



KOMUNIKASI DENGAN APLIKASI DESKTOP BERBASIS JARINGAN



I Komang Setia Buana

Client/Server dengan Java Remote Method Invocation (Java RMI)

- RMI menyediakan sarana dimana client dan server dapat berkomunikasi dan saling bertukar informasi
- RMI memungkinkan pengembang perangkat lunak untuk merancang aplikasi terdistribusi dimana methods dari remote object dapat dipanggil dari JVM (Java Virtual Machine) lain, yang mungkin berjalan pada host yang berbeda.
- Dalam RMI, server akan membuat remote objects, membuat referensi, dan menunggu client untuk memanggil methods dari remote object ini. Sedangkan client akan mendapatkan remote reference dari satu atau lebih remote object dan memanggil methods untuk remote object tersebut.

Praktikum

- Buat database “akademik” dan table “mahasiswa” dengan struktur tabel

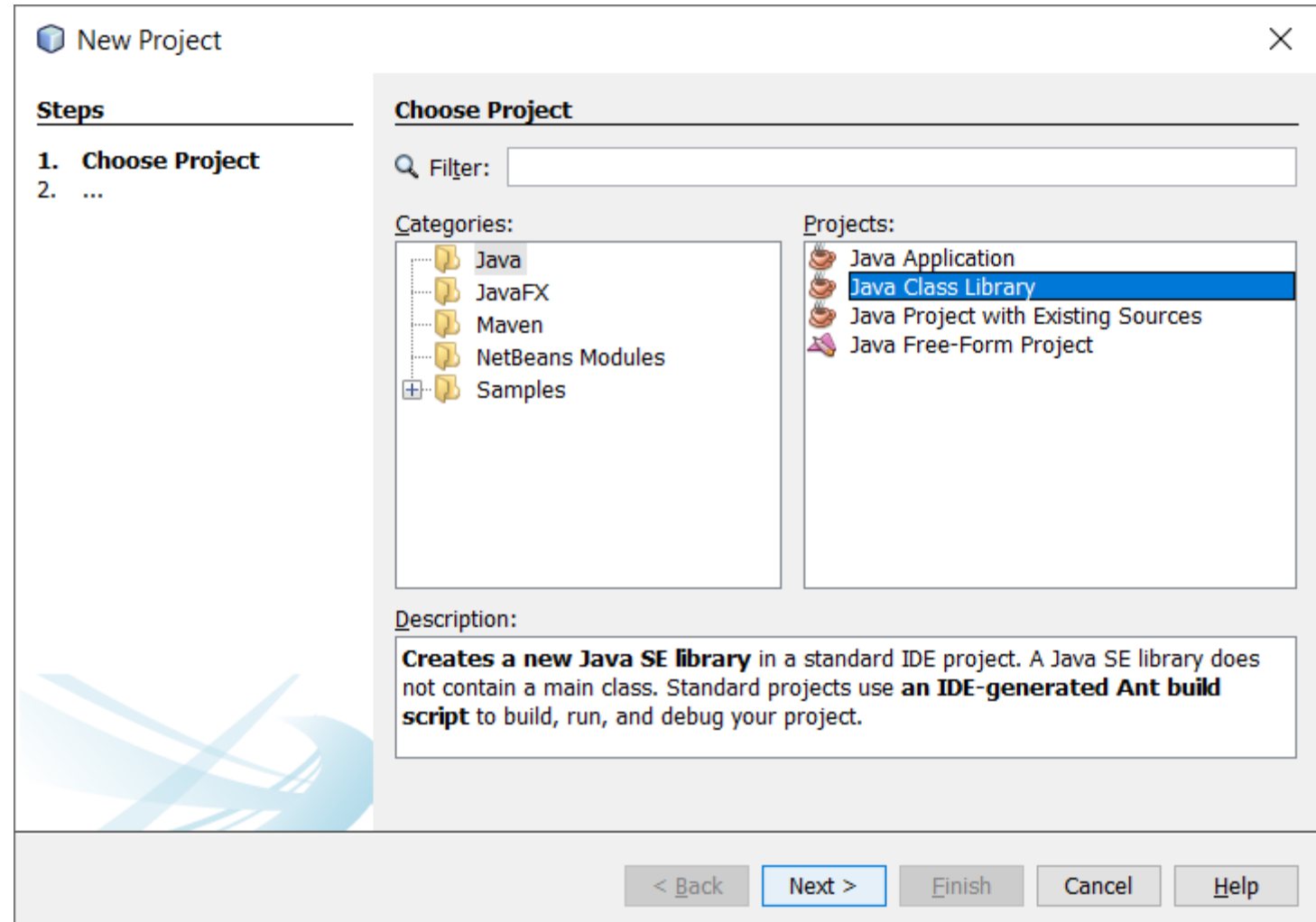
```
mysql> desc mahasiswa;
```

Field	Type	Null	Key	Default	Extra
Nim	varchar(30)	NO	PRI	NULL	
nama	varchar(100)	NO		NULL	
prodi	varchar(50)	NO		NULL	

