Laporan Hasil Praktikum

Pemrograman Desktop

Tugas 14

ALYA AIMAN SALSABILA ARIF

1817101379

Tingkat III Rekayasa Perangkat Lunak Kripto

Politeknik Siber dan Sandi Negara
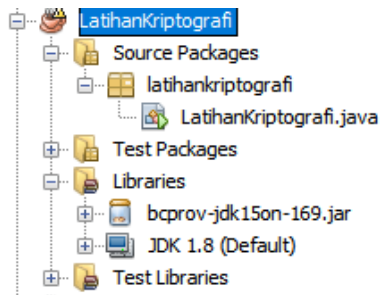
2020/2021

Daftar Isi

Buat projek baru dengan nama LatihanKriptografi. Tambahkan *library* yang sudah diunduh

Masukkan *source code* berikut pada kelas LatihanKriptografi

```
46          byte[] cipherText = new byte[cipher.getOutputSize(input.length)];
47          int ctLength = cipher.update(input, 0, input.length, cipherText, 0);
48          ctLength += cipher.doFinal(cipherText, ctLength);
49          System.out.println(new String(cipherText));
50          System.out.println(ctLength);
51
52          // decryption pass
53          cipher.init(Cipher.DECRYPT_MODE, key);
54          byte[] plainText = new byte[cipher.getOutputSize(ctLength)];
55          int ptLength = cipher.update(cipherText, 0, ctLength, plainText, 0);
56          ptLength += cipher.doFinal(plainText, ptLength);
57          System.out.println(new String(plainText));
58          System.out.println(ptLength);
59      }
60
61  }
```

Jalankan program. Berikut hasil yang ditampilkan

```
Output - LatihanKriptografi (run)
    run:
    Alya Aiman
    (□z□lj□>□□□□A□□
    16
    Alya Aiman
    10
    BUILD SUCCESSFUL (total time: 10 seconds)
```

2. Studi Kasus

Buat basis data baru dengan nama stokbarang dan tabel dengan nama stokbarang dan user

```
MariaDB [(none)]> create database stokbarang;
Query OK, 1 row affected (0.003 sec)

MariaDB [(none)]> use stokbarang;
Database changed
MariaDB [stokbarang]> create table stokbarang (kode int(10) not null primary key a
uto_increment,
    -> nama varchar(25) not null,
    -> jumlah int(10) not null);
Query OK, 0 rows affected (0.676 sec)

MariaDB [stokbarang]> create table user (id int(11) not null primary key auto_incr
ement,
    -> username varchar(30) not null,
    -> password varchar(30) not null);
Query OK, 0 rows affected (0.223 sec)
```
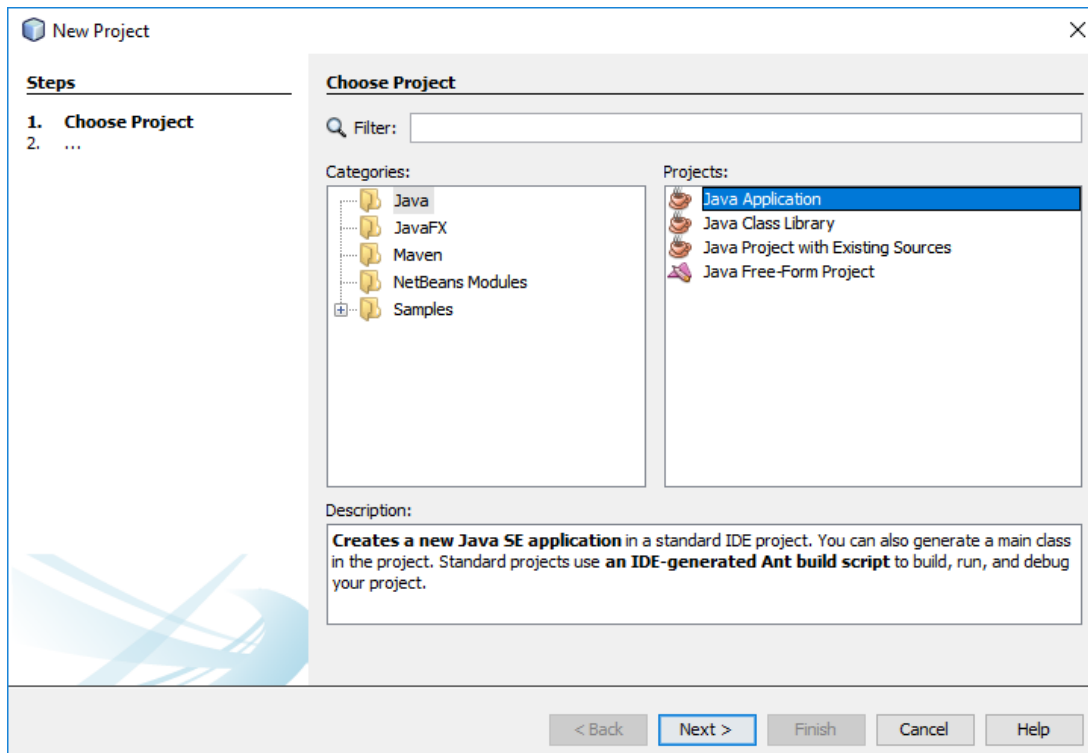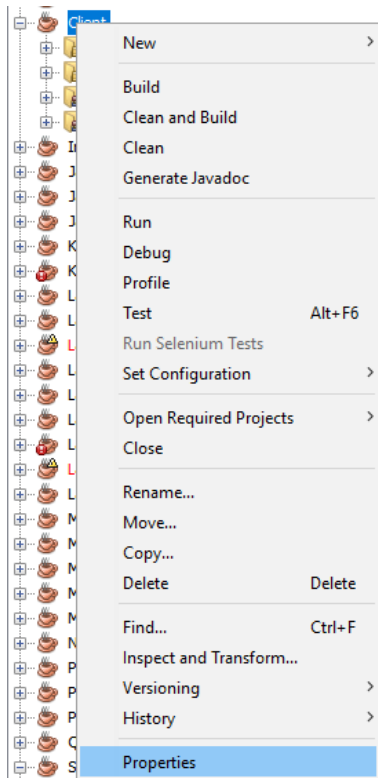
Buat proyek *Java Class Library* dengan nama Komunikasi.api sebagai penghubung antara *client* dan *server*
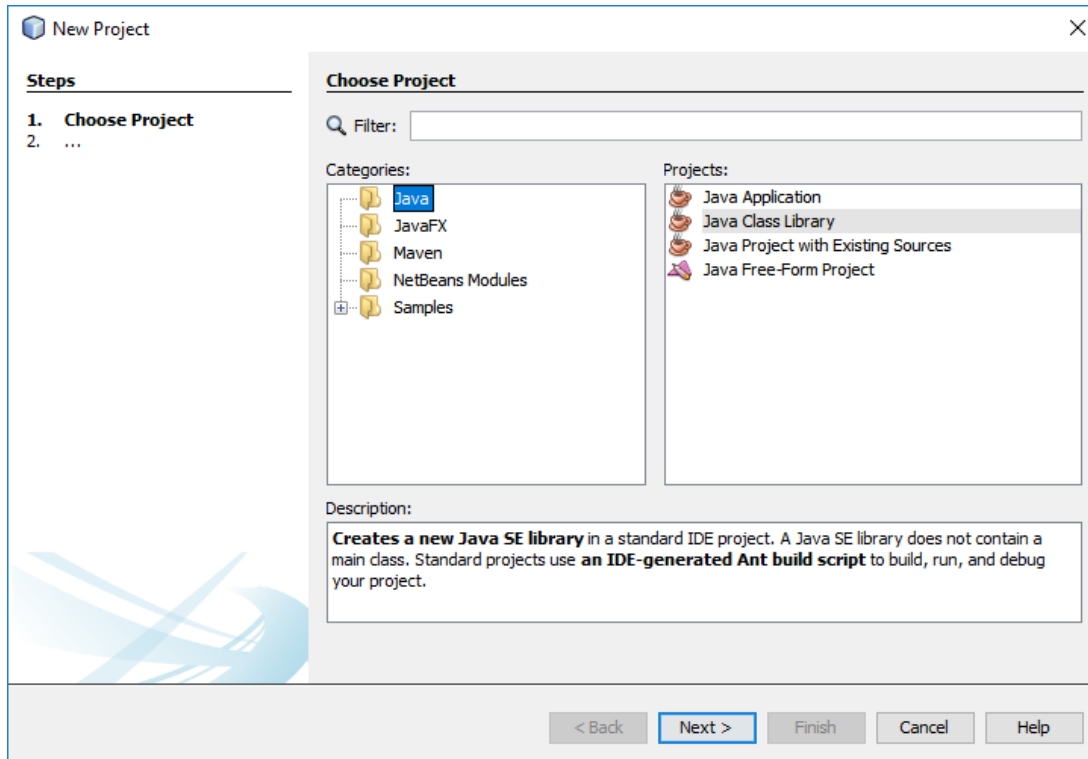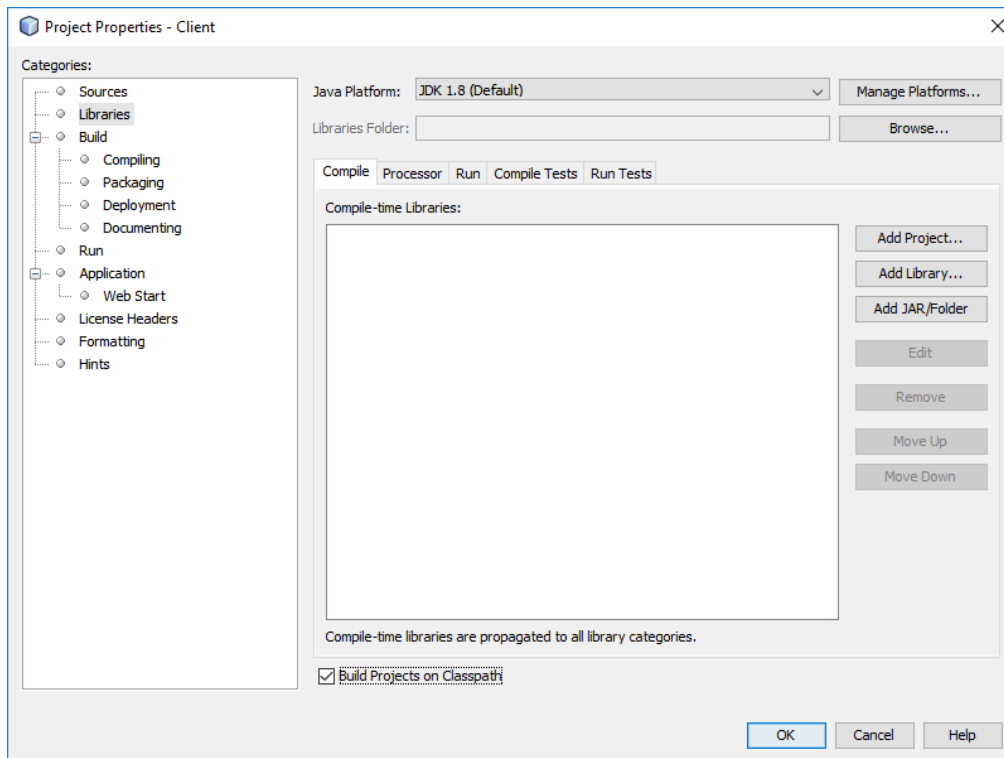

