

Nama : Muhammad Ridho

Kelas : 4IA06

NPM : 51421055

Materi : Konsep Dasar Object Relation Mapping(ORM) Dan Framework Hibernate

Mata Praktikum : Rekayasa Perangkat Lunak 2

ACTIVITY 4

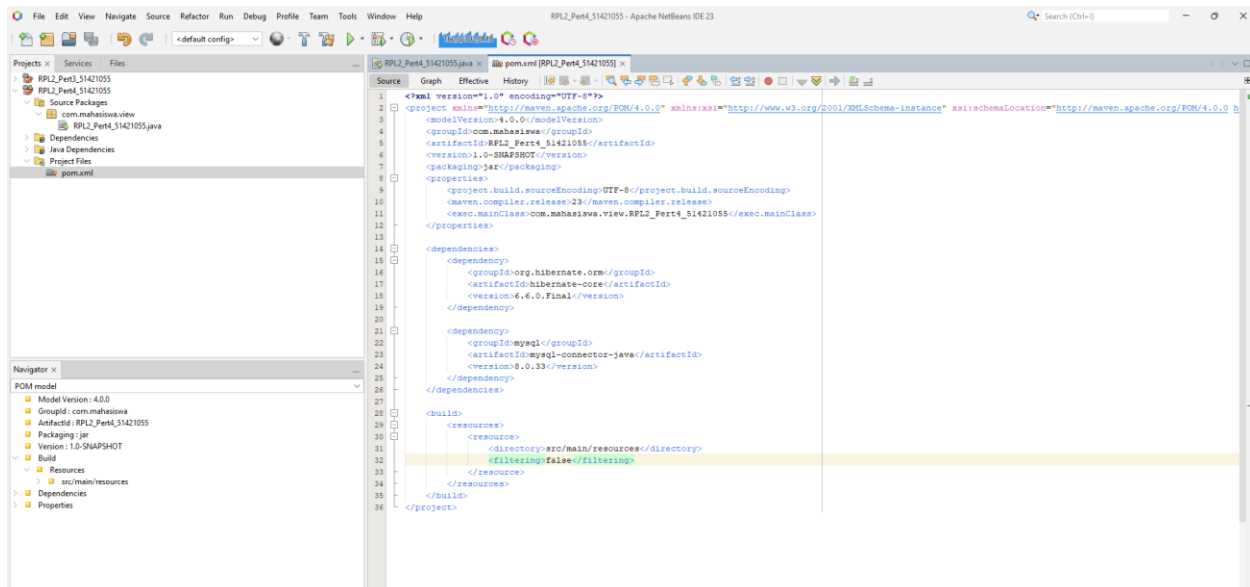
SOAL

1. Sebutkan dan jelaskan semua komponen GUI yang digunakan pada JFrame “MahasiswaView.java”.
2. Screenshoot hasil codingan kalian lengkap dengan window projects, navigator dan ouputnya.

