

Laporan Hasil Praktikum

Pemrograman Desktop



Tugas 11

ALYA AIMAN SALSABILA ARIF

1817101379

Tingkat III Rekayasa Perangkat Lunak Kripto

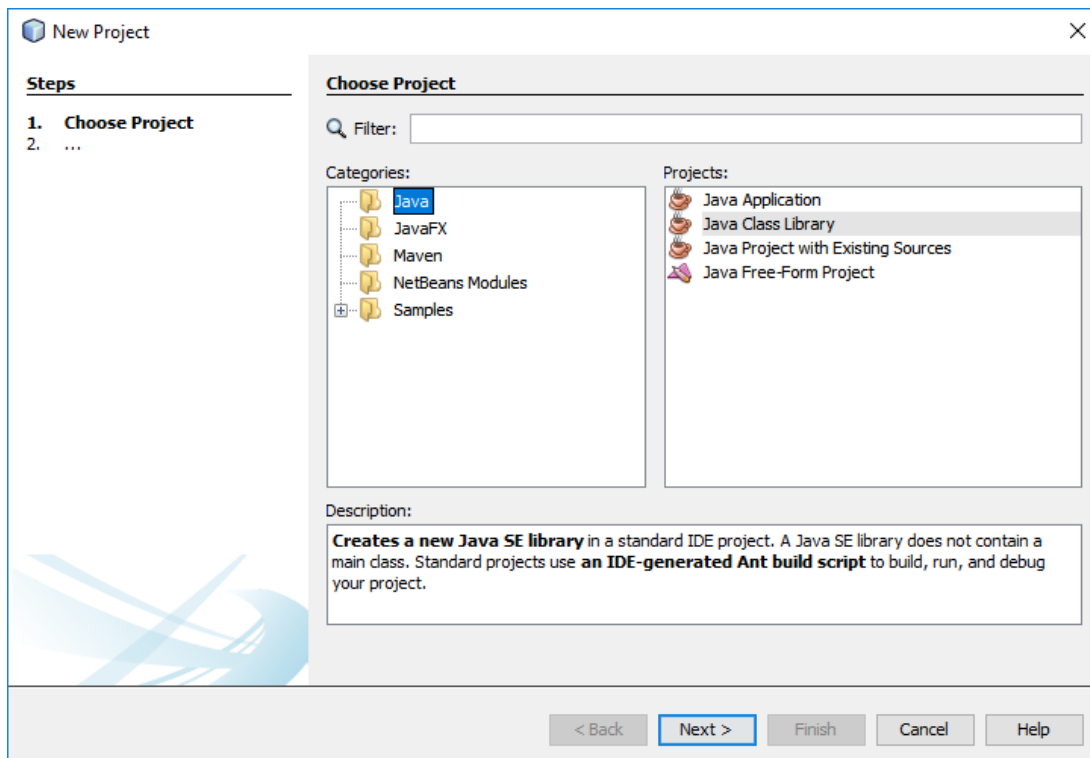
Politeknik Siber dan Sandi Negara

2020/2021

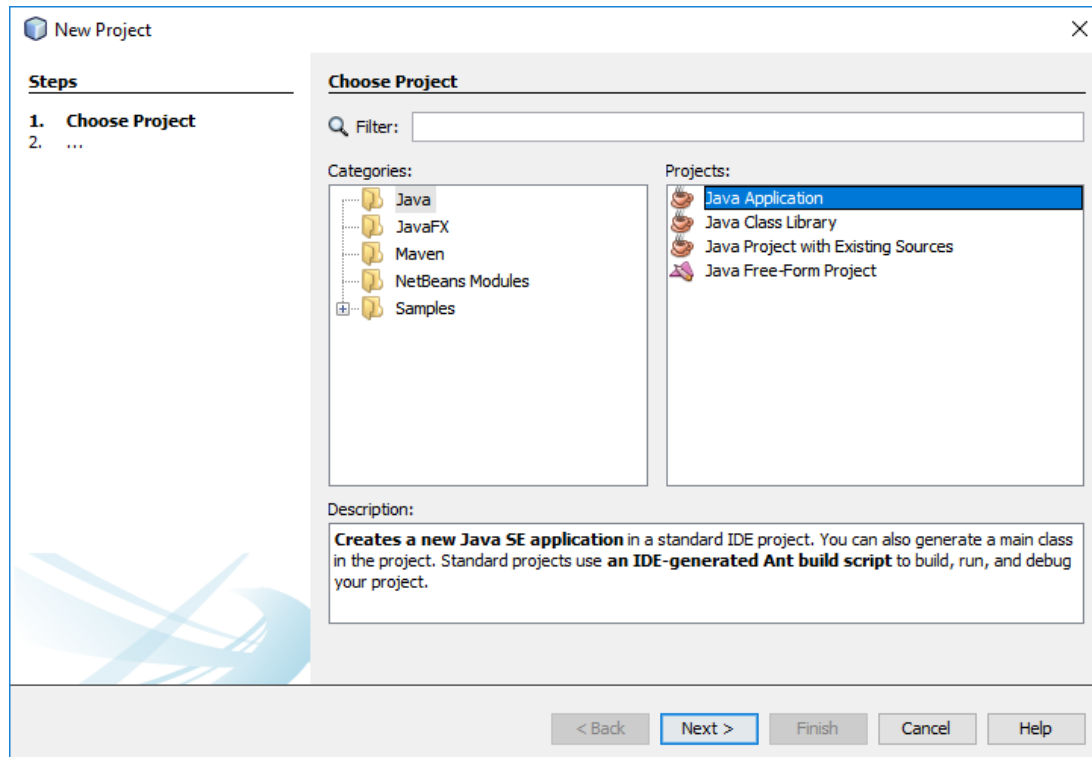
Buat basis data baru akademik dan tabel mahasiswa

```
MariaDB [(none)]> create database akademik;  
Query OK, 1 row affected (0.003 sec)  
  
MariaDB [(none)]> use akademik;  
Database changed  
MariaDB [akademik]> create table mahasiswa (nim varchar(30) not null primary key,  
-> nama varchar(100) not null,  
-> prodi varchar(50) not null);  
Query OK, 0 rows affected (0.242 sec)
```