Buat proyek *Java Application* dengan nama Server dan Client
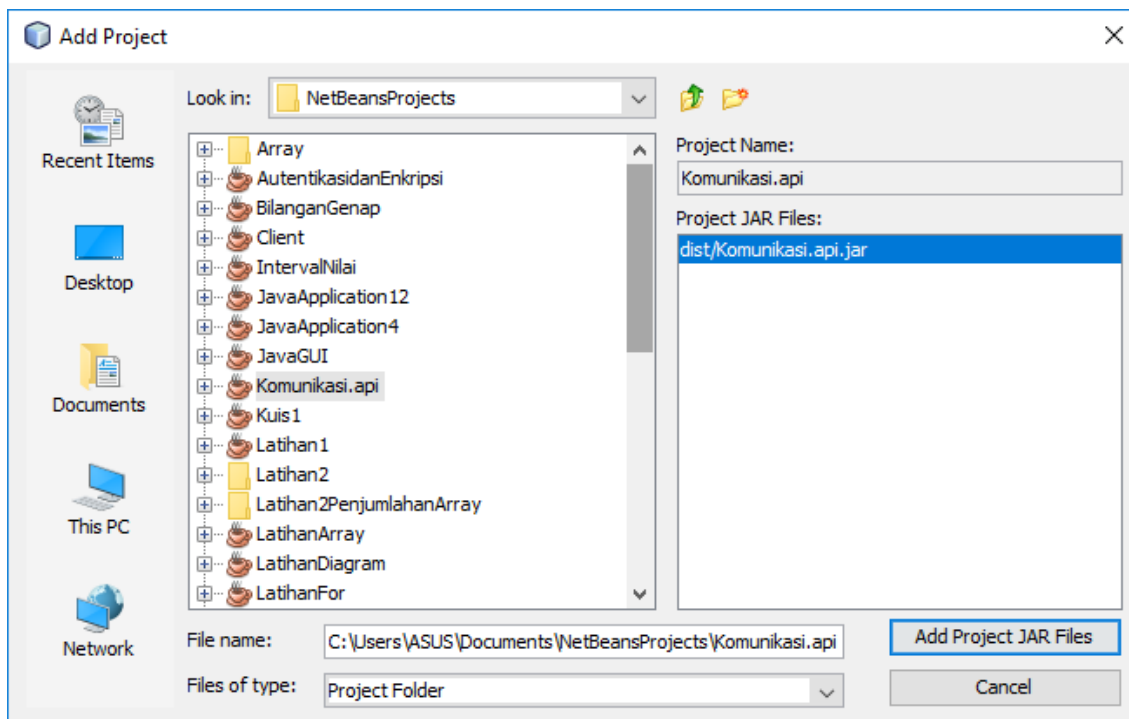
Untuk menghubungkan antara *client* dan *server* klik kanan pada *project* Client kemudian pilih *Properties* pada proyek Server dan Client
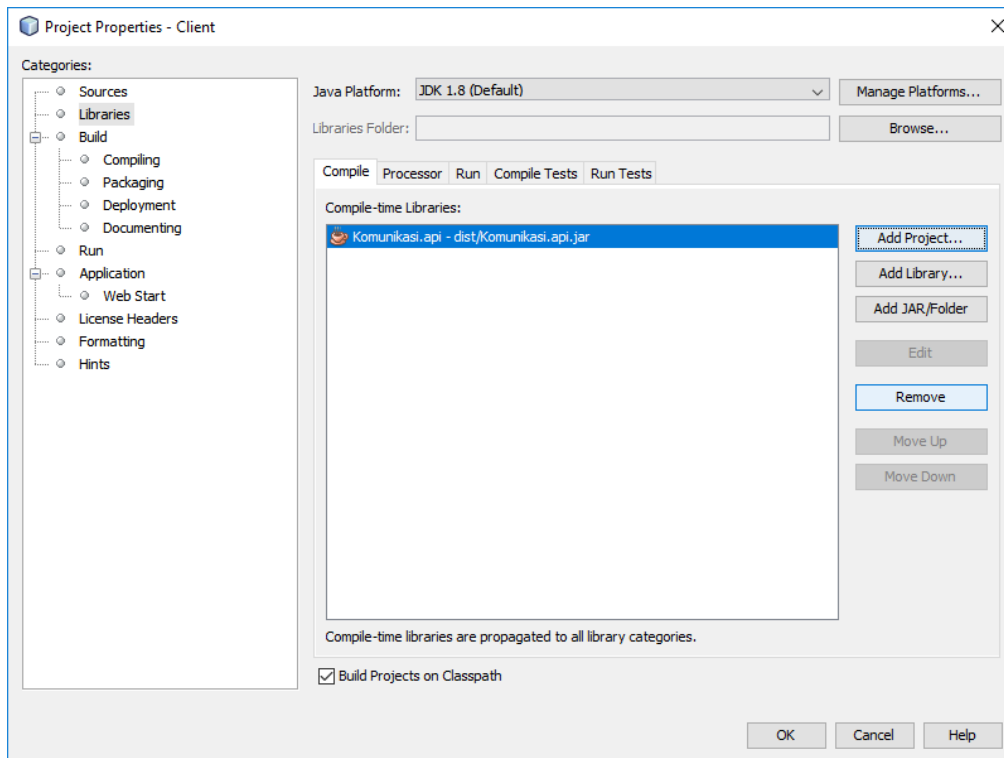
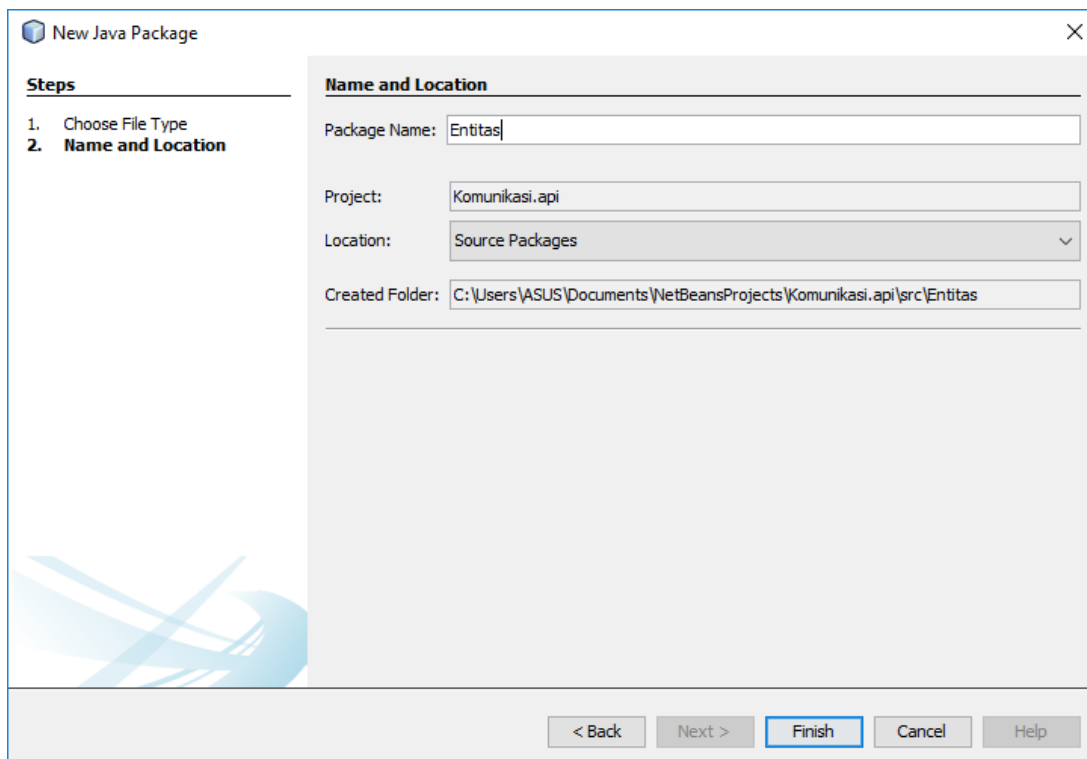Berikut tampilan yang dihasilkan. Pilih *Libraries -> Add Project*