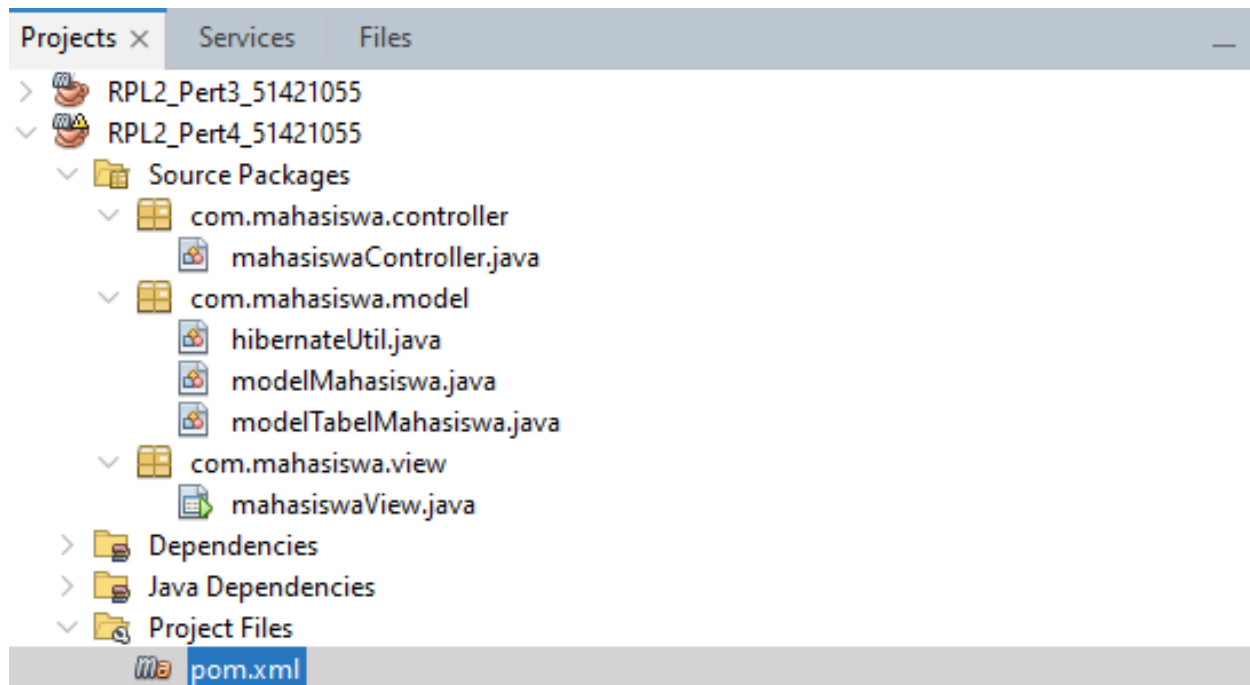
JAWABAN

1. JLabel = untuk membuat label seperti npm, nama, semester, dan IPK
JTextField = untuk membuat input dari data npm, nama, semester, dan IPK
JButton = untuk membuat tombol untuk input dan hapus data npm, nama, semester, dan IPK
dataTable = untuk membuat table yang berisikan data dari npm, nama, semester, dan IPK
- 2.

Pom.xml

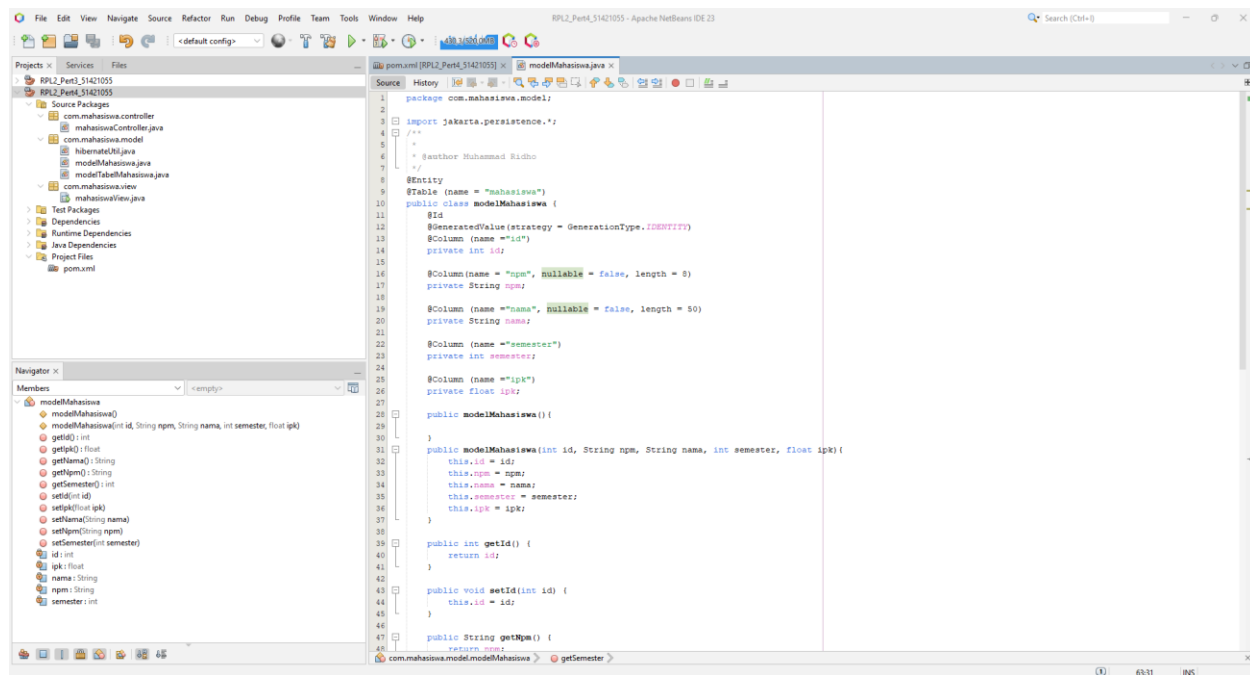


Setelah Dibuat projectnya kita buat dependencies pada pom.xml.