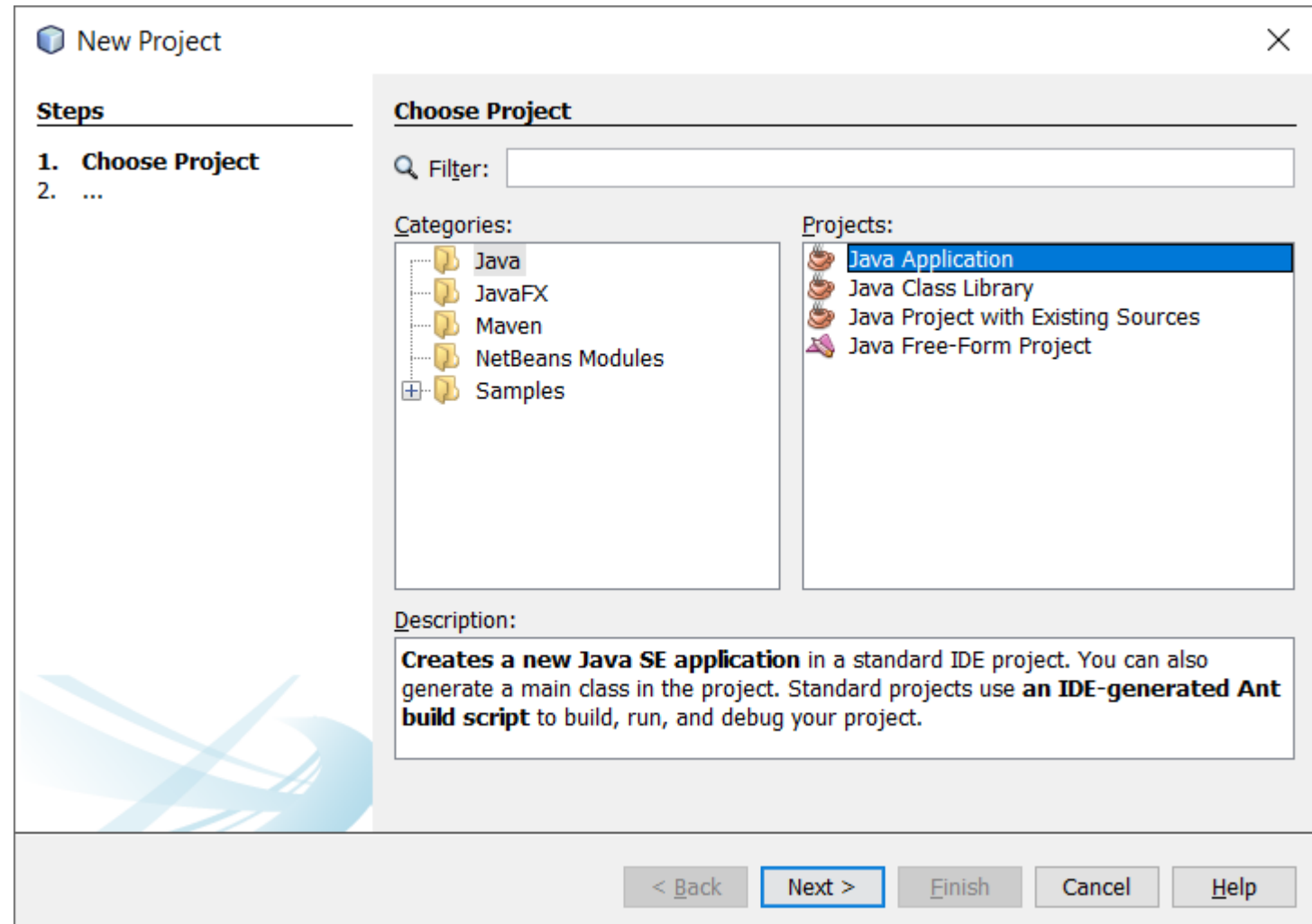
```
3 rows in set (0.02 sec)
```

Praktikum

- Pertama kita jalankan Netbeans kemudian buat Project terlebih dahulu **New Project->Java Class Library** seperti pada gambar disamping
- Pilih **Next** kemudian berinama Java library pada project Name **Komunikasi.api** sebagai penghubung antara client dan server



- Setelah membuat project **Komunikasi.api** kemudian kita buat kembali project baru **New Project->Java Application** kemudian **Next** seperti pada gambar di samping
- Setelah kita **Next**, beri nama project yang dibuat dengan **Server**