Pilih *project* Komunikasi.api yang sudah dibuat sebelumnya kemudian pilih *Add Project JAR Files*
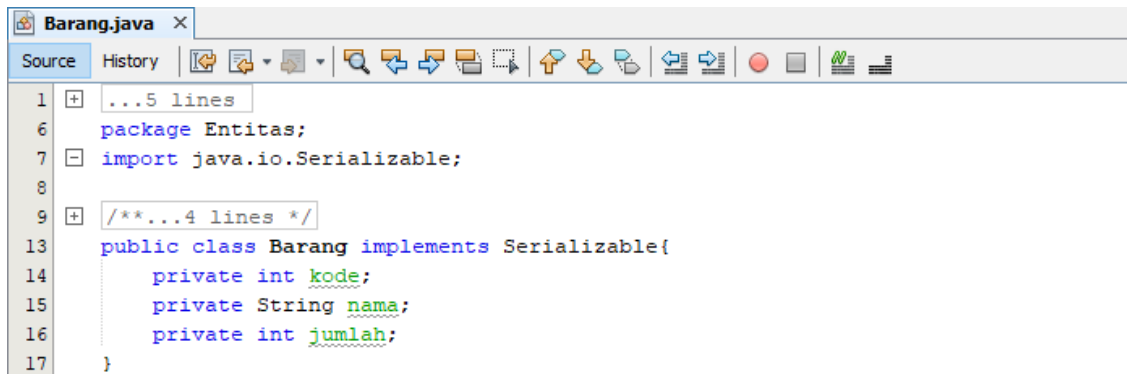
Klik *OK*

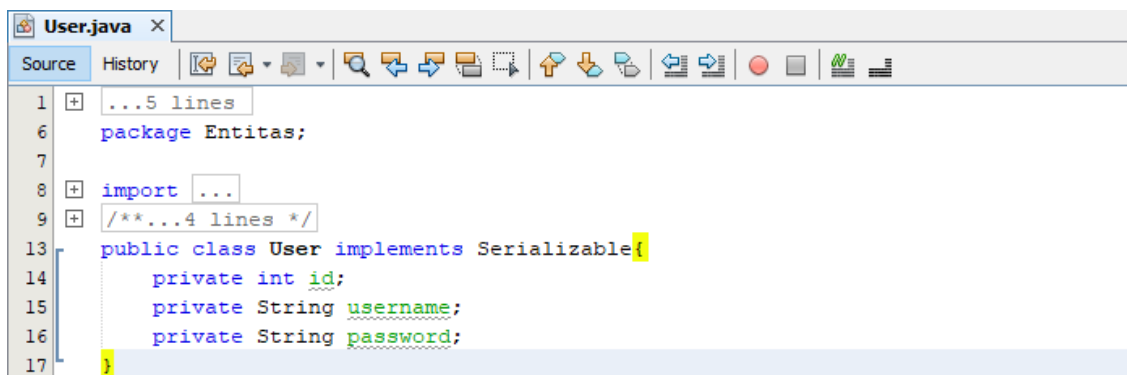Buat *package* dengan nama Entitas pada proyek Komunikasi.api

Buat kelas Barang dan masukkan *source code* berikut pada kelas Barang


```
1 ⊞   ...5 lines
6     package Entitas;
7 ⊟   import java.io.Serializable;
8
9 ⊞   /**...4 lines */
13    public class Barang implements Serializable{
14        private int kode;
15        private String nama;
16        private int jumlah;
17    }
```

Buat kelas User dan masukkan *source code* berikut pada kelas User
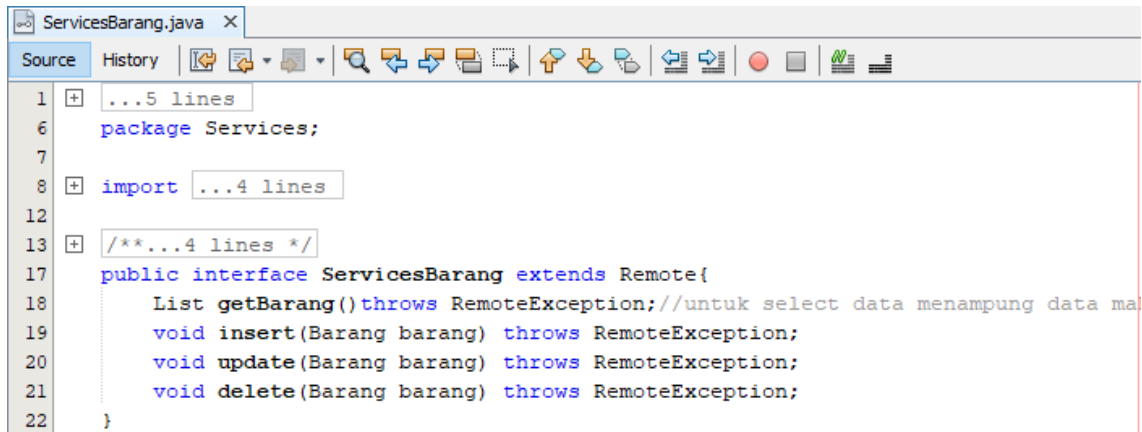

```
1 ⊞   ...5 lines
6     package Entitas;
7
8 ⊞   import ...
9 ⊞   /**...4 lines */
13    public class User implements Serializable{
14        private int id;
15        private String username;
16        private String password;
17    }
```

Klik kanan pada dua kelas tersebut kemudian pilih *Insert Code*. Kemudian pilih *Getter and Setter*

Navigate	>
Show Javadoc	Alt+F1
Find Usages	Alt+F7
Call Hierarchy	
Insert Code...	Alt+Insert
Fix Imports	Ctrl+Shift+I
Refactor	>
Format	Alt+Shift+F
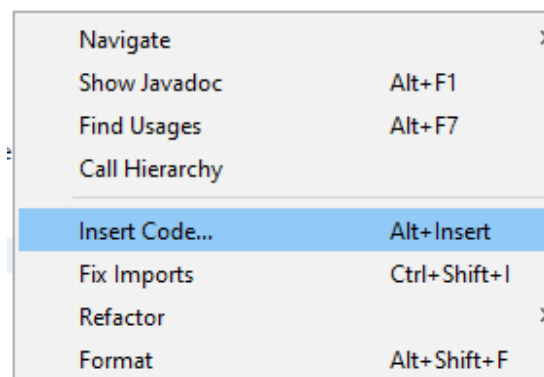
Pada *project* Komunikasi.api, buat *package* baru dengan nama Services. Buat *Interface* ServicesBarang dan ServicesUser. Masukkan *source code* berikut pada *interface* ServicesBarang

```java
package Services;

import ...4 lines

/**...4 lines */
public interface ServicesBarang extends Remote{
    List getBarang()throws RemoteException;//untuk select data menampung data mal
    void insert(Barang barang) throws RemoteException;
    void update(Barang barang) throws RemoteException;
    void delete(Barang barang) throws RemoteException;
}
```