Setelah menambah dependencies pada pom.xml kita membuat java package dan java class sesuai keperluan dan tidak pula pada package com.mahasiswa.view kita jadikan JFrame agar terlihat interfacenya.

modalMahasiswa.java



Sourcecode:

package com.mahasiswa.model;

```

import jakarta.persistence.*;

/**
 *
 * @author Muhammad Ridho
 */
@Entity
@Table (name = "mahasiswa")
public class modelMahasiswa {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column (name = "id")
    private int id;


    @Column(name = "npm", nullable = false, length = 8)
    private String npm;


    @Column (name = "nama", nullable = false, length = 50)
    private String nama;


    @Column (name = "semester")
    private int semester;


    @Column (name = "ipk")
    private float ipk;

    public modelMahasiswa(){

    }

    public modelMahasiswa(int id, String npm, String nama, int semester, float ipk){

        this.id = id;

        this.npm = npm;

```

```
        this.nama = nama;

        this.semester = semester;

        this.ipk = ipk;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getNpm() {
        return npm;
    }

    public void setNpm(String npm) {
        this.npm = npm;
    }

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public int getSemester() {
```

```

        return semester;
    }

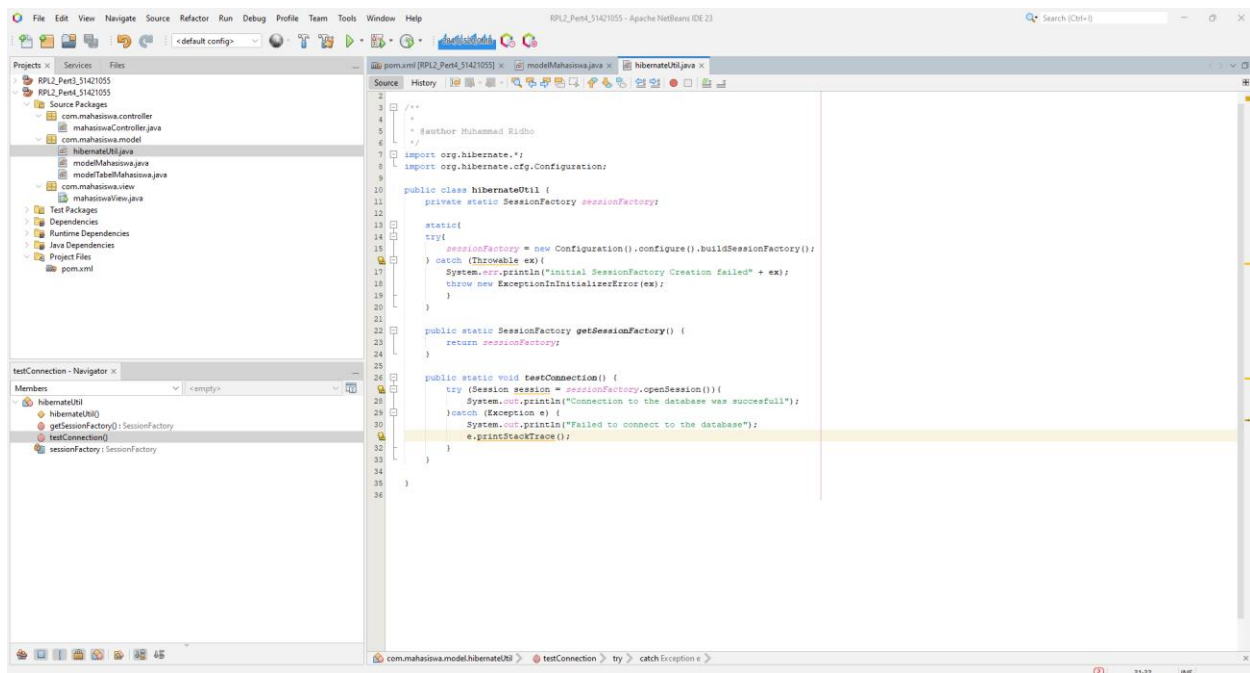
    public void setSemester(int semester) {
        this.semester = semester;
    }

    public float getIpk() {
        return ipk;
    }

    public void setIpk(float ipk) {
        this.ipk = ipk;
    }
}

```

hibernateUtil.java



Sourcecode:

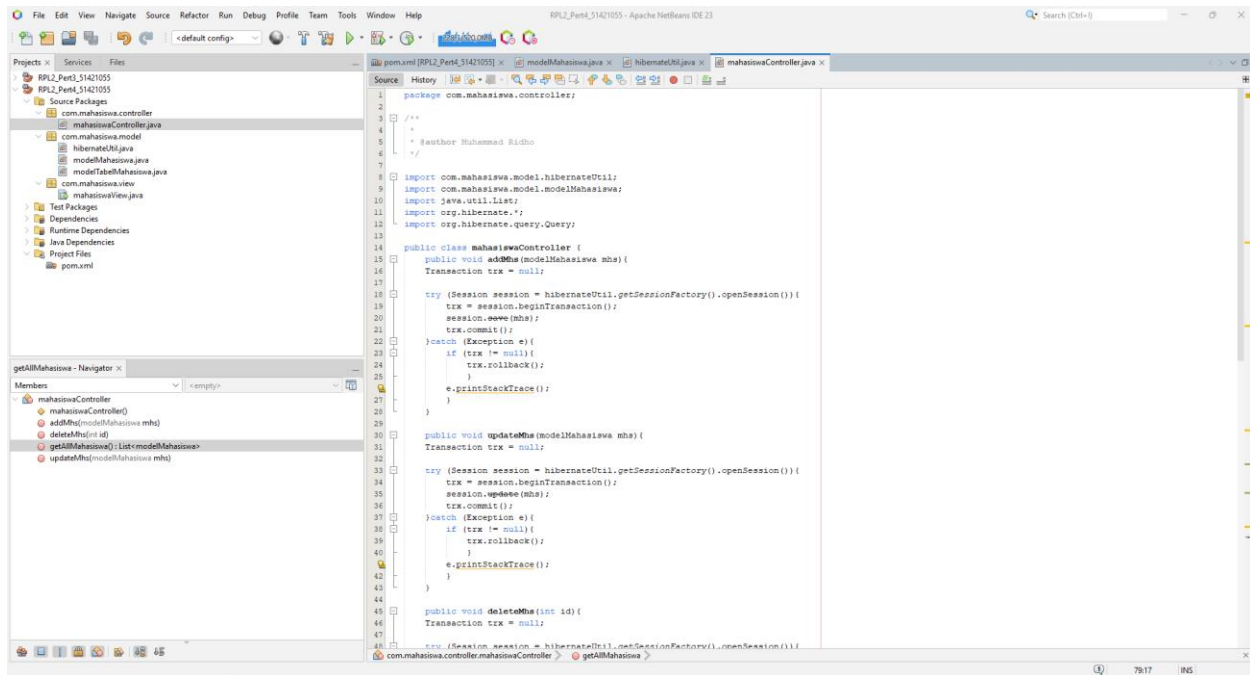
```
package com.mahasiswa.model;
```

```

/**
 *
 * @author Muhammad Ridho
 */
import org.hibernate.*;
import org.hibernate.cfg.Configuration;
public class hibernateUtil {
    private static SessionFactory sessionFactory;
    static{
        try{
            sessionFactory = new Configuration().configure().buildSessionFactory();
        } catch (Throwable ex){
            System.err.println("initial SessionFactory Creation failed" + ex);
            throw new ExceptionInInitializerError(ex);
        }
    }
    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }
    public static void testConnection() {
        try (Session session = sessionFactory.openSession()){
            System.out.println("Connection to the database was succesfull");
        }catch (Exception e) {
            System.out.println("Failed to connect to the database");
            e.printStackTrace();
        }
    }
}

```

mahasiswaController.java



Sourcecode:

```
package com.mahasiswa.controller;
```

```
/**
```

```
*
```

```
* @author Muhammad Ridho
```

```
*/
```

```
import com.mahasiswa.model.hibernateUtil;
```

```
import com.mahasiswa.model.modelMahasiswa;
```

```
import java.util.List;
```

```
import org.hibernate.*;
```

```
import org.hibernate.query.Query;
```

```
public class mahasiswaController {
```

```
    public void addMhs(modelMahasiswa mhs){
```

```
        Transaction trx = null;
```

```
        try (Session session = hibernateUtil.getSessionFactory().openSession()){
```

```

        trx = session.beginTransaction();

        session.save(mhs);

        trx.commit();
    }catch (Exception e){
        if (trx != null){
            trx.rollback();
        }
        e.printStackTrace();
    }
}

public void updateMhs(modelMahasiswa mhs){
    Transaction trx = null;

    try (Session session = hibernateUtil.getSessionFactory().openSession()){
        trx = session.beginTransaction();

        session.update(mhs);

        trx.commit();
    }catch (Exception e){
        if (trx != null){
            trx.rollback();
        }
        e.printStackTrace();
    }
}

public void deleteMhs(int id){
    Transaction trx = null;

    try (Session session = hibernateUtil.getSessionFactory().openSession()){
        trx = session.beginTransaction();

        modelMahasiswa mhs = session.get(modelMahasiswa.class, id);

        if (mhs != null){