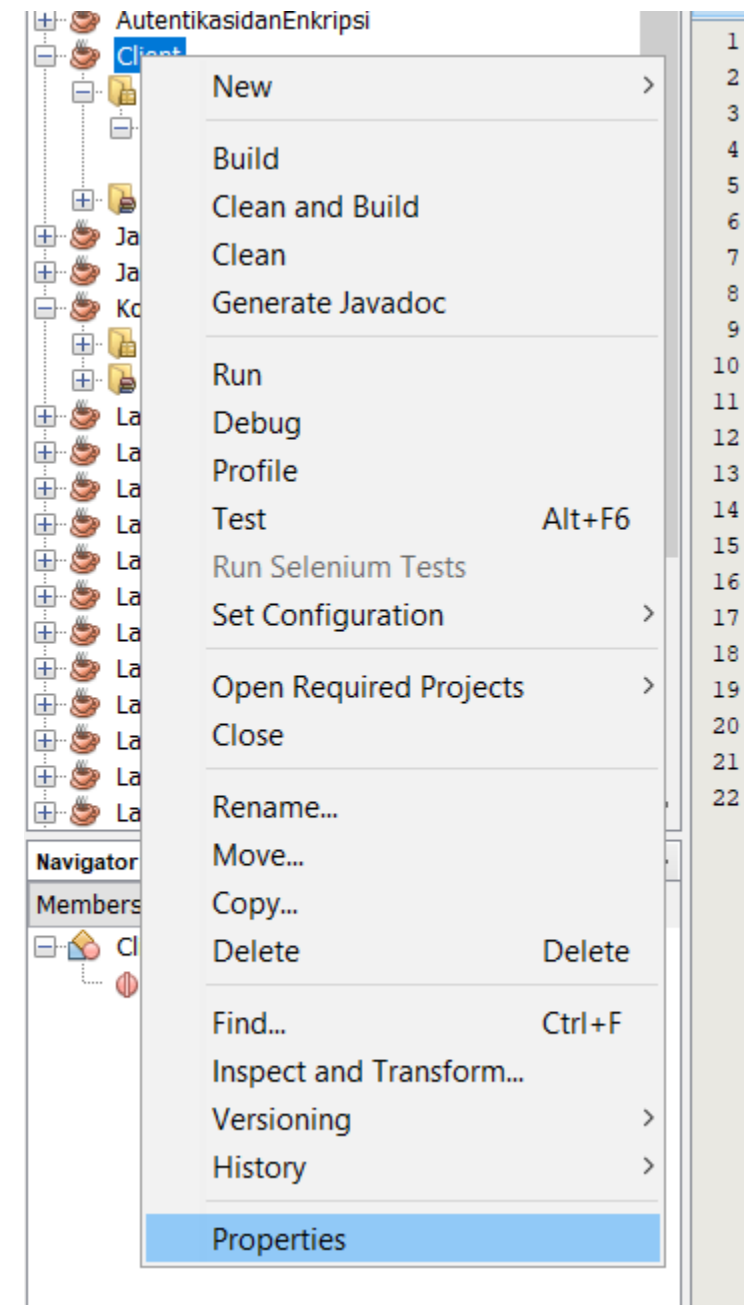


Praktikum

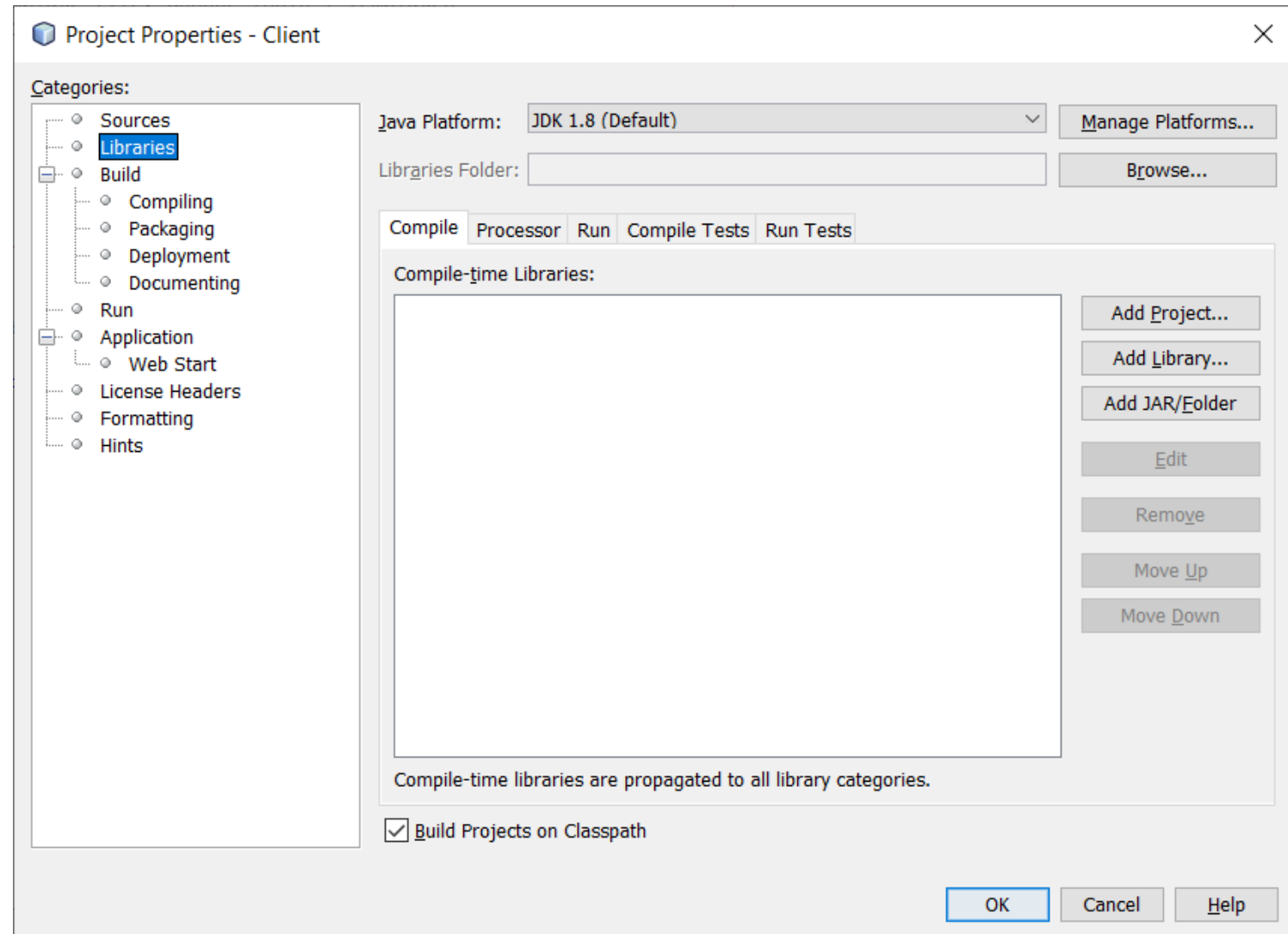
- Kemudian kita buat lagi kembali project baru **New Project->Java Application** dengan nama **Client**
- Kita sudah punya 3 project dengan nama **Client**, **Server** dan **Komunikasi.api**