Masukkan *source code* berikut pada *interface* ServicesUser

```java
package Services;

import ...4 lines

/**...4 lines */
public interface ServicesUser extends Remote{
    List getUser()throws RemoteException;
    void insert(User user) throws RemoteException;
}
```

9

Buat *package* dengan nama Koneksi pada proyek Server dan buat kelas Koneksi pada *package* tersebut. Masukkan *source code* berikut pada kelas Koneksi
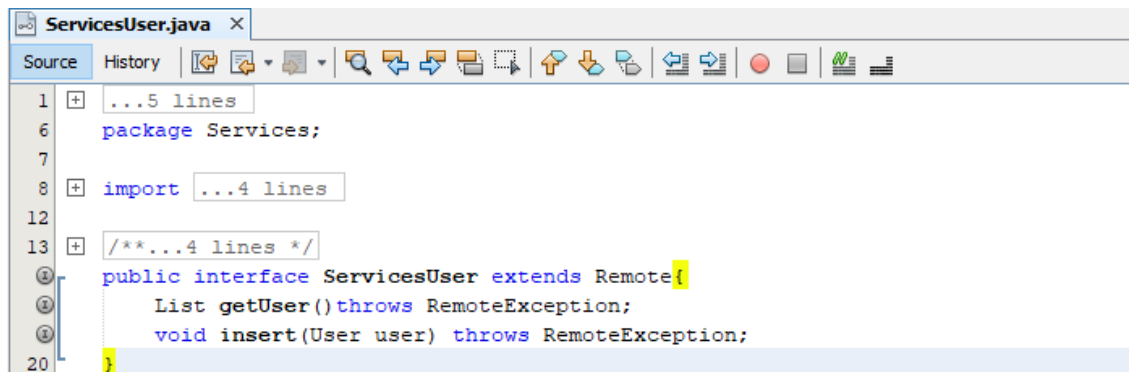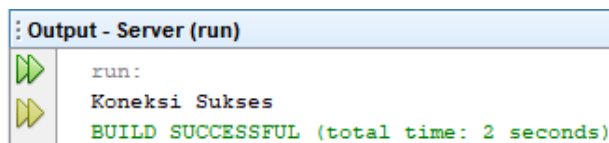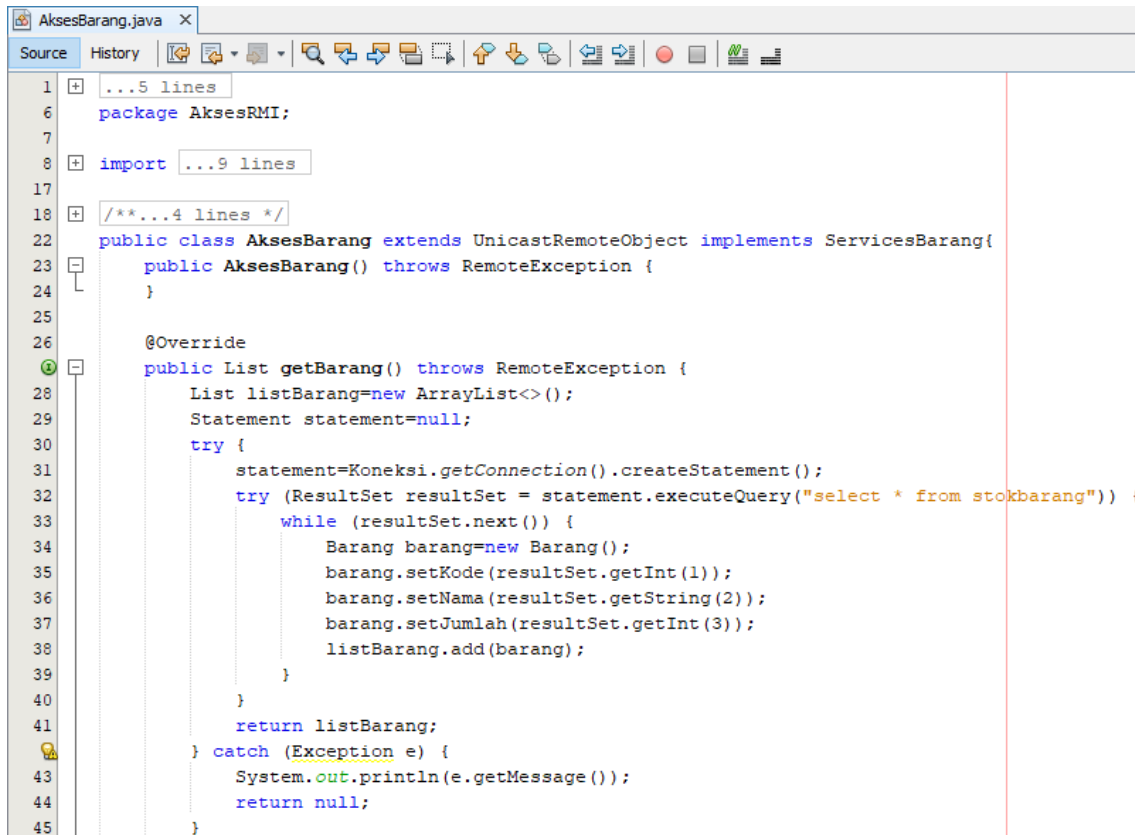
```java
...5 lines
package Koneksi;

import ...2 lines

/**...4 lines */
public class Koneksi {
    private static Connection connection=null;

    public static Connection getConnection() {
    try {
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        connection=DriverManager.getConnection("jdbc:mysql://localhost/stokbarang","root", "");
        System.out.println("Koneksi Sukses");
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
    return connection;
    }
    public static void main(String[] args) {
        Koneksi.getConnection();
    }
}
```

Jalankan kelas Koneksi jika koneksi sukses maka muncul tampilan seperti berikut

```
Output - Server (run)
run:
Koneksi Sukses
BUILD SUCCESSFUL (total time: 2 seconds)
```