```



```

        session.delete(mhs);

        System.out.println("Berhasil dihapus");
    }

    trx.commit();
}catch (Exception e){
    if (trx != null){
        trx.rollback();
    }
    e.printStackTrace();
}
}

public List<modelMahasiswa> getAllMahasiswa(){
    Transaction trx = null;

    List<modelMahasiswa> listMhs = null;

    try (Session session = hibernateUtil.getSessionFactory().openSession()){
        trx = session.beginTransaction();

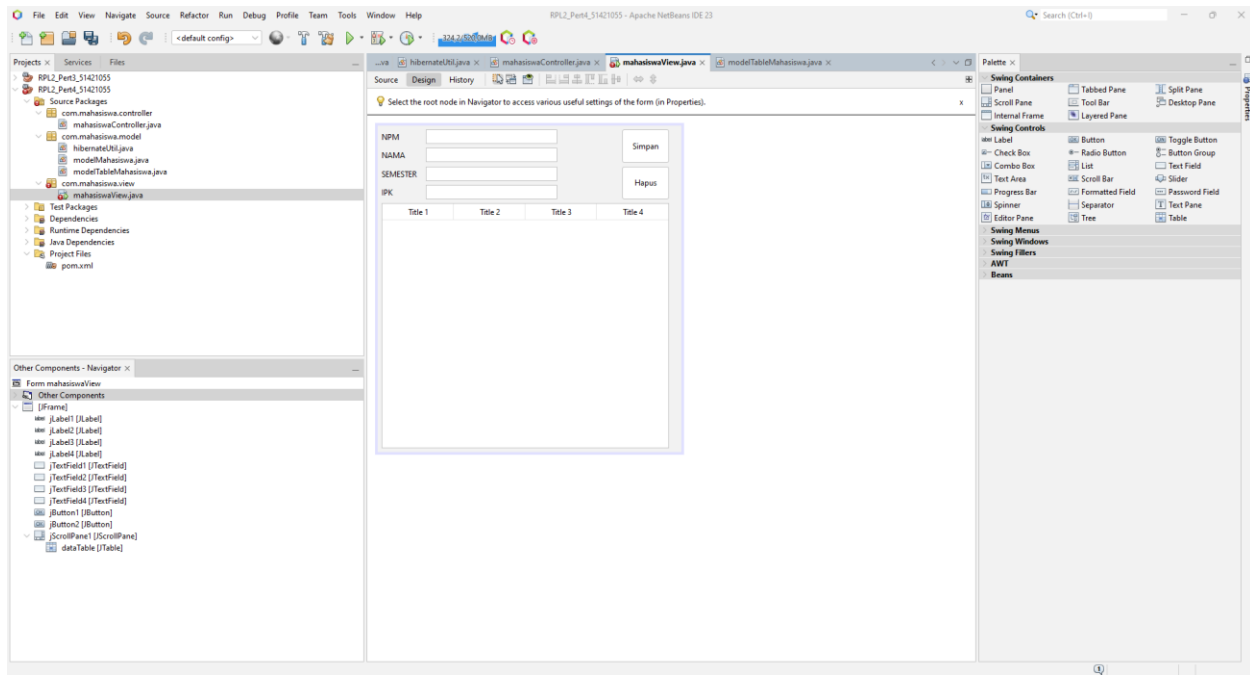
        Query<modelMahasiswa> query = session.createQuery("from modelMahasiswa",
modelMahasiswa.class);

        listMhs = query.list();

        trx.commit();
    }catch (Exception e){
        if (trx != null){
            trx.rollback();
        }
        e.printStackTrace();
    }
    return listMhs;
}
}

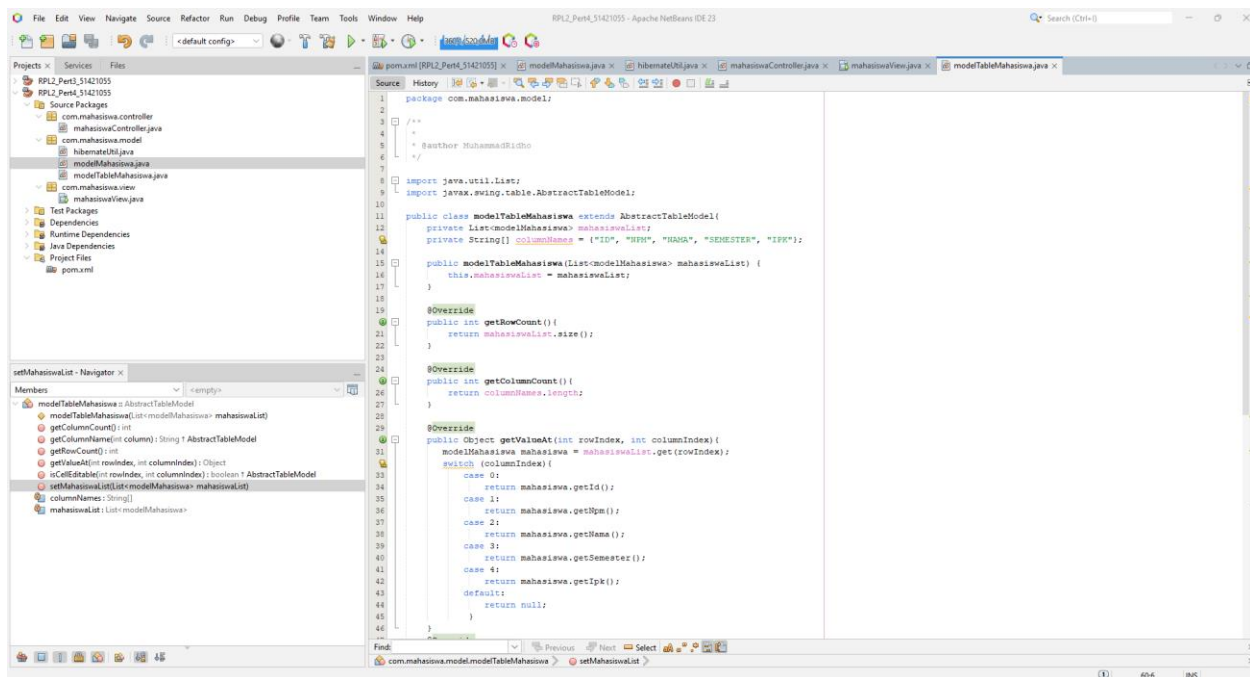
```