Praktikum

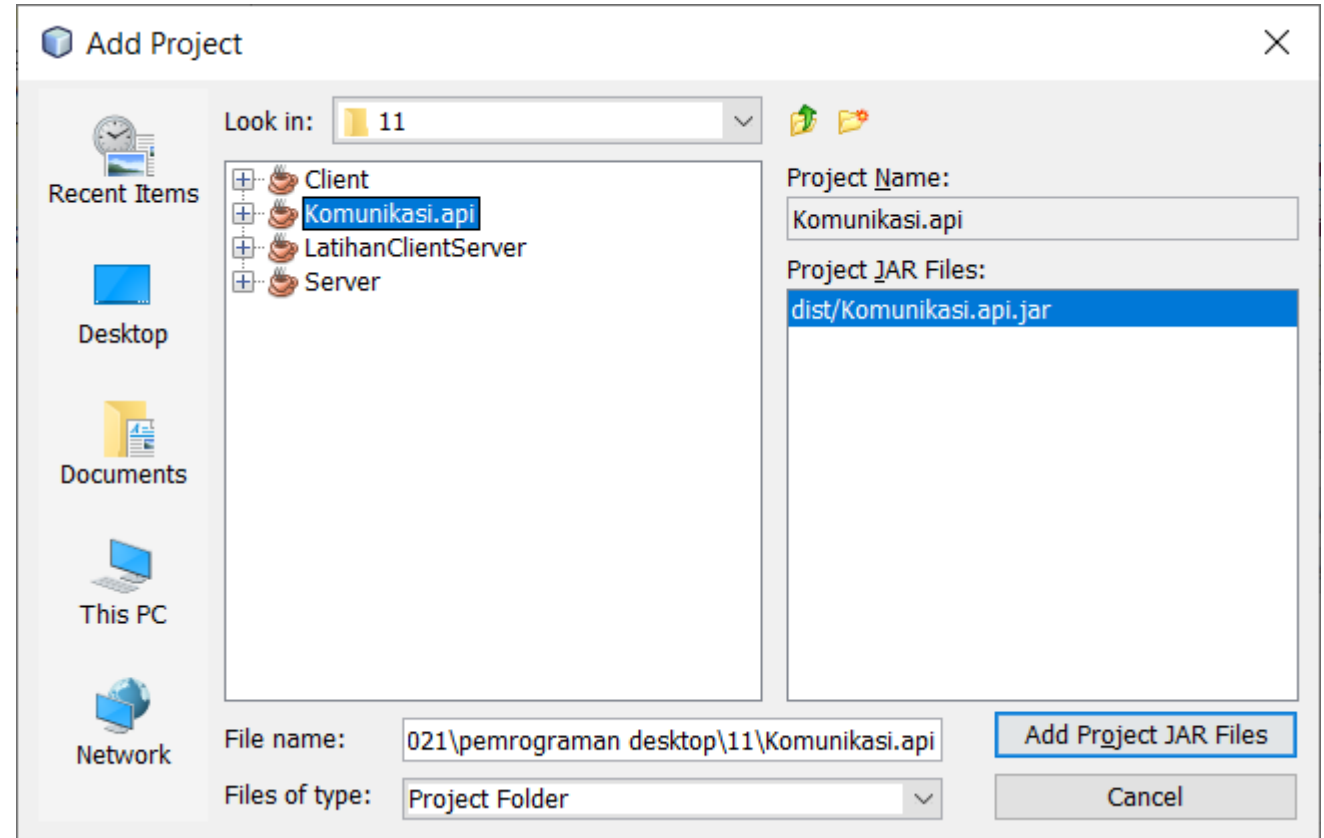
- Selanjutnya untuk menghubungkan antara client dan server klik kanan pada project Client kemudian pilih properties seperti pada contoh gambar di Disamping



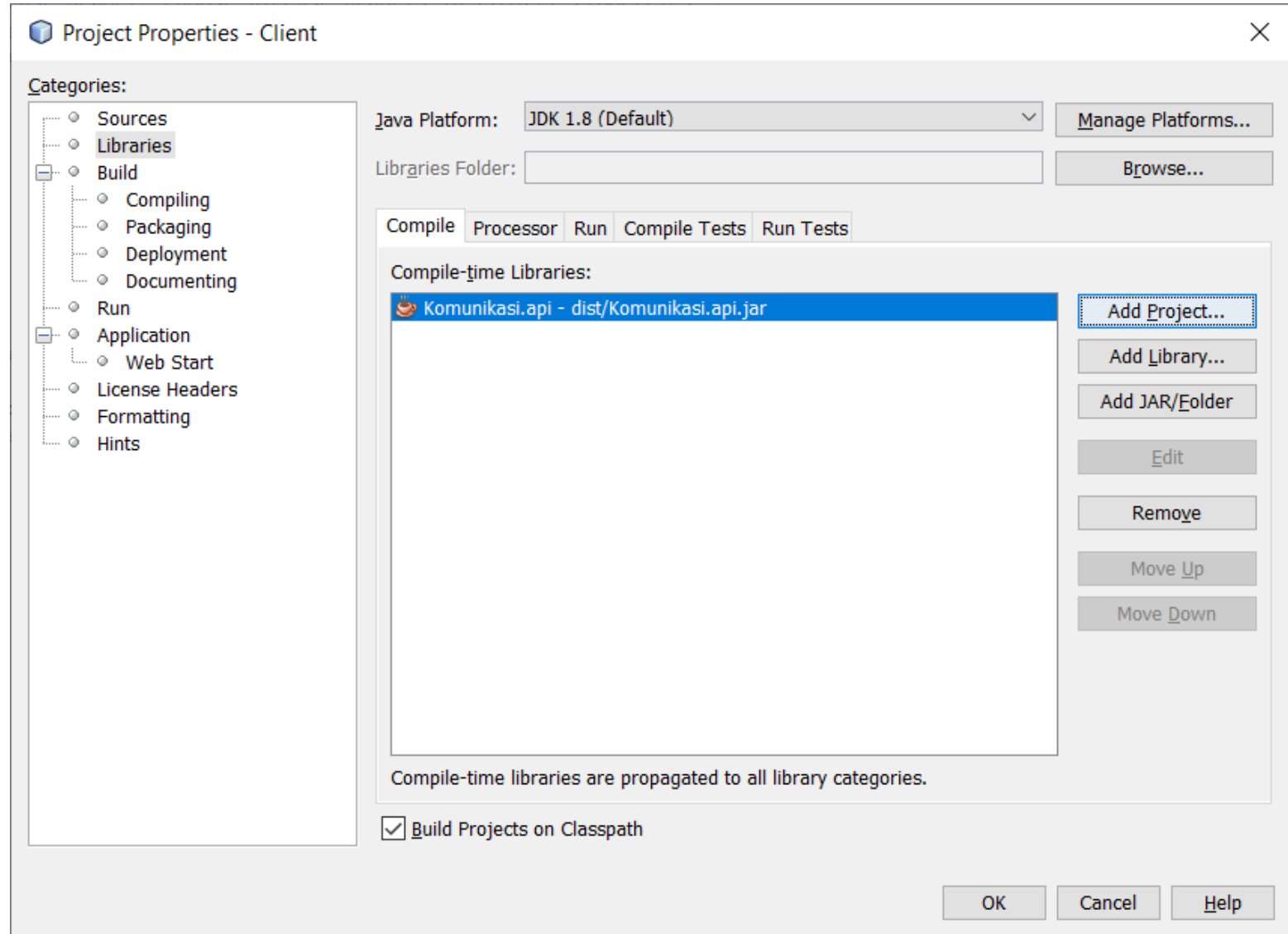
- Setelah pilih properties maka akan muncul tampilan properties pilih **libraries -> add Project**