10

Buat *package* baru pada proyek Server dengan nama *package* AksesRMI yang berisi kelas baru AksesBarang dan AksesUser. Masukkan *source code* berikut pada kelas AksesBarang

```
AksesBarang.java ×
Source  History

1  ⊞  ...5 lines
6      package AksesRMI;
7
8  ⊞  import ...9 lines
17
18 ⊞  /**...4 lines */
22     public class AksesBarang extends UnicastRemoteObject implements ServicesBarang{
23 ⊟       public AksesBarang() throws RemoteException {
24         }
25
26         @Override
   ⊟       public List getBarang() throws RemoteException {
28             List listBarang=new ArrayList<>();
29             Statement statement=null;
30             try {
31                 statement=Koneksi.getConnection().createStatement();
32                 try (ResultSet resultSet = statement.executeQuery("select * from stokbarang")) {
33                     while (resultSet.next()) {
34                         Barang barang=new Barang();
35                         barang.setKode(resultSet.getInt(1));
36                         barang.setNama(resultSet.getString(2));
37                         barang.setJumlah(resultSet.getInt(3));
38                         listBarang.add(barang);
39                     }
40                 }
41                 return listBarang;
           } catch (Exception e) {
43                 System.out.println(e.getMessage());
44                 return null;
45             }
```

```java
46          finally{
47              if (statement!=null) {
48                  try {
49                      statement.close();
50                  } catch (Exception e) {
51                  }
52              }
53          }
54      }
55
56      @Override
57      public void insert(Barang barang) throws RemoteException {
58      //throw new UnsupportedOperationException("Not supported yet."); //To change body of ger
59          Statement statement=null;
60          try {
61              statement= Koneksi.getConnection().createStatement();
62              statement.executeUpdate("insert into stokbarang values ('"+barang.getKode()+"'"
63                      + ",'"+barang.getNama()+"','"+barang.getJumlah()+"')");
64          } catch (Exception e) {
65              System.out.println(e.getMessage());
66
67          }finally{
68              if (statement!=null) {
69                  try {
70                      statement.close();
71                  } catch (Exception e) {
72                  }
73              }
74          }
75      }
76
77      @Override
78      public void update(Barang barang) throws RemoteException {
79      //throw new UnsupportedOperationException("Not supported yet."); //To change body of generat
80          Statement statement=null;
81          try {
82              statement=Koneksi.getConnection().createStatement();
83              statement.executeUpdate("update stokbarang set nama='"+barang.getNama() +
84                      "',jumlah='"+barang.getJumlah()+"'where kode='"+barang.getKode()+"'");
85
86          } catch (Exception e) {
87              System.out.println(e.getMessage());
88
89          }finally{
90              if (statement!=null) {
91                  try {
92                      statement.close();
93                  } catch (Exception e) {
94                  }
95
96              }
97          }
98      }
99
100     @Override
101     public void delete(Barang barang) throws RemoteException {
102     // throw new UnsupportedOperationException("Not supported yet."); //To change body of genera
103         Statement statement=null;
104         try {
105             statement=Koneksi.getConnection().createStatement();
106             statement.executeUpdate("delete from stokbarang where kode='"+barang.getKode()+"'");
107         } catch (Exception e) {
```
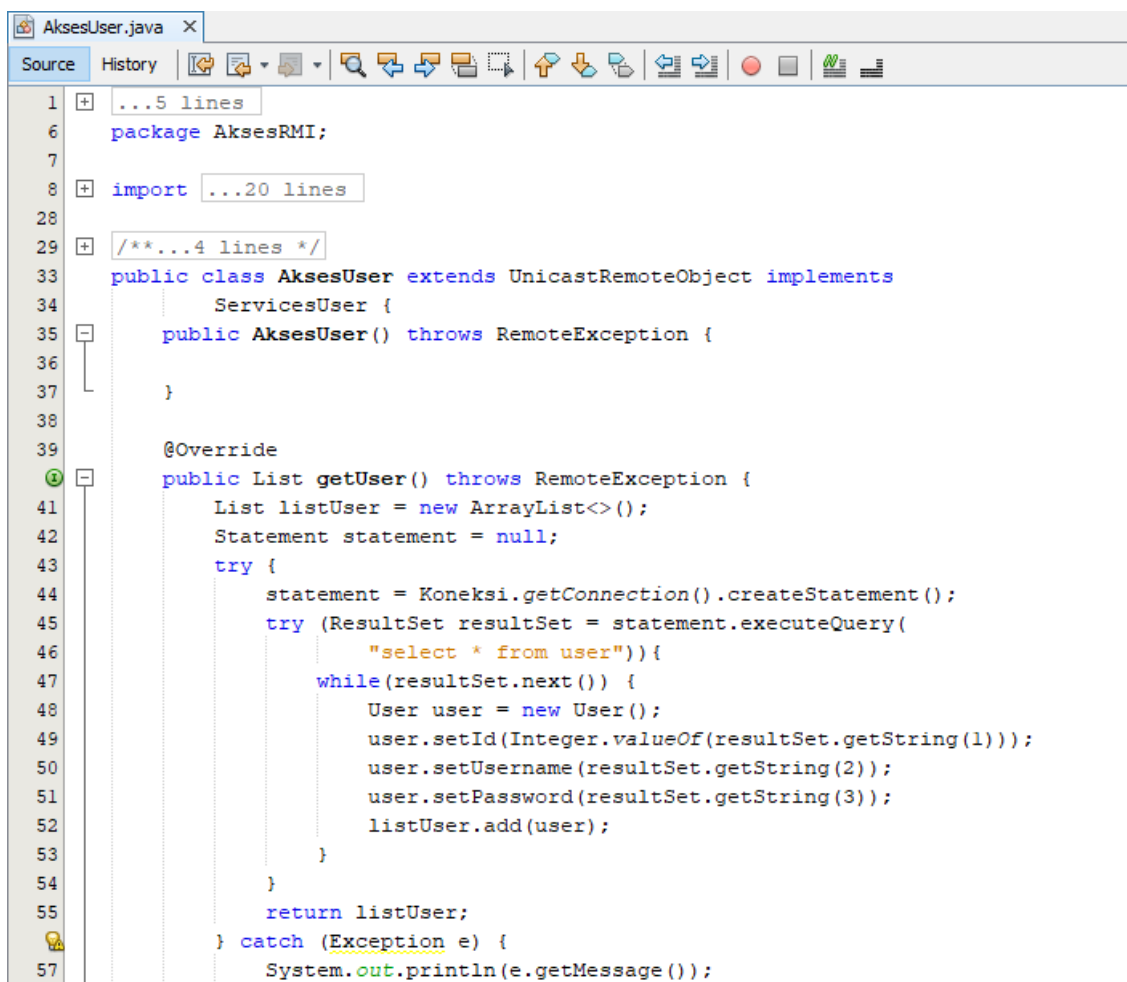
```
108              System.out.println(e.getMessage());
109          }finally{
110              if (statement!=null) {
111                  try {
112                      statement.close();
                    } catch (Exception e) {
114                  }
115              }
116          }
117      }
118  }
```

Masukkan *source code* berikut kelas AksesUser.java

```
AksesUser.java  ×
Source  History

  1  ⊞  ...5 lines
  6     package AksesRMI;
  7
  8  ⊞  import  ...20 lines
 28
 29  ⊞  /**...4 lines */
 33     public class AksesUser extends UnicastRemoteObject implements
 34             ServicesUser {
 35         public AksesUser() throws RemoteException {
 36
 37         }
 38
 39         @Override
 40         public List getUser() throws RemoteException {
 41             List listUser = new ArrayList<>();
 42             Statement statement = null;
 43             try {
 44                 statement = Koneksi.getConnection().createStatement();
 45                 try (ResultSet resultSet = statement.executeQuery(
 46                         "select * from user")){
 47                     while(resultSet.next()) {
 48                         User user = new User();
 49                         user.setId(Integer.valueOf(resultSet.getString(1)));
 50                         user.setUsername(resultSet.getString(2));
 51                         user.setPassword(resultSet.getString(3));
 52                         listUser.add(user);
 53                     }
 54                 }
 55                 return listUser;
                } catch (Exception e) {
 57                 System.out.println(e.getMessage());
```

13

```java
 58                 return null;
 59             }
 60             finally {
 61                 if(statement != null){
 62                     try {
 63                         statement.close();
                        } catch (Exception e) {
 65                     }
 66                 }
 67             }
 68         }
 69
 70     public void insert(User user) throws RemoteException {
 71         Statement statement = null;
 72         try {
 73             statement = Koneksi.getConnection().createStatement();
 74             String newuser = enkripsi(user.getUsername());
 75             String newpass = enkripsi(user.getPassword());
 76             statement.executeUpdate("insert into user (username, password) values ('"
                        + newuser + "','"+newpass+"')");
        } catch (Exception e) {
 79             System.out.println("gagal");
 80         } finally {
 81             if(statement != null) {
 82                 try {
 83                     statement.close();
                    } catch (Exception e) {
 85                 }
 86             }
 87         }
 88     }
 89
 90     public String enkripsi(String plain) throws NoSuchAlgorithmException, NoSuchProviderException,
 91             NoSuchPaddingException, InvalidKeyException, ShortBufferException, IllegalBlockSizeException,
 92             BadPaddingException{
 93         // TODO code application logic here
 94         Security.addProvider(new org.bouncycastle.jce.provider.BouncyCastleProvider());
 95         byte[] input = plain.getBytes();
 96         byte[] keyBytes = new byte[] { 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09,
 97         0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f };
 98
 99         SecretKeySpec key = new SecretKeySpec(keyBytes, "AES");
100
101         Cipher cipher = Cipher.getInstance("AES/ECB/PKCS7Padding", "BC");
102
103         System.out.println(new String(input));
104
105         // encryption pass
106         cipher.init(Cipher.ENCRYPT_MODE, key);
107
108         byte[] cipherText = new byte[cipher.getOutputSize(input.length)];
109         int ctLength = cipher.update(input, 0, input.length, cipherText, 0);
110         ctLength += cipher.doFinal(cipherText, ctLength);
111         String newpass = (new String(cipherText));
112         System.out.println(newpass);
113         return newpass;
114
115
116     }
117
```