mahasiswaView.java



Pembuatan form pada mahasiswa.view.java menggunakan jframe.

modelTableMahasiswa.java



Sourcecode:

```
package com.mahasiswa.model;
```

```
/**
```

```
*
```

```
* @author MuhammadRidho
```

```
*/
```

```
import java.util.List;
```

```
import javax.swing.table.AbstractTableModel;
```

```
public class modelTableMahasiswa extends AbstractTableModel{
```

```
    private List<modelMahasiswa> mahasiswaList;
```

```
    private String[] columnNames = {"ID", "NPM", "NAMA", "SEMESTER", "IPK"};
```

```
    public modelTableMahasiswa(List<modelMahasiswa> mahasiswaList) {
```

```
        this.mahasiswaList = mahasiswaList;
```

```
    }
```

```
    @Override
```

```
    public int getRowCount(){
```

```
        return mahasiswaList.size();
```

```
    }
```

```
    @Override
```

```
    public int getColumnCount(){
```

```
        return columnNames.length;
```

```
    }
```

```
    @Override
```

```
    public Object getValueAt(int rowIndex, int columnIndex){
```

```
        modelMahasiswa mahasiswa = mahasiswaList.get(rowIndex);
```

```
        switch (columnIndex){
```

```
            case 0:
```

```
                return mahasiswa.getId();
```

case 1:

return mahasiswa.getNpm();

case 2:

return mahasiswa.getNama();

case 3:

return mahasiswa.getSemester();

case 4:

return mahasiswa.getIdpk();

default:

return null;

}

}

@Override

public String getColumnName(int column){

return columnNames[column];

}

@Override

public boolean isCellEditable(int rowIndex, int columnIndex){

return false;

