(String data){  userList.add(data);  }    public void writeUsers(){  String[] tempList = new String[(userList.size())];  userList.toArray(tempList);  for(String token:tempList){  onlineTextArea.append(token + "\n");  }  }  /\*\*  \* This method is called from within the constructor to initialize the form.  \* WARNING: Do NOT modify this code. The content of this method is always  \* regenerated by the Form Editor.  \*/  @SuppressWarnings("unchecked")  // <editor-fold defaultstate="collapsed" desc="Generated Code">  private void initComponents() {  jLabel1 = new javax.swing.JLabel();  usernameField = new javax.swing.JTextField();  connectButton = new javax.swing.JButton();  disconnectButton = new javax.swing.JButton();  jLabel2 = new javax.swing.JLabel();  jScrollPane1 = new javax.swing.JScrollPane();  chatTextArea = new javax.swing.JTextArea();  jScrollPane2 = new javax.swing.JScrollPane();  onlineTextArea = new javax.swing.JTextArea();  jLabel3 = new javax.swing.JLabel();  sendUserArea = new javax.swing.JTextField();  jScrollPane3 = new javax.swing.JScrollPane();  sendTextArea = new javax.swing.JTextArea();  sendButton = new javax.swing.JButton();  setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);  jLabel1.setFont(new java.awt.Font("Verdana", 0, 12)); // NOI18N  jLabel1.setText("Username");  connectButton.setText("Connect");  connectButton.addActionListener(new java.awt.event.ActionListener() {  public void actionPerformed(java.awt.event.ActionEvent evt) {  connectButtonActionPerformed(evt);  }  });  disconnectButton.setText("Disconnect");  jLabel2.setFont(new java.awt.Font("Verdana", 1, 12)); // NOI18N  jLabel2.setText("Friend Conversation");  chatTextArea.setEditable(false);  chatTextArea.setColumns(20);  chatTextArea.setLineWrap(true);  chatTextArea.setRows(5);  jScrollPane1.setViewportView(chatTextArea);  onlineTextArea.setEditable(false);  onlineTextArea.setColumns(20);  onlineTextArea.setLineWrap(true);  onlineTextArea.setRows(5);  jScrollPane2.setViewportView(onlineTextArea);  jLabel3.setFont(new java.awt.Font("Verdana", 0, 12)); // NOI18N  jLabel3.setText("Send To :");  sendTextArea.setColumns(20);  sendTextArea.setRows(5);  jScrollPane3.setViewportView(sendTextArea);  sendButton.setFont(new java.awt.Font("Verdana", 0, 12)); // NOI18N  sendButton.setText("Send");  javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());  getContentPane().setLayout(layout);  layout.setHorizontalGroup(  layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()  .addContainerGap()  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addGroup(layout.createSequentialGroup()  .addComponent(jLabel1)  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  .addComponent(usernameField, javax.swing.GroupLayout.PREFERRED\_SIZE, 165, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  .addComponent(connectButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 87, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  .addComponent(disconnectButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)  .addGap(35, 35, 35)  .addComponent(jLabel2)  .addGap(32, 32, 32))  .addGroup(layout.createSequentialGroup()  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addComponent(jScrollPane1)  .addGroup(layout.createSequentialGroup()  .addComponent(jLabel3)  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  .addComponent(sendUserArea))  .addGroup(layout.createSequentialGroup()  .addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED\_SIZE, 328, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  .addComponent(sendButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)))  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, 174, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addContainerGap())))  );  layout.setVerticalGroup(  layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addGroup(layout.createSequentialGroup()  .addContainerGap()  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)  .addComponent(jLabel1)  .addComponent(usernameField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addComponent(connectButton)  .addComponent(disconnectButton)  .addComponent(jLabel2))  .addGap(18, 18, 18)  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addComponent(jScrollPane2)  .addGroup(layout.createSequentialGroup()  .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 231, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)  .addComponent(jLabel3)  .addComponent(sendUserArea, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)  .addComponent(jScrollPane3, javax.swing.GroupLayout.DEFAULT\_SIZE, 101, Short.MAX\_VALUE)  .addComponent(sendButton, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))  .addGap(0, 0, Short.MAX\_VALUE)))  .addContainerGap())  );  pack();  }// </editor-fold>  private void connectButtonActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  if(isConnected == false){  username=usernameField.getText();  usernameField.setEditable(false);  try{  computername=Inet4Address.getLocalHost().getHostAddress();  }catch(Exception ex){  computername="1235";  }  try{  sock = new Socket(serverIP, Port);  InputStreamReader streamreader = new InputStreamReader(sock.getInputStream());  reader = new BufferedReader(streamreader);  writer = new PrintWriter(sock.getOutputStream());  writer.println("Con:"+username+":has connected.:"+computername);  writer.flush();  isConnected = true;  ListenThread();  }catch(Exception ex){  chatTextArea.append("Cannot connect!. Try Again \n");  usernameField.setEditable(true);  }  }  }  /\*\*  \* @param args the command line arguments  \*/  public static void main(String args[]) {  /\* Set the Nimbus look and feel \*/  //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">  /\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.  \* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html  \*/  try {  for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {  if ("Nimbus".equals(info.getName())) {  javax.swing.UIManager.setLookAndFeel(info.getClassName());  break;  }  }  } catch (ClassNotFoundException ex) {  java.util.logging.Logger.getLogger(FrameKIJChat.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  } catch (InstantiationException ex) {  java.util.logging.Logger.getLogger(FrameKIJChat.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  } catch (IllegalAccessException ex) {  java.util.logging.Logger.getLogger(FrameKIJChat.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  } catch (javax.swing.UnsupportedLookAndFeelException ex) {  java.util.logging.Logger.getLogger(FrameKIJChat.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  }  //</editor-fold>  /\* Create and display the form \*/  java.awt.EventQueue.invokeLater(new Runnable() {  public void run() {  new FrameKIJChat().setVisible(true);  }  });  }  // Variables declaration - do not modify  private javax.swing.JTextArea chatTextArea;  private javax.swing.JButton connectButton;  private javax.swing.JButton disconnectButton;  private javax.swing.JLabel jLabel1;  private javax.swing.JLabel jLabel2;  private javax.swing.JLabel jLabel3;  private javax.swing.JScrollPane jScrollPane1;  private javax.swing.JScrollPane jScrollPane2;  private javax.swing.JScrollPane jScrollPane3;  private javax.swing.JTextArea onlineTextArea;  private javax.swing.JButton sendButton;  private javax.swing.JTextArea sendTextArea;  private javax.swing.JTextField sendUserArea;  private javax.swing.JTextField usernameField;  // End of variables declaration  } |

Server :



Server listening to client



Server menerima koneksi dari client