14
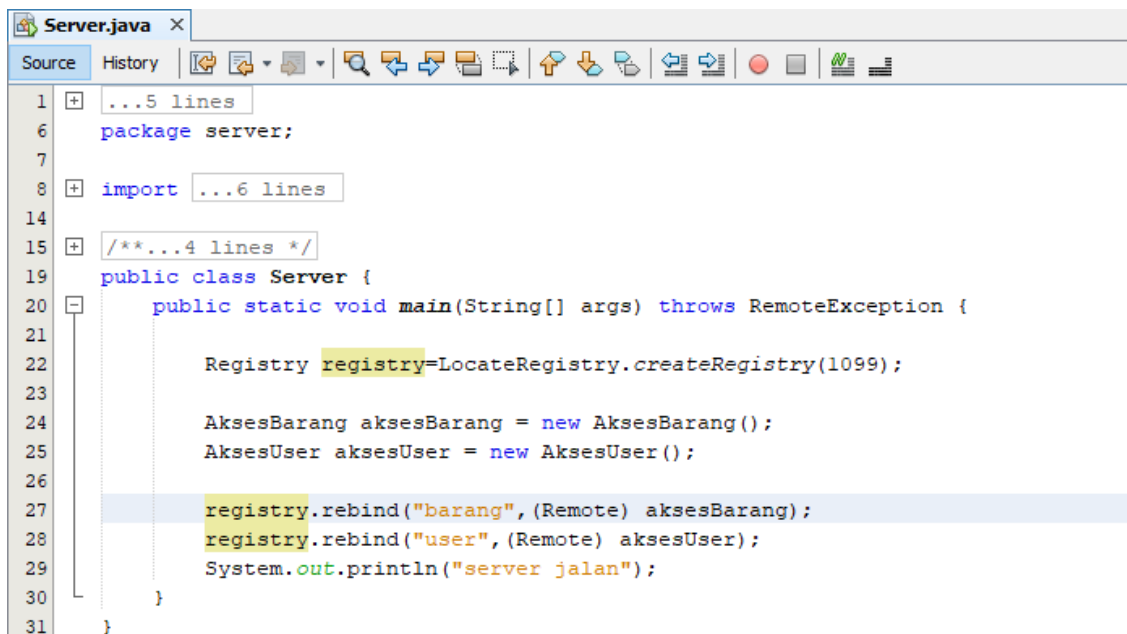
```
    ●     public String dekripsi(String plain) throws NoSuchAlgorithmException, NoSuchProviderException,
119             NoSuchPaddingException, InvalidKeyException, ShortBufferException, IllegalBlockSizeException,
120 ⊟           BadPaddingException{
121         // TODO code application logic here
122         Security.addProvider(new org.bouncycastle.jce.provider.BouncyCastleProvider());
123         byte[] input = plain.getBytes();
124         byte[] keyBytes = new byte[] { 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09,
125         0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f };
126
127         SecretKeySpec key = new SecretKeySpec(keyBytes, "AES");
128
129         Cipher cipher = Cipher.getInstance("AES/ECB/PKCS7Padding", "BC");
130
131         System.out.println(new String(input));
132
133         // decryption pass
134         cipher.init(Cipher.DECRYPT_MODE, key);
135         byte[] plainText = new byte[cipher.getOutputSize(input.length)];
136         int ptLength = cipher.update(input, 0, input.length, plainText, 0);
137         ptLength += cipher.doFinal(plainText, ptLength);
138         String pass = (new String(plainText));
139         return pass;
140     }
141 }
```

Buka kelas Server pada *package* Server. Masukkan *source code* berikut pada kelas Server.
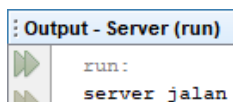
```
🖼 Server.java ✕
Source  History  🖉 🖩 ▾ 🖩 ▾ | 🔍 🖵 🖵 🖵 🖵 | 🖧 🖧 🖧 | 🖆 🖆 | ● ■ | 🖩 🖩
 1 ⊞  ...5 lines
 6     package server;
 7
 8 ⊞  import  ...6 lines
14
15 ⊞  /**...4 lines */
19     public class Server {
20 ⊟     public static void main(String[] args) throws RemoteException {
21
22         Registry registry=LocateRegistry.createRegistry(1099);
23
24         AksesBarang aksesBarang = new AksesBarang();
25         AksesUser aksesUser = new AksesUser();
26
27         registry.rebind("barang",(Remote) aksesBarang);
28         registry.rebind("user",(Remote) aksesUser);
29         System.out.println("server jalan");
30     }
31 }
```

Jalankan Kelas Server. *Server* berhasil dijalankan

```
Output - Server (run)
▷    run:
▷▷   server jalan
```

Buka *project* Client. Buat form dengan nama Login pada *package* Client. Buatlah desain kelas Login

Masukkan *source code* berikut pada kelas Login

```
148          tabelUser.setModel(new DefaultTableModel(data, header));
149          JTableHeader head = tabelUser.getTableHeader();
150          head.setBackground(Color.orange);
151      }
152
153      private void insert(ServicesUser servicesUser) throws RemoteException{
154          User u = new User();
155          u.setUsername(tUsername.getText());
156          u.setPassword(tPassword.getText());
157          servicesUser.insert(u);
158      }
159
160      public String enkripsi(String plain) throws InvalidKeyException, ShortBufferException,
161              IllegalBlockSizeException, BadPaddingException, NoSuchAlgorithmException,
162          NoSuchProviderException, NoSuchPaddingException{
163          // TODO code application logic here
164          Security.addProvider(new org.bouncycastle.jce.provider.BouncyCastleProvider());
165          byte[] input = plain.getBytes();
166          byte[] keyBytes = new byte[] { 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06,
                  0x07, 0x08, 0x09,
168          0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f };
169
170          SecretKeySpec key = new SecretKeySpec(keyBytes, "AES");
171
172          Cipher cipher = Cipher.getInstance("AES/ECB/PKCS7Padding", "BC");
173
174          System.out.println(new String(input));
175
176          // encryption pass
177          cipher.init(Cipher.ENCRYPT_MODE, key);
178
179          byte[] cipherText = new byte[cipher.getOutputSize(input.length)];
180          int ctLength = cipher.update(input, 0, input.length, cipherText, 0);
181          ctLength += cipher.doFinal(cipherText, ctLength);
182          String newpass = (new String(cipherText));
183          System.out.println(newpass);
184          return newpass;
185      }
186
187      private void clear(){
188          tUsername.setText("");
189          tPassword.setText("");
190      }
191
192      private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
193          // TODO add your handling code here:
194          try {
                ResultSet r =null;
196              String username = tUsername.getText();
197              String password = tPassword.getText();
198              username = enkripsi(username);
199              password = enkripsi(password);
200
201              try {
202                  Connection c = Koneksi.getConnection();
203                  String sql = "select * from user where username = ? and password = ?";
204                  PreparedStatement p = c.prepareStatement(sql);
205                  p.setString(1, username);
206                  p.setString(2, password);
207                  r = p.executeQuery();
208
```

17

```java
            if (r.next())    {
                JOptionPane.showMessageDialog(this, "Login Sukses");
                StokBarang sb = new StokBarang();
                sb.setVisible(true);
                this.dispose();
            }   else    {
                JOptionPane.showMessageDialog(this, "Kesalahan username/password");
            }
        } catch (SQLException ex) {
            Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
        }

    } catch (NoSuchAlgorithmException ex) {
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
    } catch (NoSuchProviderException ex) {
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
    } catch (NoSuchPaddingException ex) {
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
    } catch (InvalidKeyException ex) {
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
    } catch (ShortBufferException ex) {
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
    } catch (IllegalBlockSizeException ex) {
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
    } catch (BadPaddingException ex) {
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    System.exit(0);
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (tUsername.getText().equals("") || tPassword.getText().
            equals("")) {
        JOptionPane.showMessageDialog(null, "Silahkan Input Data Dulu");
    } else{
        try {
            insert(servicesUser);
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
        }
    }
    clear();
}

/**...3 lines */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                new Login().setVisible(true);
```

18

```
  292          } catch (RemoteException ex) {
  293              Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
  294          } catch (NotBoundException ex) {
  295              Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
  296          }
  297        }
  298      });
  299    }
  300
  301    // Variables declaration - do not modify
  302    private javax.swing.JButton jButton1;
  303    private javax.swing.JButton jButton2;
  304    private javax.swing.JButton jButton3;
  305    private javax.swing.JLabel jLabel1;
  306    private javax.swing.JLabel jLabel2;
  307    private javax.swing.JPasswordField tPassword;
  308    private javax.swing.JTextField tUsername;
  309    // End of variables declaration
        }
```

Buat kelas koneksi dan masukkan *source code* berikut

```java
package client;

import ...2 lines

/**...4 lines */
public class Koneksi {
    private static Connection connection=null;

    public static Connection getConnection() {
    try {
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        connection=DriverManager.getConnection("jdbc:mysql://localhost/stokbarang","root", "");
        System.out.println("Koneksi Sukses");
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
    return connection;
    }
    public static void main(String[] args) {
        Koneksi.getConnection();
    }
}
```