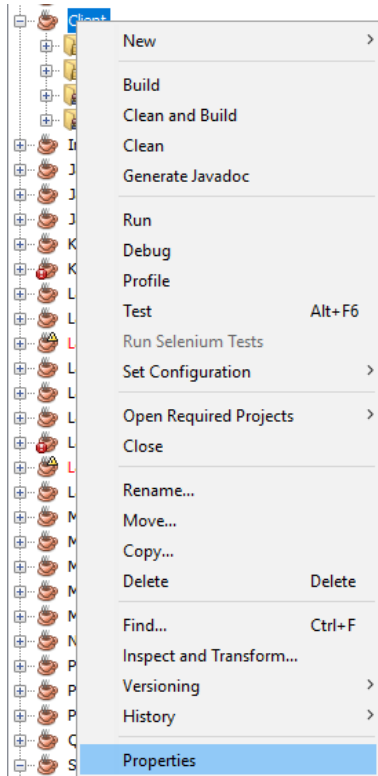
Jalankan Netbeans kemudian buat *Project* terlebih dahulu *New Project* -> *Java Class Library*. Pilih *Next* kemudian beri nama *Java library* pada *project name* Komunikasi.api sebagai penghubung antara *client* dan *server*



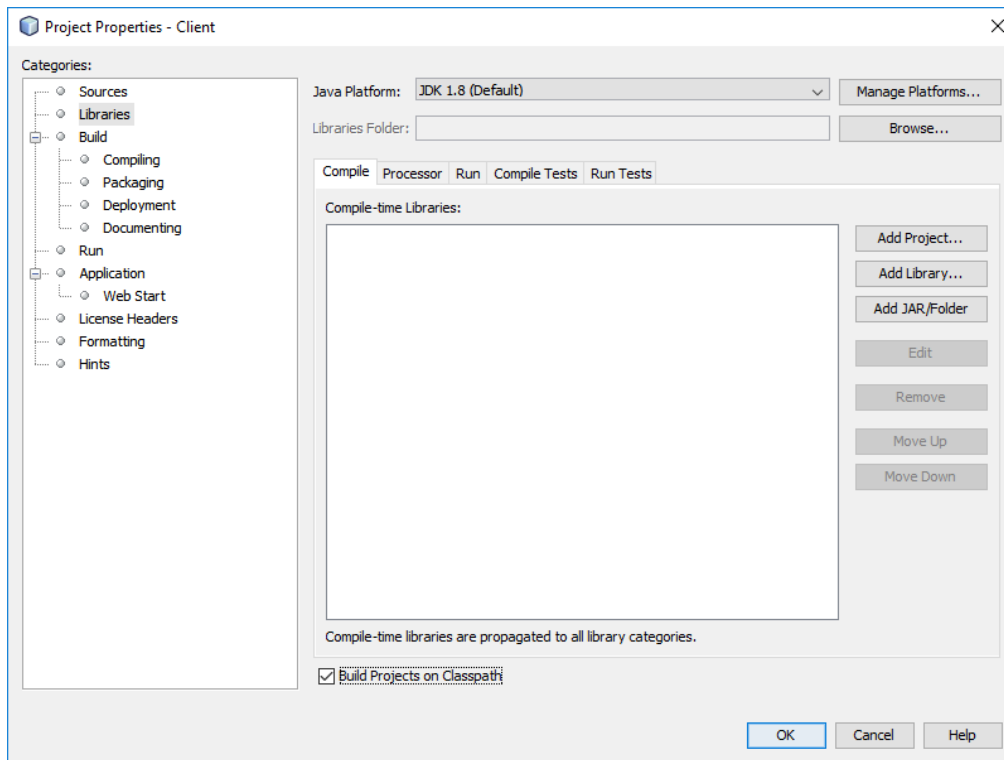
Buat kembali *project* baru *New Project* -> *Java Application* kemudian *Next*. Setelah ditekan *Next*, beri nama *project* yang dibuat dengan *Server*



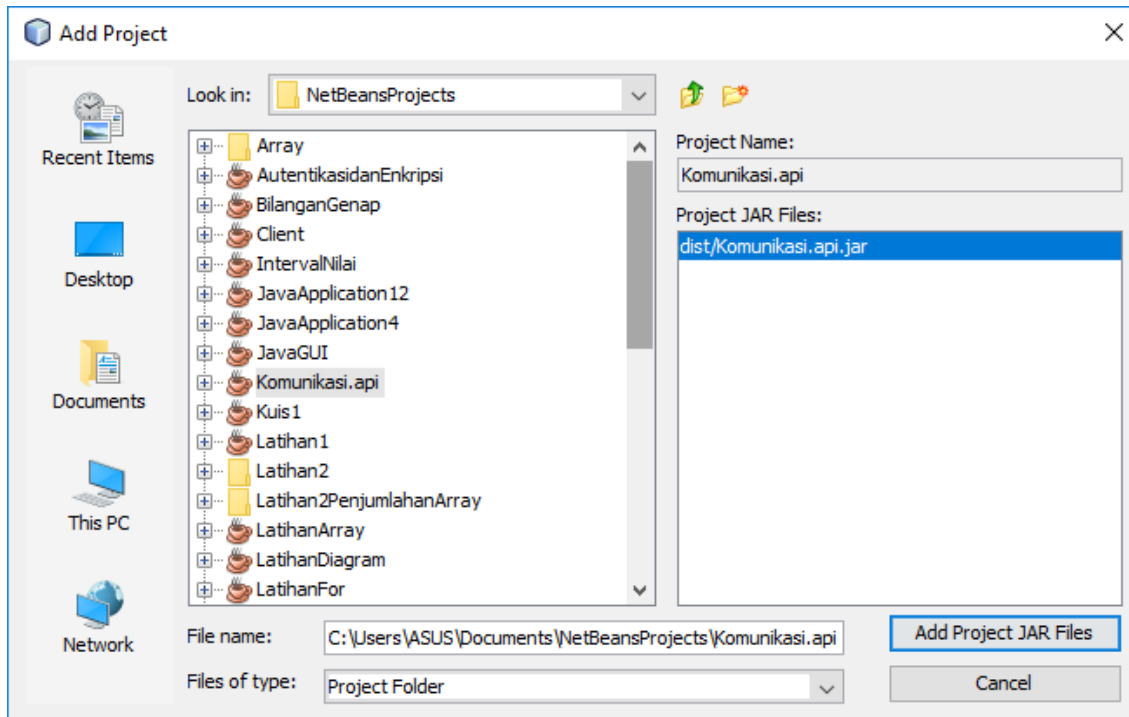
Buat kembali *project* baru *New Project* -> *Java Application* dengan nama Client. Sehingga, terdapat tiga *project* yang dimiliki dengan nama Client, Server dan Komunikasi.api. Untuk menghubungkan antara *client* dan *server* klik kanan pada *project* Client kemudian pilih *Properties*