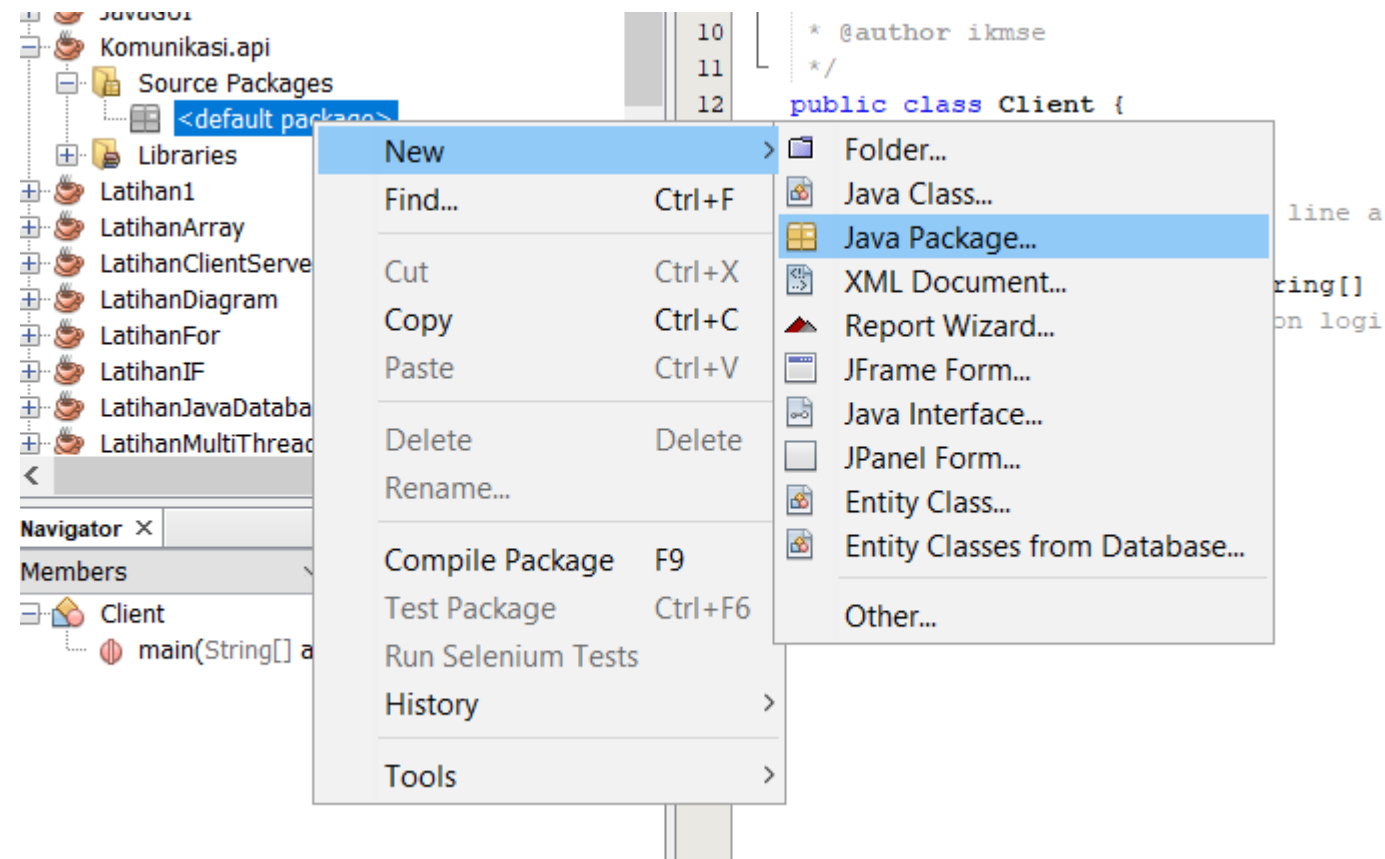
- Kemudian pilih project **Komunikasi.api** yang sudah kita buat sebelumnya kemudian pilih **Add Project JAR Files** seperti pada contoh gambar di samping



- Setelah muncul seperti gambar disamping silahkan klik OK



- Lakukan Juga pada project Server pilih **Properties->Libraries->Add JAR Files->Komunikasi-api->OK** sama seperti pada Client
- Kemudian pada **Komunikasi.api** pada **Source Package->Default Package** klik kanan **New->Java Package** seperti pada gambar di samping



- Kemudian beri nama Package dengan nama **Entitas -> Finish**

New Java Package

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Package Name:

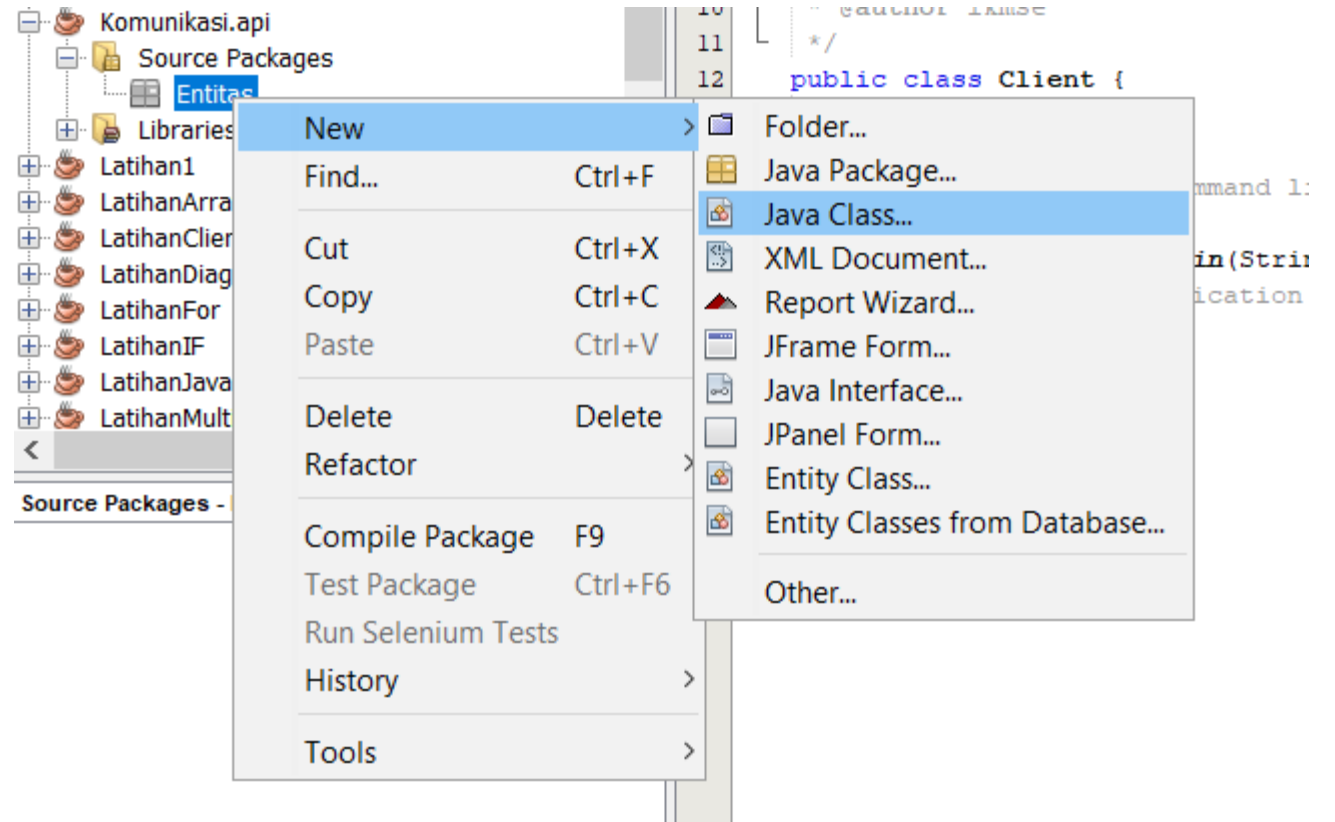
Project:

Location:

Created Folder:

< Back Next > **Finish** Cancel Help

- Kemudian klik kanan pada Package Entitas->New->Java Class
- Berinama Java class Misalkan Mahasiswa



```

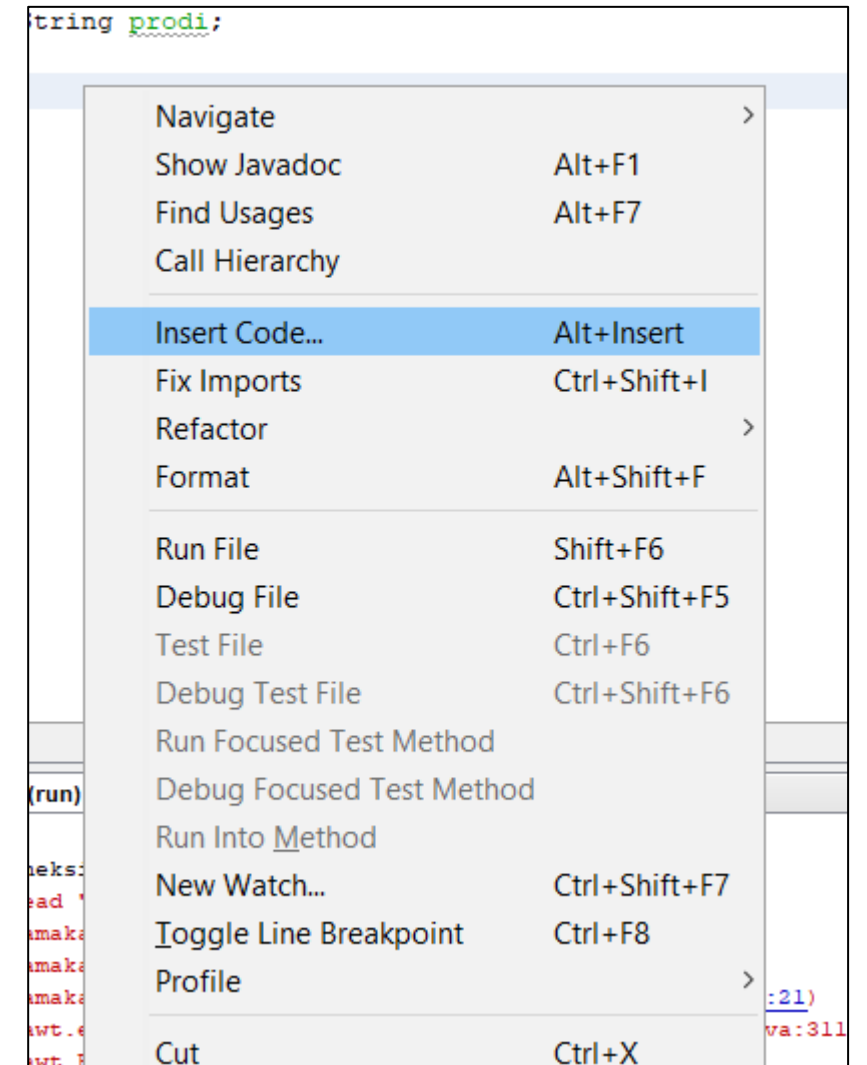
package Entitas;

import java.io.Serializable;

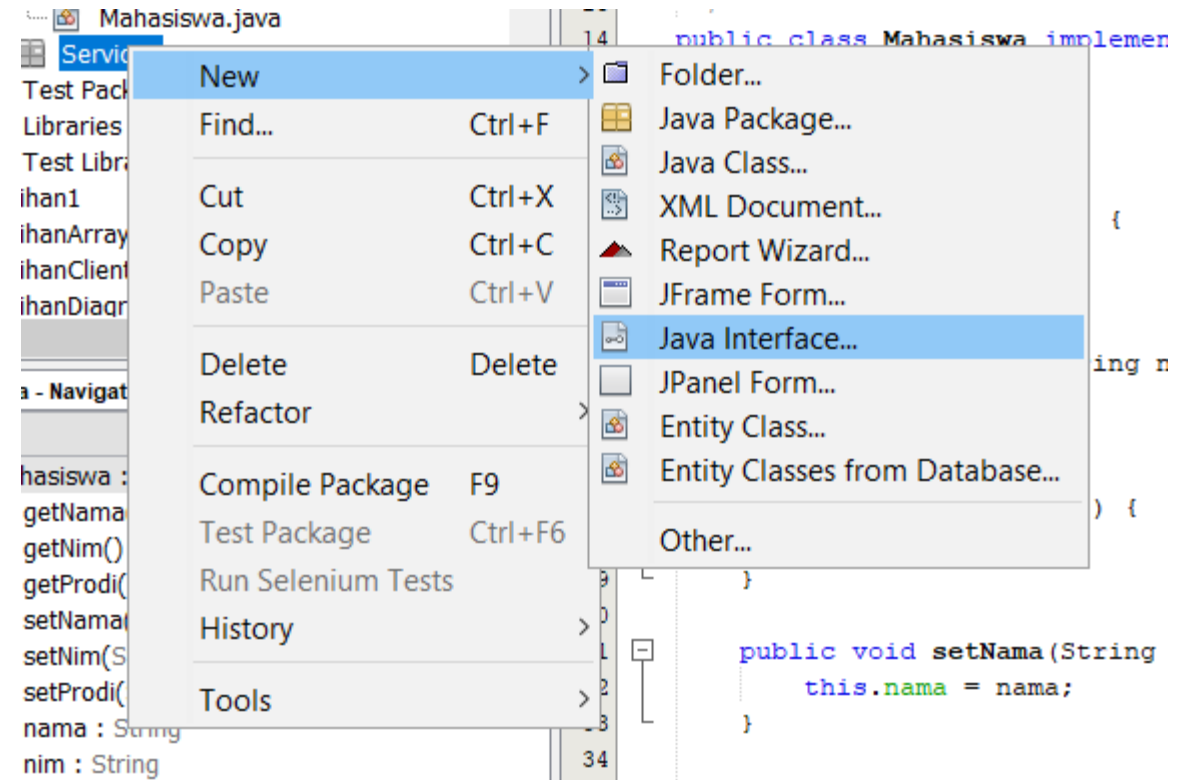
/**
 *
 * @author ikmse
 */
public class Mahasiswa implements Serializable{
    private String nim;
    private String nama;
    private String prodi;