Buatlah kelas StokBarang dengan JFrame Form. Buatlah desain kelas StokBarang
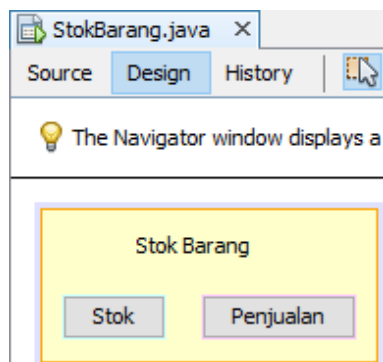
Masukkan *source code* berikut pada kelas StokBarang

```
StokBarang.java ×
Source  Design  History   |   ...
   1  ⊞  ...5 lines
   6     package client;
   7
   8  ⊞  import ...4 lines
  12
  13  ⊞  /**...4 lines */
  17     public class StokBarang extends javax.swing.JFrame {
  18
  19  ⊞      /** Creates new form stokbarang ...3 lines */
  22  ⊟      public StokBarang() {
  23             initComponents();
  24         }
  25
  26  ⊞      /** This method is called from within the constructor to initialize the form ...5
  31         @SuppressWarnings("unchecked")
  32  ⊞      Generated Code
 104
 105  ⊟      private void bStokActionPerformed(java.awt.event.ActionEvent evt) {
 106             // TODO add your handling code here:
 107             Stok stok = null;
 108             try {
 109                 stok = new Stok();
      ⚑      }   catch (RemoteException ex) {
 111                 Logger.getLogger(StokBarang.class.getName()).log(Level.SEVERE, null, ex);
 112             }   catch (NotBoundException ex) {
 113                 Logger.getLogger(StokBarang.class.getName()).log(Level.SEVERE, null, ex);
 114             }
      ⚑      stok.setVisible(true);
 116             this.dispose();
 117         }
 118
 119  ⊟      private void bPenjualanActionPerformed(java.awt.event.ActionEvent evt) {
 120             // TODO add your handling code here:
 121             Penjualan penjualan = null;
 122             try {
 123                 penjualan = new Penjualan();
      ⚑      }   catch (RemoteException ex)  {
 125                 Logger.getLogger(StokBarang.class.getName()).log(Level.SEVERE, null, ex);
 126             }   catch (NotBoundException ex) {
 127                 Logger.getLogger(StokBarang.class.getName()).log(Level.SEVERE, null, ex);
 128             }
      ⚑      penjualan.setVisible(true);
 130             this.dispose();
 131         }
 132
 133         // Variables declaration - do not modify
 134         private javax.swing.JButton bPenjualan;
 135         private javax.swing.JButton bStok;
 136         private javax.swing.JLabel jLabel1;
 137         private javax.swing.JPanel jPanel1;
 138         // End of variables declaration
 139     }
```

Buatlah kelas Stok dengan JFrame Form. Buatlah desain kelas Stok

Masukkan *source code* berikut pada kelas Stok

```java
231          data[x][2] = b.getJumlah();
232          ++x;
233      }
234      tblBarang.setModel(new DefaultTableModel(data, header));
235      JTableHeader head = tblBarang.getTableHeader();
236      head.setBackground(Color.orange);
237      // end
238      }
239
240      private void insert(ServicesBarang servicesBarang) throws RemoteException{
241
242          Barang b=new Barang();
243          b.setKode(Integer.valueOf(tKode.getText()));
244          b.setNama(tNama.getText());
245          b.setJumlah(Integer.valueOf(tJumlah.getText()));
246          servicesBarang.insert(b);
247      }
248
249      private void update(ServicesBarang servicesBarang) throws RemoteException{
250          Barang b = new Barang();
251          b.setKode(Integer.valueOf(tKode.getText()));
252          b.setNama(tNama.getText());
253          b.setJumlah(Integer.valueOf(tJumlah.getText()));
254          servicesBarang.update(b);
255      }
256      private void delete(ServicesBarang servicesBarang) throws RemoteException{
257          Barang b = new Barang();
258          b.setKode(Integer.valueOf(tKode.getText()));
259          servicesBarang.delete(b);
260      }
261
```

```java
262    private void clear(){
263        tKode.setText("");
264        tNama.setText("");
265        tJumlah.setText("");
266    }
267
268    private void tNamaActionPerformed(java.awt.event.ActionEvent evt) {
269        // TODO add your handling code here:
270    }
271
272    private void bSimpanActionPerformed(java.awt.event.ActionEvent evt) {
273        // TODO add your handling code here:
274        if (tKode.getText().equals("") || tNama.getText().equals("")
275                || tJumlah.getText().equals("")) {
276            JOptionPane.showMessageDialog(null, "Silahkan Input Data Dulu");
277        }
278        else{
279            try {
280                insert(servicesBarang);
281                select(tblBarang);
282            } catch (Exception e) {
283                JOptionPane.showMessageDialog(null, e.getMessage());
284            }
285        }
286        clear();
287    }
288
289    private void tblBarangMouseClicked(java.awt.event.MouseEvent evt) {
290        // TODO add your handling code here:
291        tKode.setText(tblBarang.getValueAt(tblBarang.getSelectedRow(),0).toString());
292        tNama.setText(tblBarang.getValueAt(tblBarang.getSelectedRow(),1).toString());
293        tJumlah.setText(tblBarang.getValueAt(tblBarang.getSelectedRow(),2).toString());
294    }
295
296    private void bRubahActionPerformed(java.awt.event.ActionEvent evt) {
297        // TODO add your handling code here:
298        if (tKode.getText().equals("") || tNama.getText().equals("")
299                || tJumlah.getText().equals("")) {
300            JOptionPane.showMessageDialog(null, "Silahkan pilih data yang akan di Update");
301        }else{
302            try {
303                update(servicesBarang);
304                select(tblBarang);
305            } catch (Exception e) {
306                JOptionPane.showMessageDialog(null, e.getMessage());
307            }
308            clear();
309        }
310    }
311
312    private void bHapusActionPerformed(java.awt.event.ActionEvent evt) {
313        // TODO add your handling code here:
314        if (tKode.getText().equals("") || tNama.getText().equals("")
315                || tJumlah.getText().equals("")) {
316            JOptionPane.showMessageDialog(null, "Silahkan pilih data yang akan di Delete");
317        } else {
318            try {
319                delete(servicesBarang);
320                select(tblBarang);
321            } catch (Exception e) {
322                JOptionPane.showMessageDialog(null, e.getMessage());
323            }
324            clear();
```

```
325            }
326        }
327
328        private void bKeluarActionPerformed(java.awt.event.ActionEvent evt) {
329            // TODO add your handling code here:
330            StokBarang sb = new StokBarang();
331            sb.setVisible(true);
332            this.dispose();
333        }
334
335        // Variables declaration - do not modify
336        private javax.swing.JButton bHapus;
337        private javax.swing.JButton bKeluar;
338        private javax.swing.JButton bRubah;
339        private javax.swing.JButton bSimpan;
340        private javax.swing.JLabel jLabel1;
341        private javax.swing.JLabel jLabel2;
342        private javax.swing.JLabel jLabel3;
343        private javax.swing.JLabel jLabel4;
344        private javax.swing.JPanel jPanel1;
345        private javax.swing.JScrollPane jScrollPane1;
346        private javax.swing.JTextField tJumlah;
347        private javax.swing.JTextField tKode;
348        private javax.swing.JTextField tNama;
349        private javax.swing.JTable tblBarang;
350        // End of variables declaration
351    }
```

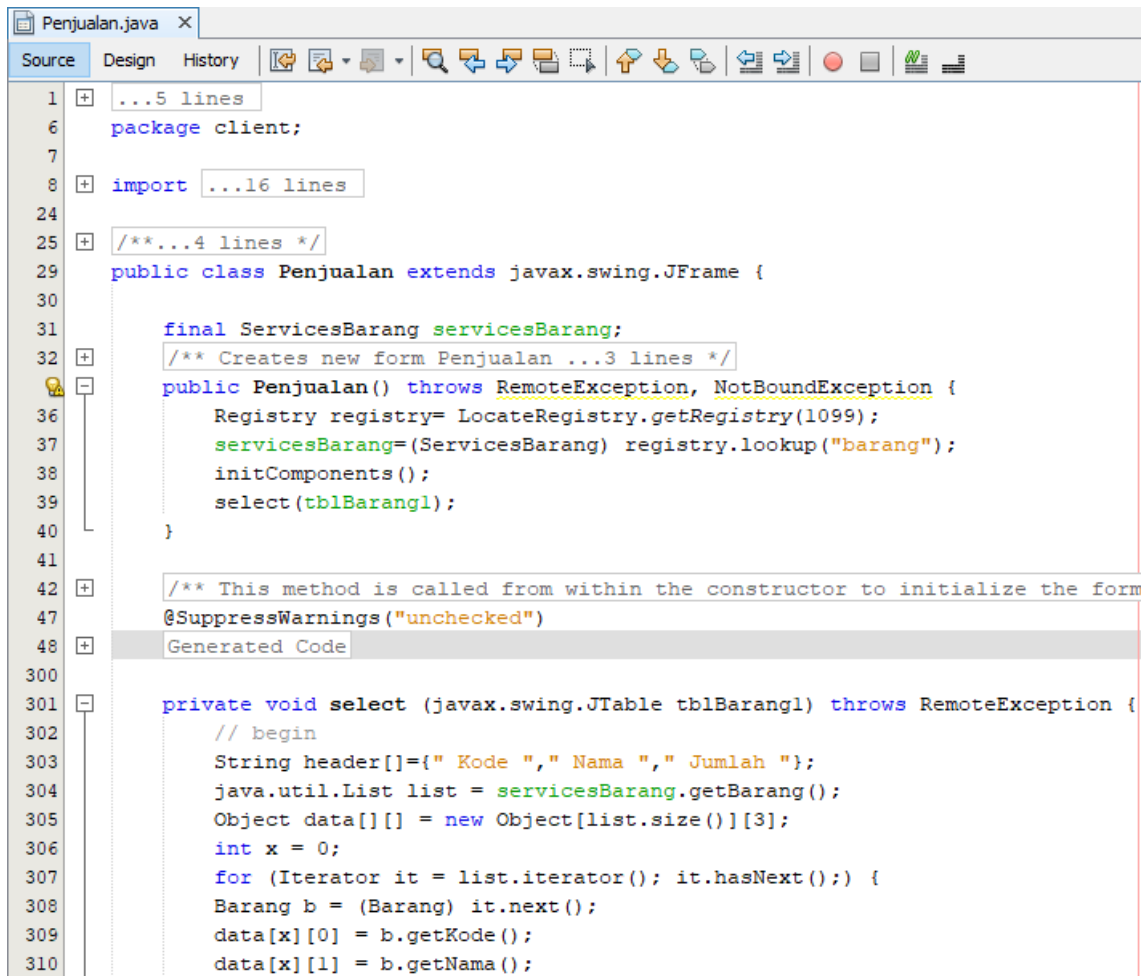Buatlah kelas Penjualan dengan JFrame Form. Buatlah desain kelas Penjualan