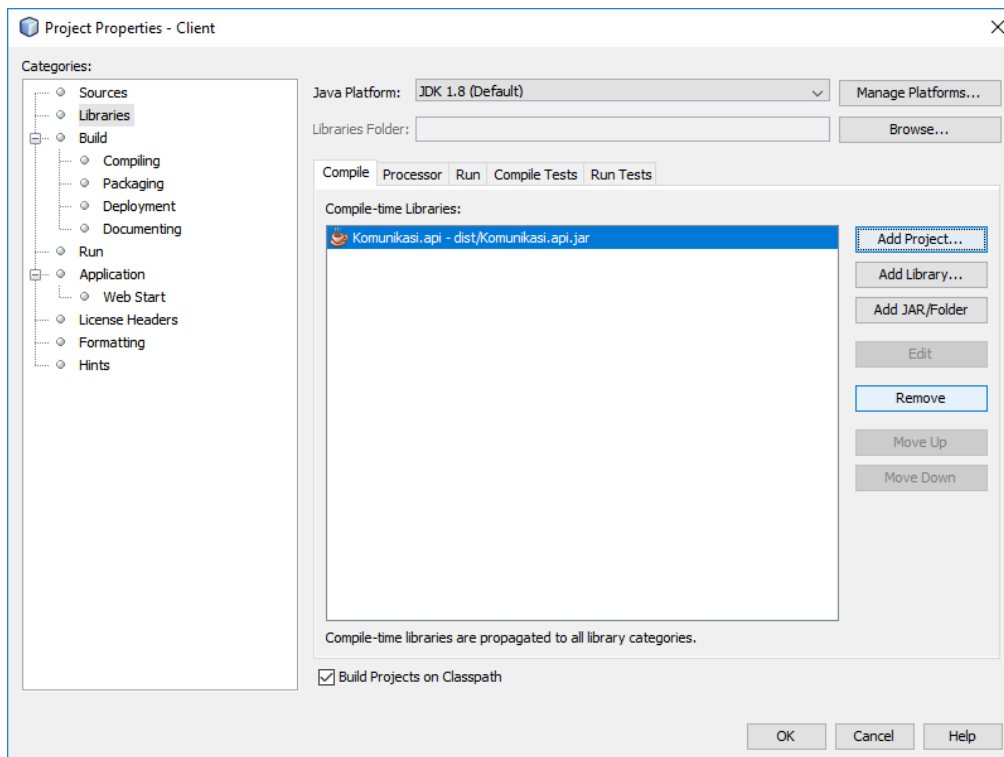
Berikut tampilan yang dihasilkan. Pilih *Libraries* -> *Add Project*



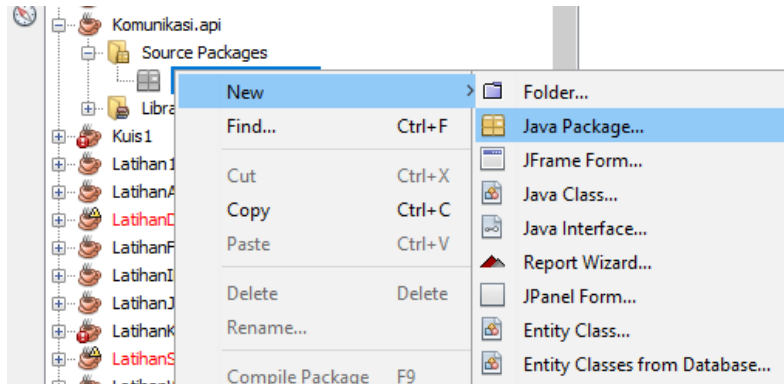
Pilih *project* Komunikasi.api yang sudah dibuat sebelumnya kemudian pilih *Add Project JAR Files*