}

public void setMahasiswaList(List<modelMahasiswa> mahasiswaList){

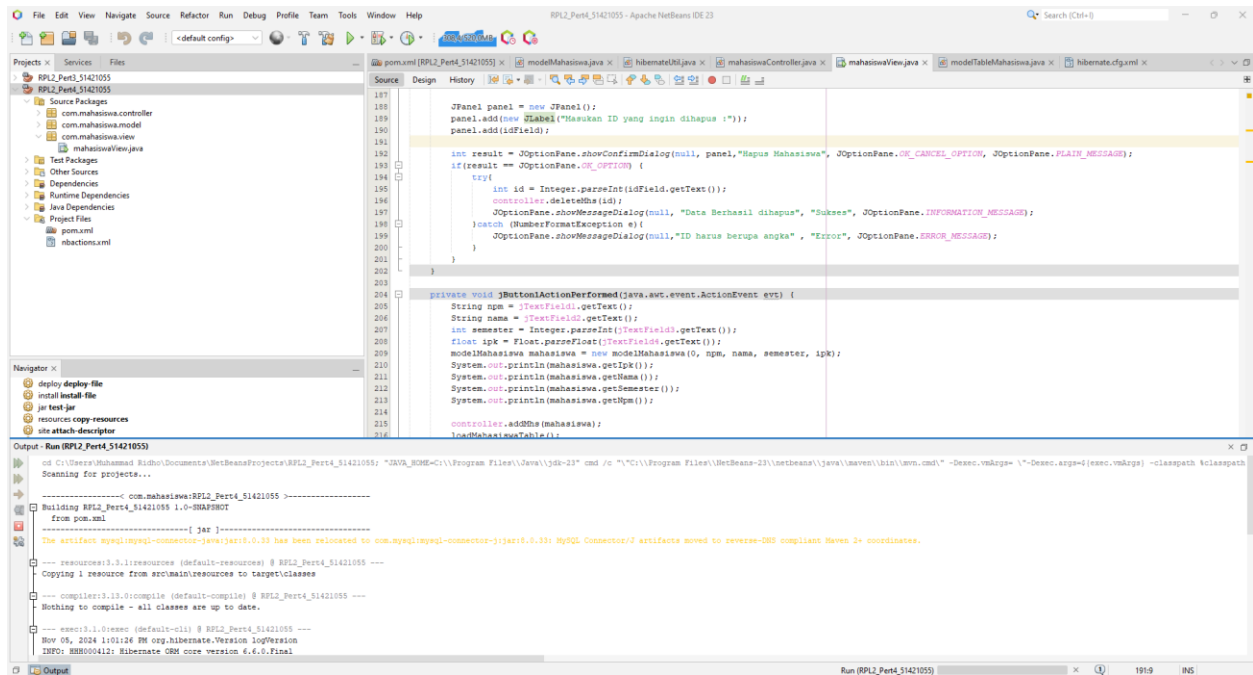
this.mahasiswaList = mahasiswaList;

fireTableDataChanged();

}

}

mahasiswaView.java



Pembuatan source code mahasiswaView untuk mengaktifkan fungsi button dan formnya..

package com.mahasiswa.view;

/**

*

* @author Muhammad Ridho

*/

import com.mahasiswa.model.modelTableMahasiswa;

import com.mahasiswa.model.modelMahasiswa;

import com.mahasiswa.controller.mahasiswaController;

import com.mahasiswa.model.hibernateUtil;

import java.util.List;

import javax.swing.*;