Masukkan *source code* berikut pada kelas Penjualan

```
Penjualan.java  ×

Source  Design  History

  1  ⊞  ...5 lines
  6      package client;
  7
  8  ⊞  import  ...16 lines
 24
 25  ⊞  /**...4 lines */
 29      public class Penjualan extends javax.swing.JFrame {
 30
 31          final ServicesBarang servicesBarang;
 32  ⊞      /** Creates new form Penjualan ...3 lines */
    ⊟      public Penjualan() throws RemoteException, NotBoundException {
 36              Registry registry= LocateRegistry.getRegistry(1099);
 37              servicesBarang=(ServicesBarang) registry.lookup("barang");
 38              initComponents();
 39              select(tblBarang1);
 40          }
 41
 42  ⊞      /** This method is called from within the constructor to initialize the form
 47          @SuppressWarnings("unchecked")
 48  ⊞      Generated Code
300
301  ⊟      private void select (javax.swing.JTable tblBarang1) throws RemoteException {
302              // begin
303              String header[]={" Kode "," Nama "," Jumlah "};
304              java.util.List list = servicesBarang.getBarang();
305              Object data[][] = new Object[list.size()][3];
306              int x = 0;
307              for (Iterator it = list.iterator(); it.hasNext();) {
308              Barang b = (Barang) it.next();
309              data[x][0] = b.getKode();
310              data[x][1] = b.getNama();
```

25

```
311          data[x][2] = b.getJumlah();
312          ++x;
313      }
314      tblBarang1.setModel(new DefaultTableModel(data, header));
315      JTableHeader head = tblBarang1.getTableHeader();
316      head.setBackground(Color.orange);
317      // end
318      }
319
320      private void update(ServicesBarang servicesBarang) throws RemoteException{
321          TableModel model = tblBarang1.getModel();
322          int index = tblBarang1.getSelectedRow();
323          int jumlah = Integer.valueOf(tJumlah1.getText());
324          int jumlah1 = Integer.valueOf(model.getValueAt(index, 2).toString());
325          jumlah = jumlah1 - jumlah;
326          Barang b=new Barang();
327          b.setKode(Integer.valueOf(tKode1.getText()));
328          b.setNama(tNama1.getText());
329          b.setJumlah(jumlah);
330          servicesBarang.update(b);
331
332      }
333
334      private void clear(){
335          tKode1.setText("");
336          tNama1.setText("");
337          tJumlah1.setText("");
338      }
339
340      private void tNamaActionPerformed(java.awt.event.ActionEvent evt) {
341          // TODO add your handling code here:
342      }
```

26

```java
        private void bInputActionPerformed(java.awt.event.ActionEvent evt) {
            // TODO add your handling code here:
        }

        private void tblBarangMouseClicked(java.awt.event.MouseEvent evt) {
            // TODO add your handling code here:
        }

        private void tNama1ActionPerformed(java.awt.event.ActionEvent evt) {
            // TODO add your handling code here:
        }

        private void bInput1ActionPerformed(java.awt.event.ActionEvent evt) {
            // TODO add your handling code here:
            if (tKode1.getText().equals("") || tNama1.getText().equals("")
            || tJumlah1.getText().equals(""))    {
                JOptionPane.showMessageDialog(null, "Silahkan Input Data Dulu");
            }
            else{
                try {
                    update(servicesBarang);
                    select(tblBarang1);
                } catch (Exception e) {
                    JOptionPane.showMessageDialog(null, e.getMessage());
                }
            }
            clear();
        }

        private void tblBarang1MouseClicked(java.awt.event.MouseEvent evt) {
            // TODO add your handling code here:
            tKode1.setText(tblBarang1.getValueAt(tblBarang1.getSelectedRow(),0).toString());
            tNama1.setText(tblBarang1.getValueAt(tblBarang1.getSelectedRow(),1).toString());
        }

        private void bKeluarActionPerformed(java.awt.event.ActionEvent evt) {
            // TODO add your handling code here:
            StokBarang sb = new StokBarang();
            sb.setVisible(true);
            this.dispose();
        }

        // Variables declaration - do not modify
        private javax.swing.JButton bInput;
        private javax.swing.JButton bInput1;
        private javax.swing.JButton bKeluar;
        private javax.swing.JFrame jFrame1;
        private javax.swing.JLabel jLabel1;
        private javax.swing.JLabel jLabel2;
        private javax.swing.JLabel jLabel3;
        private javax.swing.JLabel jLabel4;
        private javax.swing.JLabel jLabel5;
        private javax.swing.JLabel jLabel6;
        private javax.swing.JLabel jLabel7;
        private javax.swing.JLabel jLabel8;
        private javax.swing.JPanel jPanel1;
        private javax.swing.JScrollPane jScrollPane1;
        private javax.swing.JScrollPane jScrollPane2;
        private javax.swing.JTextField tJumlah;
        private javax.swing.JTextField tJumlah1;
        private javax.swing.JTextField tKode;
```

27

```
405    private javax.swing.JTextField tKode1;
406    private javax.swing.JTextField tNama;
407    private javax.swing.JTextField tNama1;
408    private javax.swing.JTable tblBarang;
409    private javax.swing.JTable tblBarang1;
410    // End of variables declaration
411 }
```

Jalankan program. Program akan menampilkan sebagai berikut

Masukkan data dan tekan *button* Daftar untuk daftar

Masukkan data dan tekan *button* Login untuk masuk

Program akan menampilkan sebagai berikut jika *username* dan *password* sesuai. Tekan *button* OK untuk melanjutkan

Program akan menampilkan sebagai berikut. Tekan *button* Stok atau Penjualan

Program akan menampilkan sebagai berikut setelah menekan *button* Stok

Masukkan data dan tekan *button* Simpan untuk menyimpan data

Program akan menampilkan sebagai berikut

Ubah teks yang terdapat pada program dan tekan *button* Rubah untuk mengubah data

Program akan menampilkan sebagai berikut
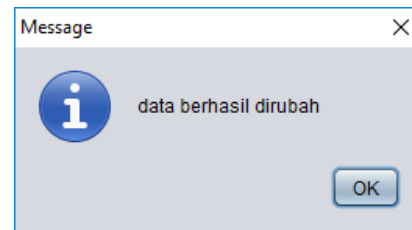
Tekan *button* Hapus untuk menghapus data

Program akan menampilkan sebagai berikut

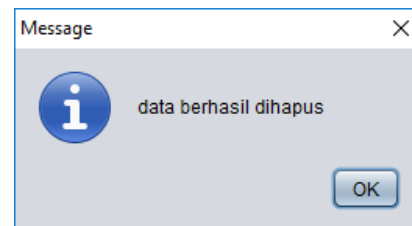Program akan menampilkan sebagai berikut setelah menekan *button* Penjualan. Program berikut merupakan program yang telah dimasukkan data pada tampilan Stok Barang.

Masukkan jumlah terjual dan tekan *button* Simpan untuk mengurangi stok barang

Program akan menampilkan sebagai berikut