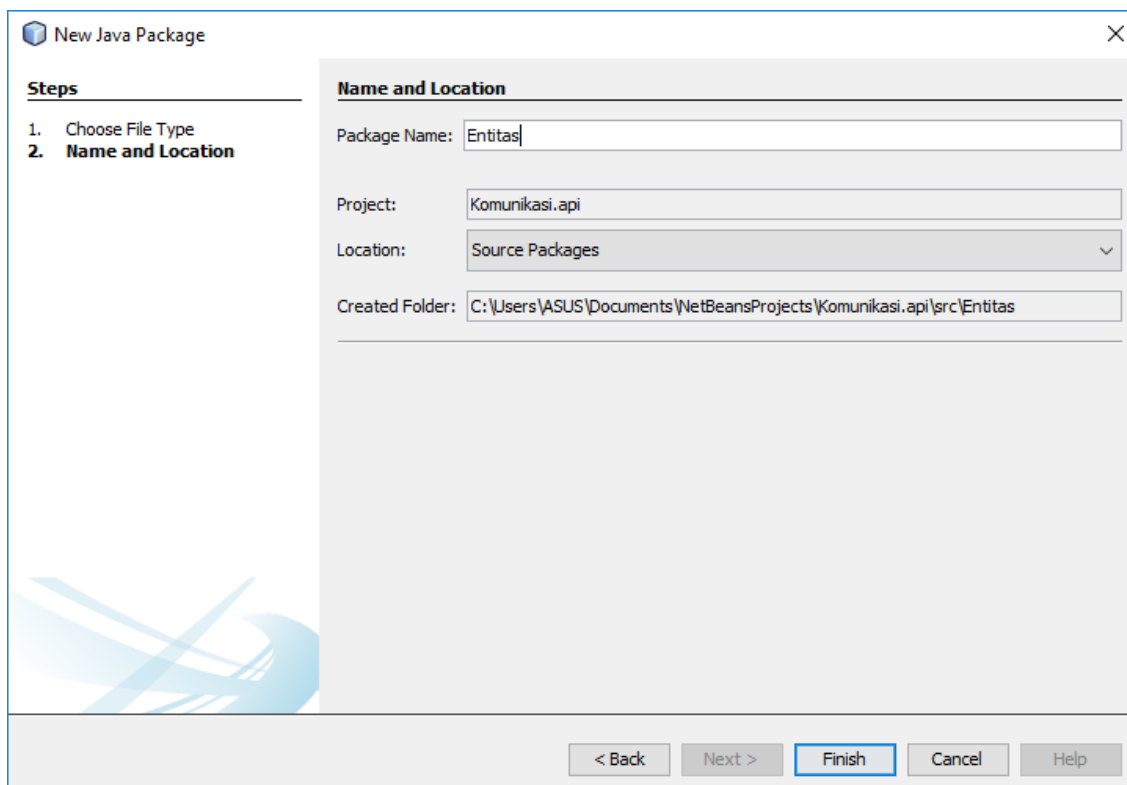
Klik *OK*



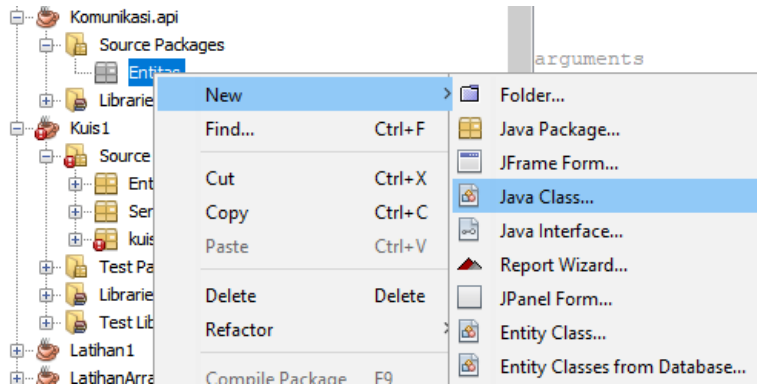
Lakukan juga pada project Server pilih *Properties->Libraries->Add Project->Komunikasi-api->OK* sama seperti pada Client. Kemudian pada Komunikasi.api pada *Source Package->Default Package* klik kanan *New->Java Package*



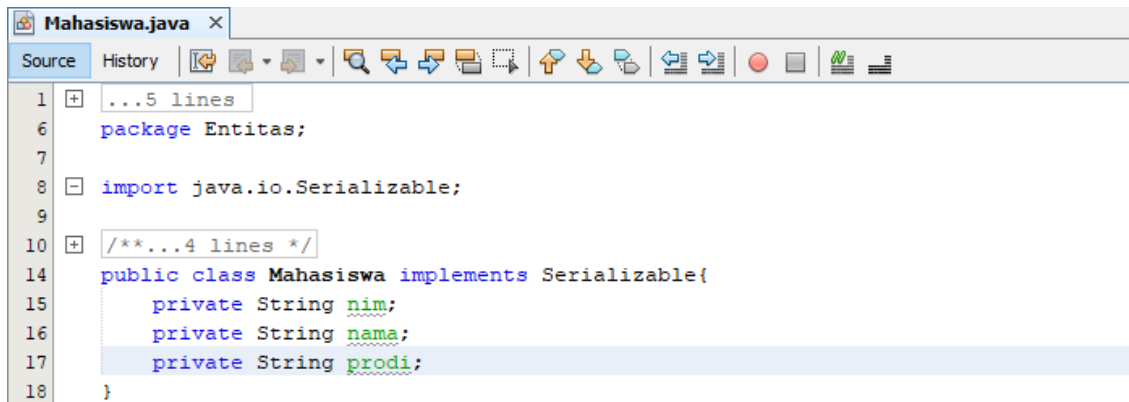
Beri nama *Package* dengan nama *Entitas* -> *Finish*



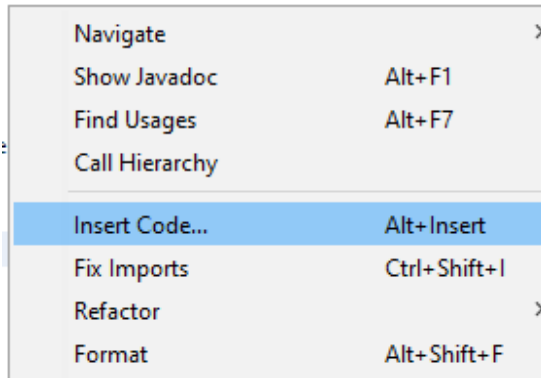
Klik kanan pada *Package* Entitas->New->Java Class. Beri nama kelas misalkan Mahasiswa



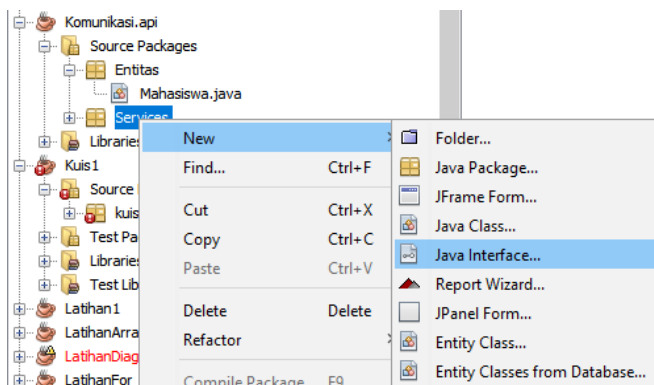
Masukkan *source code* berikut pada kelas Mahasiswa



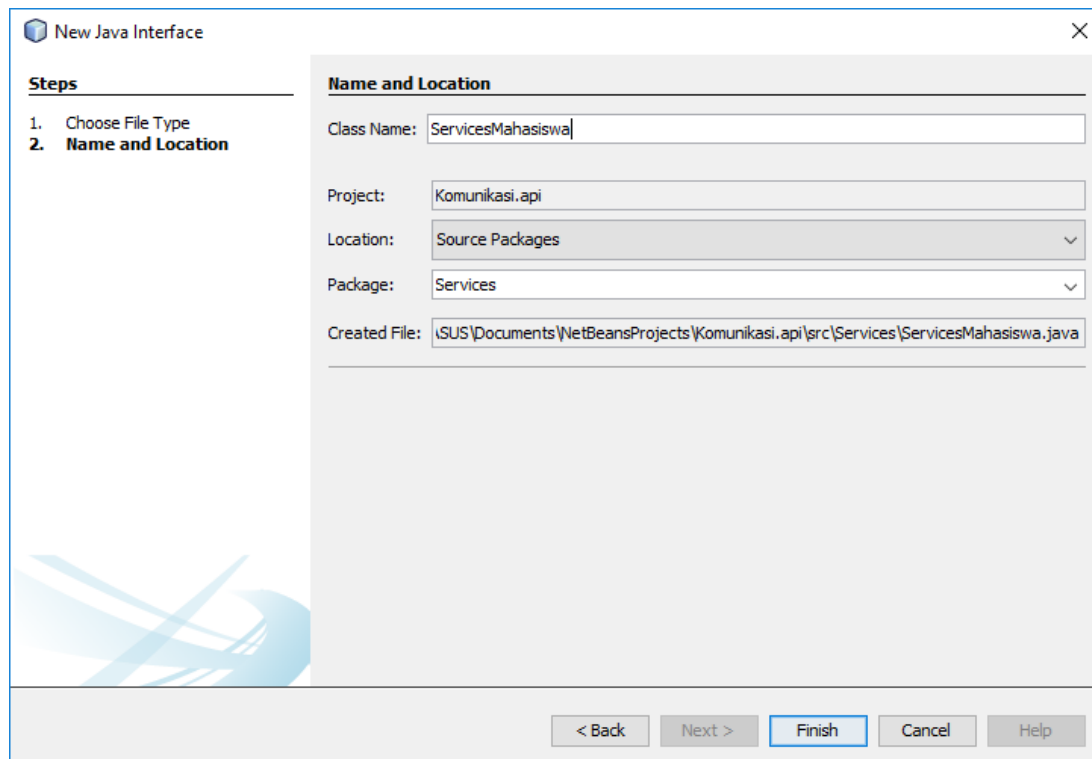
Klik kanan pada kelas kemudian pilih *Insert Code*. Kemudian pilih *Getter and Setter*