```

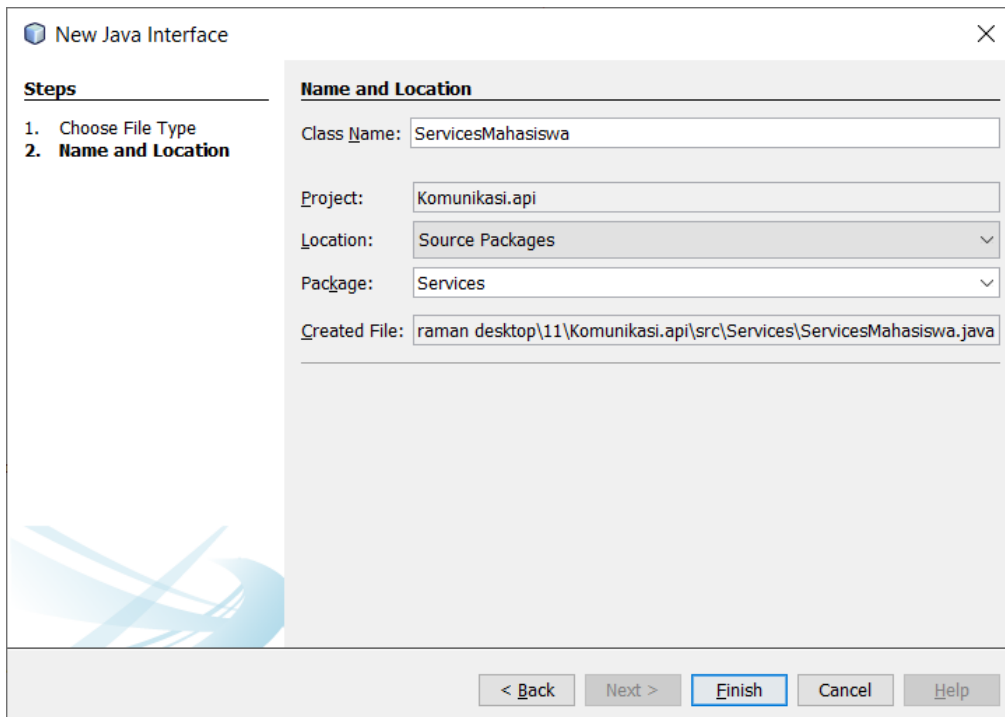
- Klik kanan pada class kemudian pilih Insert Code
- Kemudian pilih **Getter and Setter**



- Masih di 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
- Kemudian ketikkan kode disamping



```
package Services;

import Entitas.Mahasiswa;
import java.rmi.Remote;
import java.rmi.RemoteException;
import java.util.List;

/**
 *
 * @author ikmse
 */
public interface ServicesMahasiswa extends Remote {
    List getMahasiswa() throws RemoteException; // untuk select data menampung data mahasiswa
    void insert(Mahasiswa mahasiswa) throws RemoteException;
    void update(Mahasiswa mahasiswa) throws RemoteException;
    void delete(Mahasiswa mahasiswa) throws RemoteException;
}
```


- Setelah selesai menuliskan Script di atas kemudian buat package baru pada **project Server** klik kanan **new->Java package** beri nama Package **Koneksi** kemudian buat Class pada Package **Koneksi** klik kanan **New->Java Class** berinama **class Koneksi** kemudian tuliskan script berikut pada **class Koneksi**.

```
package Koneksi;

import java.sql.Connection;
import java.sql.DriverManager;

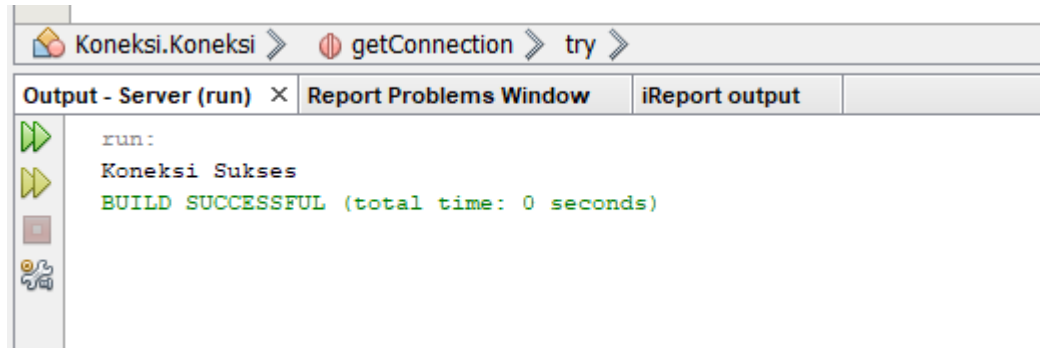
/**
 *
 * @author ikmse
 */
public class Koneksi {
    private static Connection connection=null;

    public static Connection getConnection() {
        try {
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            connection=DriverManager.getConnection("jdbc:mysql://localhost:3306/akademik","root","");
            System.out.println("Koneksi Sukses");
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
        return connection;
    }
    public static void main(String[] args) {

        Koneksi.getConnection();

    }
}
```