public class mahasiswaView extends JFrame {

private mahasiswaController controller;

```
public void loadMahasiswaTable(){  
    List<modelMahasiswa> listMahasiswa = controller.getAllMahasiswa();  
    modelTableMahasiswa tableModel = new modelTableMahasiswa(listMahasiswa);  
  
    dataTable.setModel(tableModel);  
}
```

```
/**  
 * Creates new form mahasiswaView  
 */  
public mahasiswaView() {  
    initComponents();  
    controller = new mahasiswaController();  
    hibernateUtil.testConnection();  
    loadMahasiswaTable();  
}
```

```
public void clearTextField(){  
    jTextField1.setText("");  
    jTextField2.setText("");  
    jTextField4.setText("");  
    jTextField4.setText("");  
  
}
```

```
/**  
 * 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")

JTextField idField = new JTextField(5);


JPanel panel = new JPanel();

panel.add(new JLabel("Masukan ID yang ingin dihapus :"));

panel.add(idField);


int result = JOptionPane.showConfirmDialog(null, panel, "Hapus Mahasiswa",
JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE);

if(result == JOptionPane.OK_OPTION) {
    try{
        int id = Integer.parseInt(idField.getText());

        controller.deleteMhs(id);

        JOptionPane.showMessageDialog(null, "Data Berhasil dihapus", "Sukses",
JOptionPane.INFORMATION_MESSAGE);

    }catch (NumberFormatException e){

        JOptionPane.showMessageDialog(null, "ID harus berupa angka" , "Error",
JOptionPane.ERROR_MESSAGE);

    }
}

String npm = jTextField1.getText();

String nama = jTextField2.getText();

int semester = Integer.parseInt(jTextField3.getText());

float ipk = Float.parseFloat(jTextField4.getText());

modelMahasiswa mahasiswa = new modelMahasiswa(0, npm, nama, semester, ipk);

System.out.println(mahasiswa.getIpk());

System.out.println(mahasiswa.getNama());

System.out.println(mahasiswa.getSemester());

System.out.println(mahasiswa.getNpm());

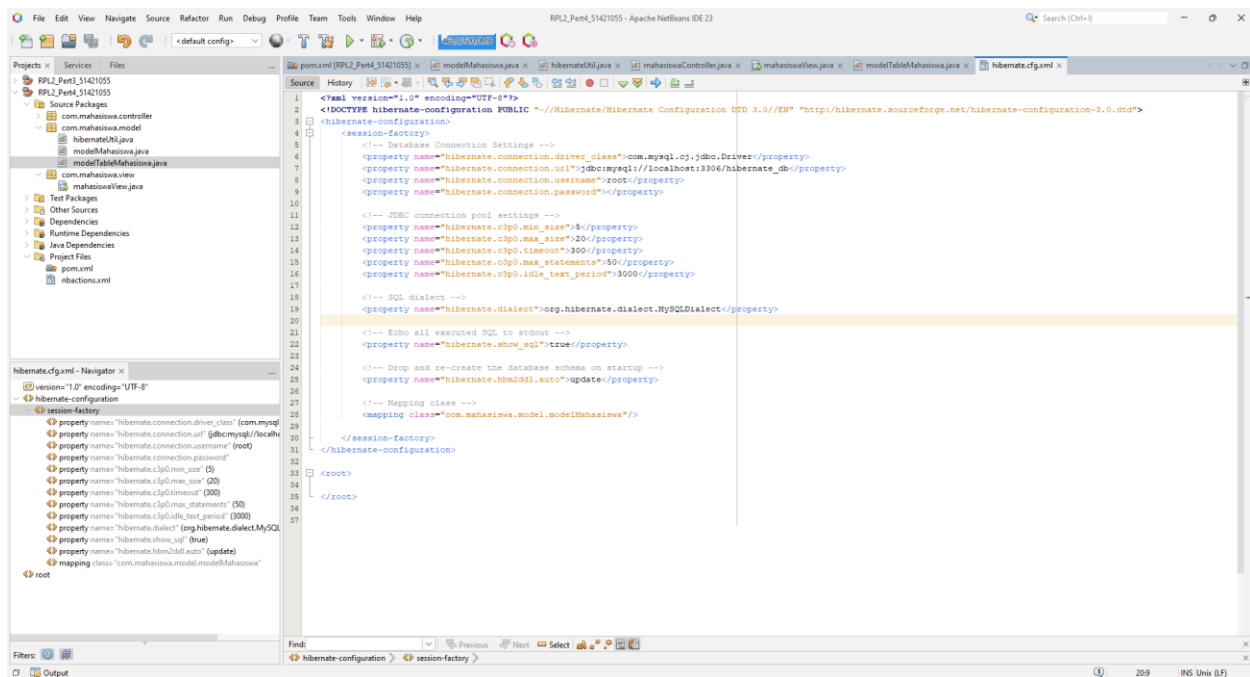
```

```
controller.addMhs(mahasiswa);
```

```
loadMahasiswaTable();
```

```
clearTextField();
```

Hibernate.cfg.xml



Sourcecode:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
```

```
<hibernate-configuration>
```

```
<session-factory>
```

```
<!-- Database Connection Settings -->
```

```
<property name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property>
```

```
<property name="hibernate.connection.url">jdbc:mysql://localhost:3306/hibernate_db</property>
```

```
<property name="hibernate.connection.username">root</property>
```



```
<property name="hibernate.connection.password"></property>
<!-- JDBC connection pool settings -->
<property name="hibernate.c3p0.min_size">5</property>
<property name="hibernate.c3p0.max_size">20</property>
<property name="hibernate.c3p0.timeout">300</property>
<property name="hibernate.c3p0.max_statements">50</property>
<property name="hibernate.c3p0.idle_text_period">3000</property>
<!-- SQL dialect -->
<property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
<!-- Echo all executed SQL to stdout -->
<property name="hibernate.show_sql">true</property>
<!-- Drop and re-create the database schema on startup -->
<property name="hibernate.hbm2ddl.auto">update</property>
<!-- Mapping class -->
<mapping class="com.mahasiswa.model.modelMahasiswa"/>
</session-factory>
</hibernate-configuration>

<root>

</root>
```

Output Source Code

Projects × Services Files

> RPL2_Pert3_51421055

RPL2_Pert4_51421055

Source Packages

com.mahasiswa.controller

com.mahasiswa.model

com.mahasiswa.view

mahasiswaView.java

Test Packages

Other Sources

Dependencies

Runtime Dependencies

Java Dependencies

Project Files

pom.xml

nbactions.xml

pom.xml [RPL2_Pert4_51421055] × modelMahasiswa.java × hiberna

NPM

NAMA

SEMESTER 7

IPK

Simpan

Hapus

ID	NPM	NAMA	SEME...	IPK
1	51421055	Muhammad Ridho	7	3.77

Navigator ×

deploy **deploy-file**

install **install-file**

jar **test-jar**

resources **copy-resources**