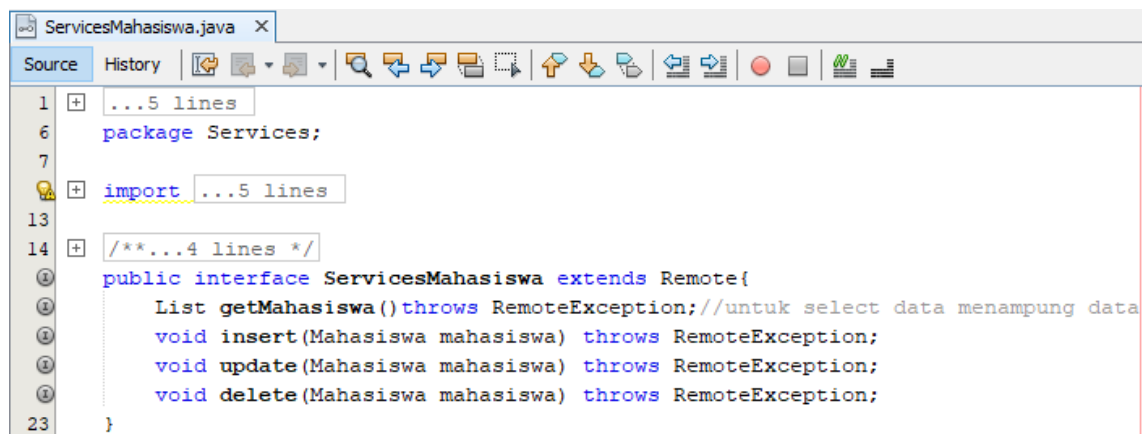
Pada *project* Komunikasi.api, buat *package* baru dengan nama Services. Buat *Interface* baru pada *package* Services klik kanan *New* -> *Java Interface*



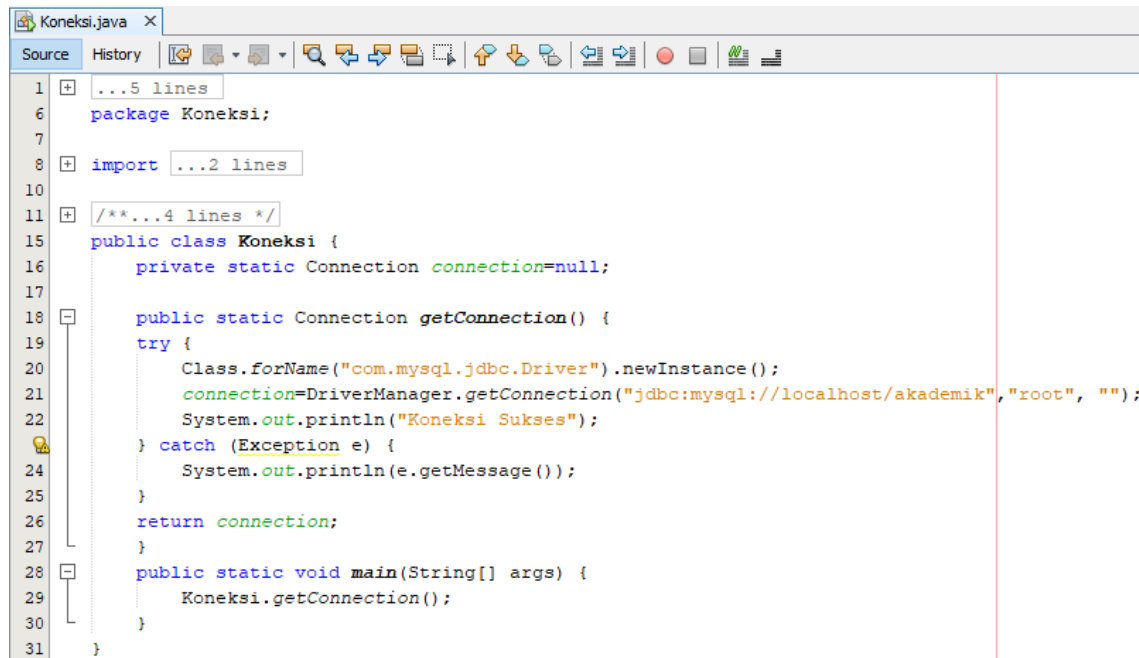
Beri nama *Interface* ServicesMahasiswa -> *Finish*



Masukkan *source code* berikut pada *interface* ServicesMahasiswa

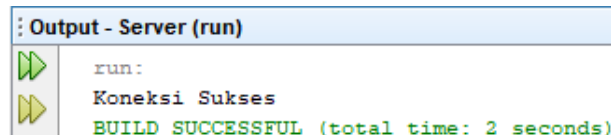


Buat *package* baru pada project Server klik kanan *new*->*Java package* beri nama *Package Koneksi* kemudian buat kelas pada *Package Koneksi* klik kanan *New*->*Java Class* bernama kelas *Koneksi*. Masukkan *source code* berikut pada kelas *Koneksi*