- Kemudian Run/jalankan **class Koneksi** jika muncul output seperti pada gambar di bawah maka koneksi Sukses.



- Setelah membuat koneksi kemudian kita buat **package baru** kembali pada **project Server** dengan nama Package **AksesRMI** yang berisi class baru **AksesMahasiswa**
- Untuk source code **class AksesMahasiswa** silahkan download “Pertemuan 11 - class AksesMahasiswa.pdf”
- Setelah selesai membuat **class AksesMahasiswa** kemudian buka **class Server** pada packge **Server**. Jika belum ada silahkan buat **class Server**

- Untuk source code **class Server** silahkan ketikkan code disamping

```
package Server;

import AksesRMI.AksesMahasiswa;
import java.rmi.Remote;
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

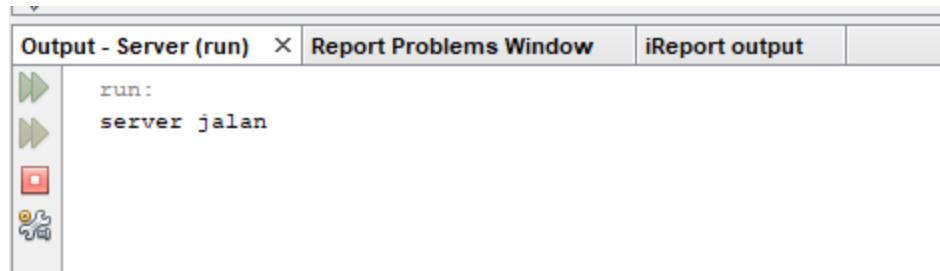
/**
 *
 * @author ikmse
 */
public class Server {
    public static void main(String[] args) throws RemoteException {

        Registry registry=LocateRegistry.createRegistry(1099);

        AksesMahasiswa aksesMahasiswa=new AksesMahasiswa();
        registry.rebind("mahasiswa",(Remote) aksesMahasiswa);

        System.out.println("server jalan");
    }
}
```

- Silahkan jalankan class Server, server berhasil dijalankan ketika muncul tampilan seperti dibawah



- Setelah selesai pada **project Server** selanjutnya buka **project Client** buat form dengan nama **DataMahasiswa** pada **package Client**

Aplikasi Client Server


Data Mahasiswa

NIM	NAMA	PRODI
1	diah	Teknik Komputer

NIM :

NAMA :

PRODI :



Aplikasi Client Server

- Sebelum membuat code untuk **insert, update dan delete**, silahkan ketikkan code yang ada di file “Pertemuan 11-class DataMahasiswa.pdf” di **class DataMahasiswa**

- Code insert

```
private void btnInsertActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:
    if (txtNim.getText().equals("") || txtNama.getText().equals(""))
    {
        JOptionPane.showMessageDialog(null, "Silahkan Input Data
Dulu");

    }
    else {
        try {
            insert(servicesMahasiswa);
            select(tabelMahasiswa);
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
        }
    }
    clear();
}
```


- Code Update

```
private void btnUpdateActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:
    if (txtNim.getText().equals("") || txtNama.getText().equals(""))
    {
        JOptionPane.showMessageDialog(null, "Silahkan pilih data yang
akan di Update");

    }
    else {
        try {
            update(servicesMahasiswa);
            select(tabelMahasiswa);
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e.getMessage());
        }
        clear();
    }
}
```

- Code Delete

```
private void btnDeleteActionPerformed(java.awt.event.ActionEvent  
evt) {  
    // TODO add your handling code here:  
    if (txtNim.getText().equals("") || txtNama.getText().equals(""))  
    {  
        JOptionPane.showMessageDialog(null, "Silahkan pilih data yang  
akan di Delete");  
  
    }  
    else {  
        try {  
            delete(servicesMahasiswa);  
            select(tabelMahasiswa);  
        } catch (Exception e) {  
            JOptionPane.showMessageDialog(null, e.getMessage());  
        }  
        clear();  
    }  
}
```

- Untuk Tabel di **Event->MouseClicked** kemudian tuliskan script berikut:

```
private void tabelMahasiswaMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
  
    txtNim.setText(tabelMahasiswa.getValueAt(tabelMahasiswa.getSelectedRow(),0).toString());  
  
    txtNama.setText(tabelMahasiswa.getValueAt(tabelMahasiswa.getSelectedRow(),1).toString());  
  
    Prodi.setSelectedItem(tabelMahasiswa.getValueAt(tabelMahasiswa.getSelectedRow(),2).toString());  
}
```

- Method main

```
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(DataMahasiswa.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(DataMahasiswa.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(DataMahasiswa.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(DataMahasiswa.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() {
            try {
                new DataMahasiswa().setVisible(true);
            } catch (RemoteException ex) {
                Logger.getLogger(DataMahasiswa.class.getName()).log(Level.SEVERE, null, ex);
            } catch (NotBoundException ex) {
                Logger.getLogger(DataMahasiswa.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    });
}
```

- Kemudian jalankan form **DataMahasiswa** jangan lupa untuk menjalankan servernya terlebih dahulu jika servernya tidak di jalankan maka akan **error**

Tugas

1. Deskripsikan hasil praktikum yang dilakukan ke dalam laporan!