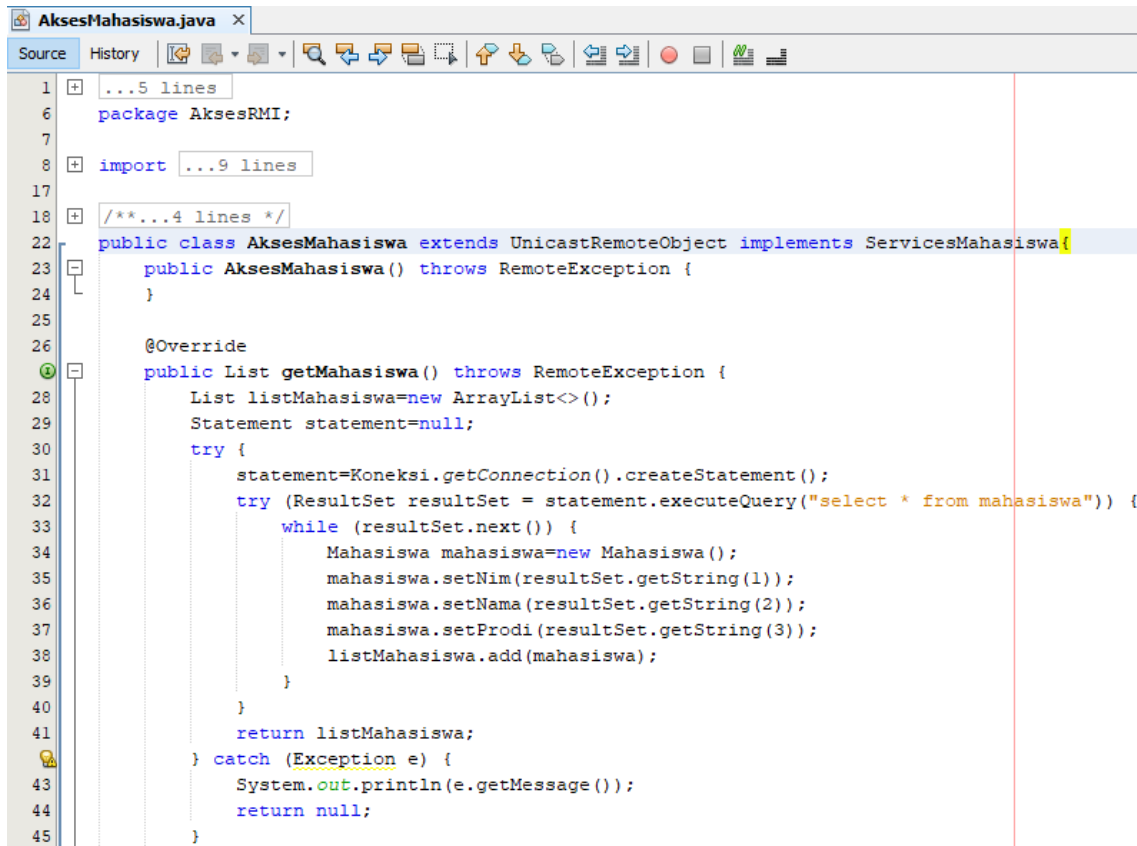
```
1  ...5 lines
6  package Koneksi;
7
8  import ...2 lines
9
10
11  /**...4 lines */
12
13  public class Koneksi {
14      private static Connection connection=null;
15
16      public static Connection getConnection() {
17          try {
18              Class.forName("com.mysql.jdbc.Driver").newInstance();
19              connection=DriverManager.getConnection("jdbc:mysql://localhost/akademik","root", "");
20              System.out.println("Koneksi Sukses");
21          } catch (Exception e) {
22              System.out.println(e.getMessage());
23          }
24          return connection;
25      }
26
27      public static void main(String[] args) {
28          Koneksi.getConnection();
29      }
30  }
```

Jalankan kelas *Koneksi* jika koneksi sukses maka muncul tampilan seperti berikut



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

Buat *package* baru kembali pada *project* Server dengan nama *Package* AksesRMI yang berisi kelas baru AksesMahasiswa. Masukkan *source code* berikut pada kelas AksesMahasiswa



```
1  ...5 lines
6  package AksesRMI;
7
8  import ...9 lines
17
18  /**...4 lines */
22 public class AksesMahasiswa extends UnicastRemoteObject implements ServicesMahasiswa {
23     public AksesMahasiswa() throws RemoteException {
24     }
25
26     @Override
27     public List getMahasiswa() throws RemoteException {
28         List listMahasiswa=new ArrayList<>();
29         Statement statement=null;
30         try {
31             statement=Koneksi.getConnection().createStatement();
32             try (ResultSet resultSet = statement.executeQuery("select * from mahasiswa")) {
33                 while (resultSet.next()) {
34                     Mahasiswa mahasiswa=new Mahasiswa();
35                     mahasiswa.setNim(resultSet.getString(1));
36                     mahasiswa.setNama(resultSet.getString(2));
37                     mahasiswa.setProdi(resultSet.getString(3));
38                     listMahasiswa.add(mahasiswa);
39                 }
40             }
41             return listMahasiswa;
42         } catch (Exception e) {
43             System.out.println(e.getMessage());
44             return null;
45         }
46     }
47 }
```

```

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(Mahasiswa mahasiswa) throws RemoteException {
58         //throw new UnsupportedOperationException("Not supported yet."); //To change body of gen
59         Statement statement=null;
60         try {
61             statement= Koneksi.getConnection().createStatement();
62             statement.executeUpdate("insert into mahasiswa values ('"+mahasiswa.getNim()+"'
63             + ", '"+mahasiswa.getNama()+"', '"+mahasiswa.getProdi()+"')");
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(Mahasiswa mahasiswa) 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 mahasiswa set nama='"+mahasiswa.getNama() +
84             "',prodi='"+mahasiswa.getProdi()+"'where nim='"+mahasiswa.getNim()+"'");
85         } catch (Exception e) {
86             System.out.println(e.getMessage());
87         }
88         finally{
89             if (statement!=null) {
90                 try {
91                     statement.close();
92                 } catch (Exception e) {
93                 }
94             }
95         }
96     }
97
98
99
100     @Override
101     public void delete(Mahasiswa mahasiswa) 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 mahasiswa where nim='"+mahasiswa.getNim()+"'");
107         } catch (Exception e) {

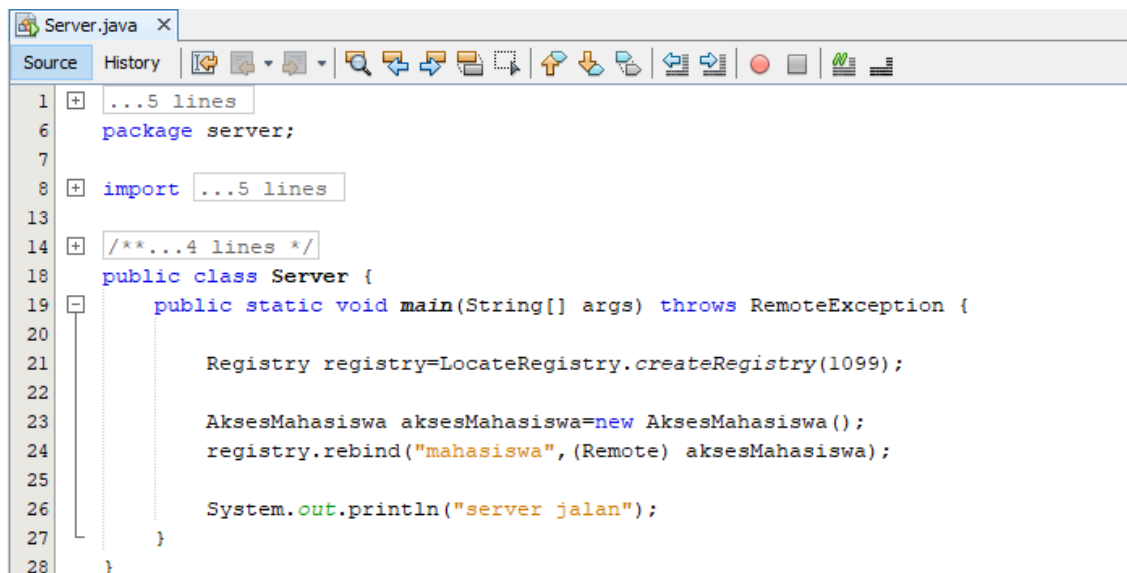
```

```

108         System.out.println(e.getMessage());
109     }finally{
110         if (statement!=null) {
111             try {
112                 statement.close();
113             } catch (Exception e) {
114             }
115         }
116     }
117 }
118 }

```

Buka kelas Server pada *package* Server. Jika belum ada, silahkan buat class Server. Masukkan *source code* berikut pada kelas Server.

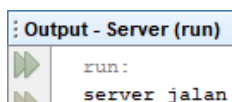


```

1  ...5 lines
6  package server;
7
8  import ...5 lines
13
14  /**...4 lines */
18  public class Server {
19      public static void main(String[] args) throws RemoteException {
20
21          Registry registry=LocateRegistry.createRegistry(1099);
22
23          AksesMahasiswa aksesMahasiswa=new AksesMahasiswa();
24          registry.rebind("mahasiswa",(Remote) aksesMahasiswa);
25
26          System.out.println("server jalan");
27      }
28  }

```

Jalankan Kelas Server. *Server* berhasil dijalankan



```

Output - Server (run)
run:
server jalan

```

Buka *project* Client kemudian buat form dengan nama DataMahasiswa pada *package* Client.
Buatlah desain kelas DataMahasiswa

The screenshot shows the NetBeans IDE with the 'DataMahasiswa.java' file open in the 'Design' view. The form is titled 'Data Mahasiswa' and contains a table with three columns: NIM, NAMA, and PRODI. To the right of the table are three input fields labeled 'NIM :', 'NAMA :', and 'PRODI :'. The 'PRODI' field is a dropdown menu with 'Rekayasa Keamanan Siber' selected. Below these fields are three buttons: 'Insert', 'Update', and 'Delete'.

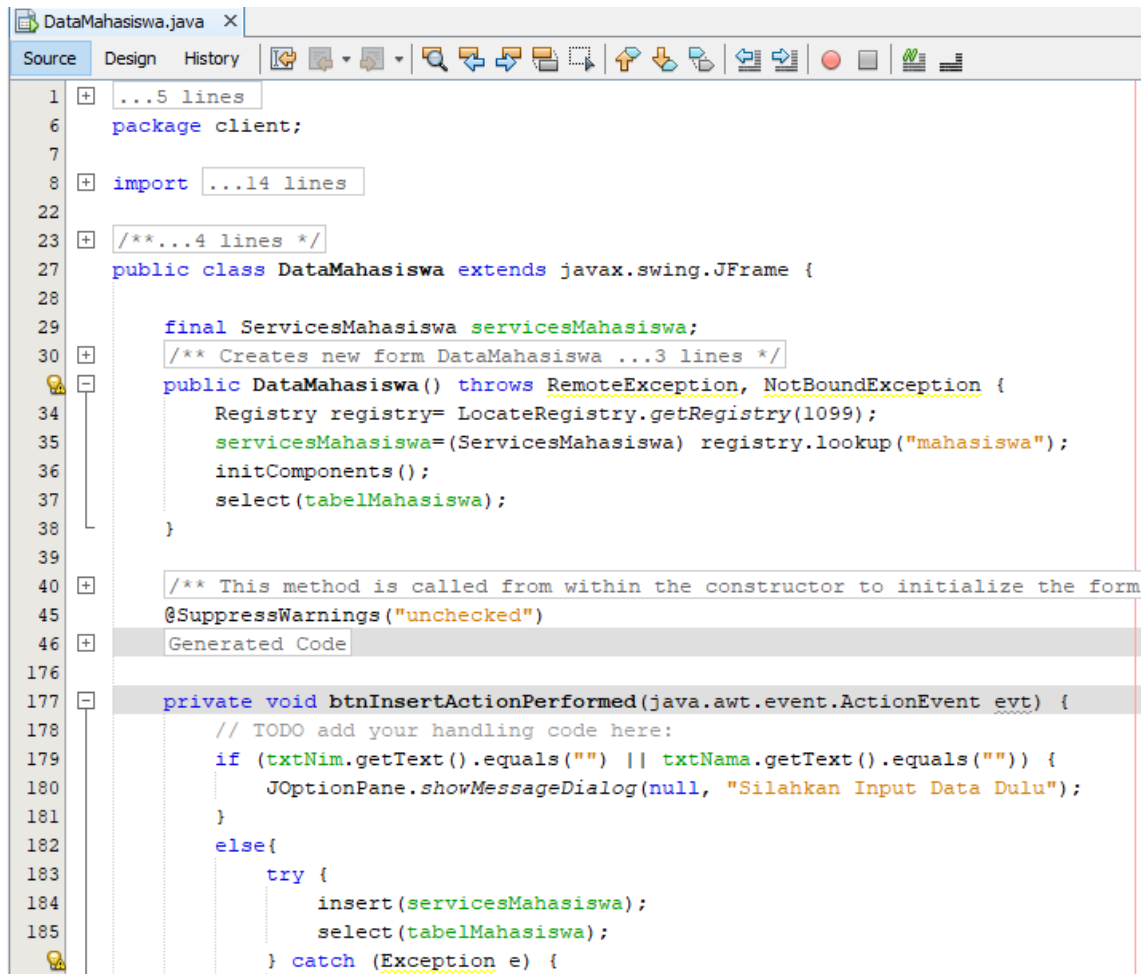
NIM	NAMA	PRODI
-----	------	-------

NIM :

NAMA :

PRODI :

Masukkan *source code* berikut pada kelas DataMahasiswa



```
1  ...5 lines
6  package client;
7
8  import ...14 lines
22
23  /**...4 lines */
27  public class DataMahasiswa extends javax.swing.JFrame {
28
29      final ServicesMahasiswa servicesMahasiswa;
30      /** Creates new form DataMahasiswa ...3 lines */
31      public DataMahasiswa() throws RemoteException, NotBoundException {
32          Registry registry= LocateRegistry.getRegistry(1099);
33          servicesMahasiswa=(ServicesMahasiswa) registry.lookup("mahasiswa");
34          initComponents();
35          select(tabelMahasiswa);
36      }
37
38
39
40      /** This method is called from within the constructor to initialize the form
41      @SuppressWarnings("unchecked")
42      Generated Code
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77  private void btnInsertActionPerformed(java.awt.event.ActionEvent evt) {
78      // TODO add your handling code here:
79      if (txtNim.getText().equals("") || txtNama.getText().equals("")) {
80          JOptionPane.showMessageDialog(null, "Silahkan Input Data Dulu");
81      }
82      else{
83          try {
84              insert(servicesMahasiswa);
85              select(tabelMahasiswa);
86          } catch (Exception e) {
```



```

187         JOptionPane.showMessageDialog(null, e.getMessage());
188     }
189 }
190 clear();
191 }
192
193 private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
194     // TODO add your handling code here:
195     if (txtNim.getText().equals("") || txtNama.getText().equals("")) {
196         JOptionPane.showMessageDialog(null, "Silahkan pilih data yang akan di Update");
197     }
198     else{
199         try {
200             update(servicesMahasiswa);
201             select(tabelMahasiswa);
202         } catch (Exception e) {
203             JOptionPane.showMessageDialog(null, e.getMessage());
204         }
205         clear();
206     }
207 }
208
209 private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
210     // TODO add your handling code here:
211     if (txtNim.getText().equals("") || txtNama.getText().equals("")) {
212         JOptionPane.showMessageDialog(null, "Silahkan pilih data yang akan di Delete");
213     }
214     else{
215         try {
216             delete(servicesMahasiswa);
217             select(tabelMahasiswa);
218         } catch (Exception e) {
219             JOptionPane.showMessageDialog(null, e.getMessage());
220         }
221         clear();
222     }
223 }
224
225 private void tabelMahasiswaMouseClicked(java.awt.event.MouseEvent evt) {
226     // TODO add your handling code here:
227     txtNim.setText(tabelMahasiswa.getValueAt(tabelMahasiswa.getSelectedRow(),0).toString());
228     txtNama.setText(tabelMahasiswa.getValueAt(tabelMahasiswa.getSelectedRow(),1).toString());
229     Prodi.setSelectedItem(tabelMahasiswa.getValueAt(tabelMahasiswa.getSelectedRow(),2).toString());
230 }
231
232 private void select (javax.swing.JTable tabelMahasiswa) throws RemoteException {
233     // begin
234     String header[]={" NIM ", " NAMA ", " PRODI "};
235     List list = servicesMahasiswa.getMahasiswa();
236     Object data[][] = new Object[list.size()][3];
237     int x = 0;
238     for (Iterator it = list.iterator(); it.hasNext();) {
239         Mahasiswa m = (Mahasiswa) it.next();
240         data[x][0] = m.getNim();
241         data[x][1] = m.getNama();
242         data[x][2] = m.getProdi();
243         ++x;
244     }
245     tabelMahasiswa.setModel(new DefaultTableModel(data, header));
246     JTableHeader head = tabelMahasiswa.getTableHeader();
247     head.setBackground(Color.orange);
248     // end

```

```

249     }
250
251     private void insert(ServicesMahasiswa servicesMahasiswa) throws RemoteException{
252         Mahasiswa m=new Mahasiswa();
253         m.setNim(txtNim.getText());
254         m.setNama(txtNama.getText());
255         m.setProdi(Prodi.getSelectedItem().toString());
256         servicesMahasiswa.insert(m);
257     }
258
259     private void update(ServicesMahasiswa servicesMahasiswa) throws RemoteException{
260         Mahasiswa m=new Mahasiswa();
261         m.setNim(txtNim.getText());
262         m.setNama(txtNama.getText());
263         m.setProdi(Prodi.getSelectedItem().toString());
264         servicesMahasiswa.update(m);
265     }
266
267     private void delete(ServicesMahasiswa servicesMahasiswa) throws RemoteException{
268         Mahasiswa m=new Mahasiswa();
269         m.setNim(txtNim.getText());
270         servicesMahasiswa.delete(m);
271     }
272
273     private void clear(){
274         txtNim.setText("");
275         txtNama.setText("");
276     }
277
278     public static void main(String args[]) {
279         /* Set the Nimbus look and feel */
280         Look and feel setting code (optional)
281
282         /* Create and display the form */
283         java.awt.EventQueue.invokeLater(new Runnable() {
284             public void run() {
285                 try {
286                     new DataMahasiswa().setVisible(true);
287                 } catch (RemoteException ex) {
288                     Logger.getLogger(DataMahasiswa.class.getName()).log(Level.SEVERE, null, ex);
289                 } catch (NotBoundException ex) {
290                     Logger.getLogger(DataMahasiswa.class.getName()).log(Level.SEVERE, null, ex);
291                 }
292             }
293         });
294     }
295
296     // Variables declaration - do not modify
297     private javax.swing.JComboBox<String> Prodi;
298     private javax.swing.JButton btnDelete;
299     private javax.swing.JButton btnInsert;
300     private javax.swing.JButton btnUpdate;
301     private javax.swing.JLabel jLabel1;
302     private javax.swing.JLabel jLabel2;
303     private javax.swing.JLabel jLabel3;
304     private javax.swing.JLabel jLabel4;
305     private javax.swing.JScrollPane jScrollPane1;
306     private javax.swing.JTable tabelMahasiswa;
307     private javax.swing.JTextField txtNama;
308     private javax.swing.JTextField txtNim;
309     // End of variables declaration
310 }

```

Jalankan program. Program akan menampilkan sebagai berikut

NIM	NAMA	PRODI
-----	------	-------

NIM :

NAMA :

PRODI :

Masukkan data dan tekan *button* Insert untuk memasukkan data

NIM	NAMA	PRODI
1817101...	Alya	Rekayas...

NIM :

NAMA :

PRODI :

Program akan menampilkan sebagai berikut

NIM	NAMA	PRODI
1817101...	Alya	Rekayas...

NIM :

NAMA :

PRODI :

Ubah data yang terdapat di program dan tekan *button* Update untuk mengubah data

NIM	NAMA	PRODI
1817101...	Alya	Rekayas...

NIM :

NAMA :

PRODI :

Program akan menampilkan sebagai berikut

NIM	NAMA	PRODI
1817101...	Alya Aiman	Rekayas...

NIM :

NAMA :

PRODI :

Hapus data yang terdapat di program dan tekan *button* Delete untuk menghapus data

NIM	NAMA	PRODI
1817101...	Alya Aiman	Rekayas...

NIM :

NAMA :

PRODI :

Program akan menampilkan sebagai berikut

NIM	NAMA	PRODI
-----	------	-------

NIM :

NAMA :

PRODI :