ACTIVITY PERTEMUAN 4

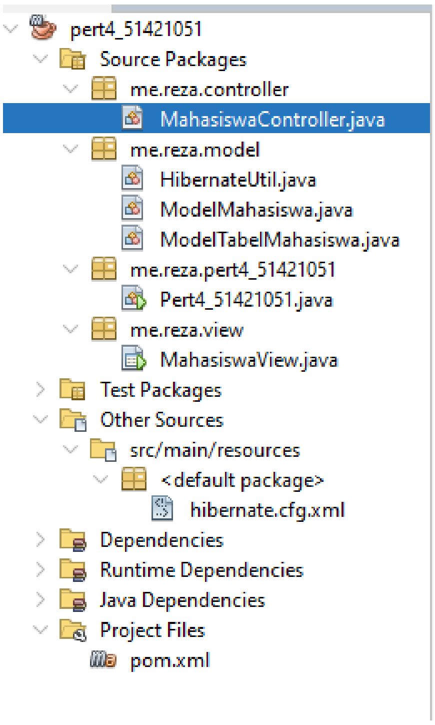
NAMA : Muhammmad Reza Rahman

NPM : 51421051

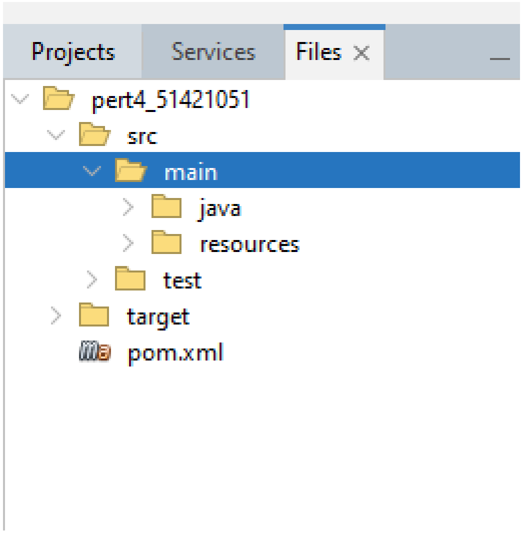
Kelas : 4IA14

Materi : Konsep Dasar ORM dan Frame Hibernate

Mata Praktikum : Rekayasa Perangkat Lunak 2



pert4_51421051
Source Packages
me.reza.controller
MahasiswaController.java
me.reza.model
HibernateUtil.java
ModelMahasiswa.java
ModelTabelMahasiswa.java
me.reza.pert4_51421051
Pert4_51421051.java
me.reza.view
MahasiswaView.java
Test Packages
Other Sources
src/main/resources
<default package>
hibernate.cfg.xml
Dependencies
Runtime Dependencies
Java Dependencies
Project Files
pom.xml



pert4_51421051
src
main
java
resources
test
target
pom.xml

Input Kode:

Pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>me.reza</groupId>
    <artifactId>pert4_51421051</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>jar</packaging>
```

```xml
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>21</maven.compiler.source>
        <maven.compiler.target>21</maven.compiler.target>
        <exec.mainClass>me.reza.pert4_51421051.Pert4_51421051</exec.mainClass>
    </properties>

    <dependencies>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.33</version>
        </dependency>
        <dependency>
            <groupId>org.hibernate.orm</groupId>
            <artifactId>hibernate-core</artifactId>
            <version>6.6.0.Final</version>
        </dependency>
    </dependencies>
    <build>
    <resources>
        <resource>
            <directory>src/main/resources</directory>
            <filtering>false</filtering>
        </resource>
    </resources>
</build>
</project>
```

ModelMahasiswa.java

```java
package me.reza.model;

import jakarta.persistence.*;
/**
 *
 * @author WINDOWS 10
 */
@Entity
@Table(name = "mahasiswa")
public class ModelMahasiswa {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private int id;

    @Column(name = "npm", nullable = false, length = 10)
    private String npm;
    @Column(name = "nama", nullable = false, length = 55)
    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;
    }

}
```

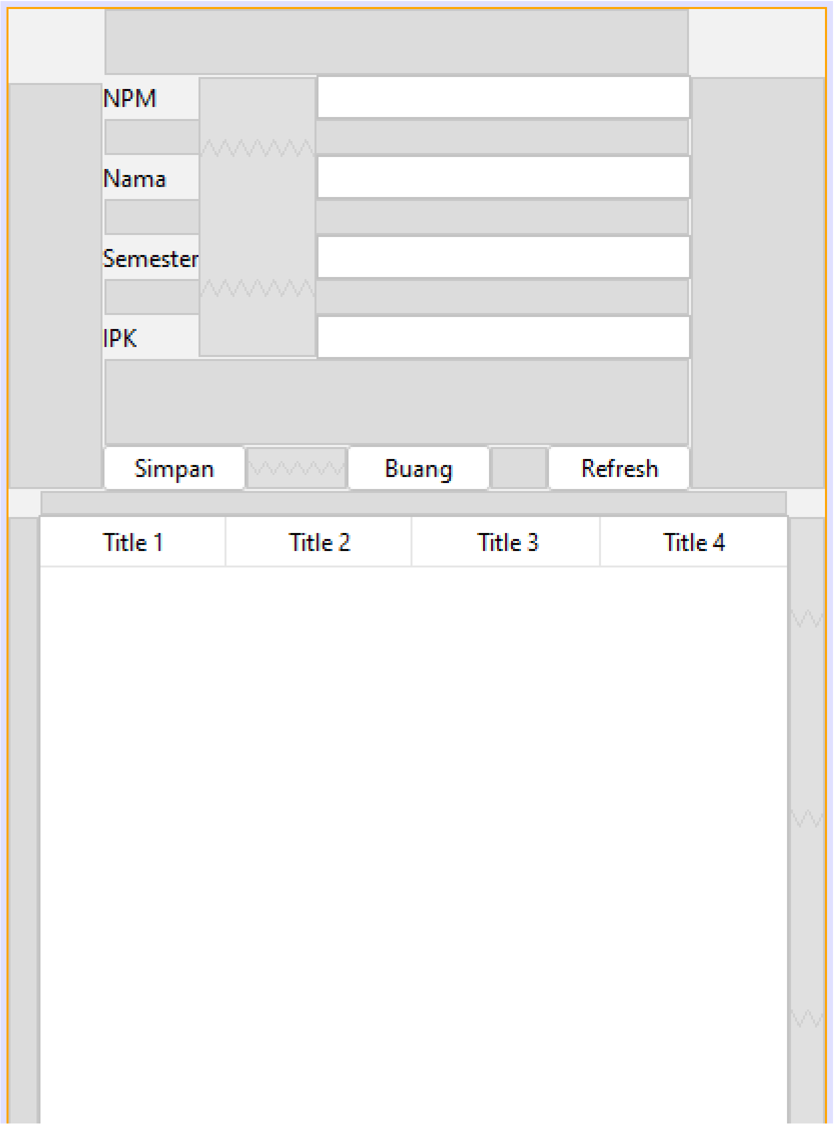
HibernateUtil.java

```
package me.reza.model;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
/**
 *
 * @author WINDOWS 10
 */
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 successful!");
        } catch (Exception e) {
            System.err.println("Failed to connect to the database.");
            e.printStackTrace();
        }
    }
}
```

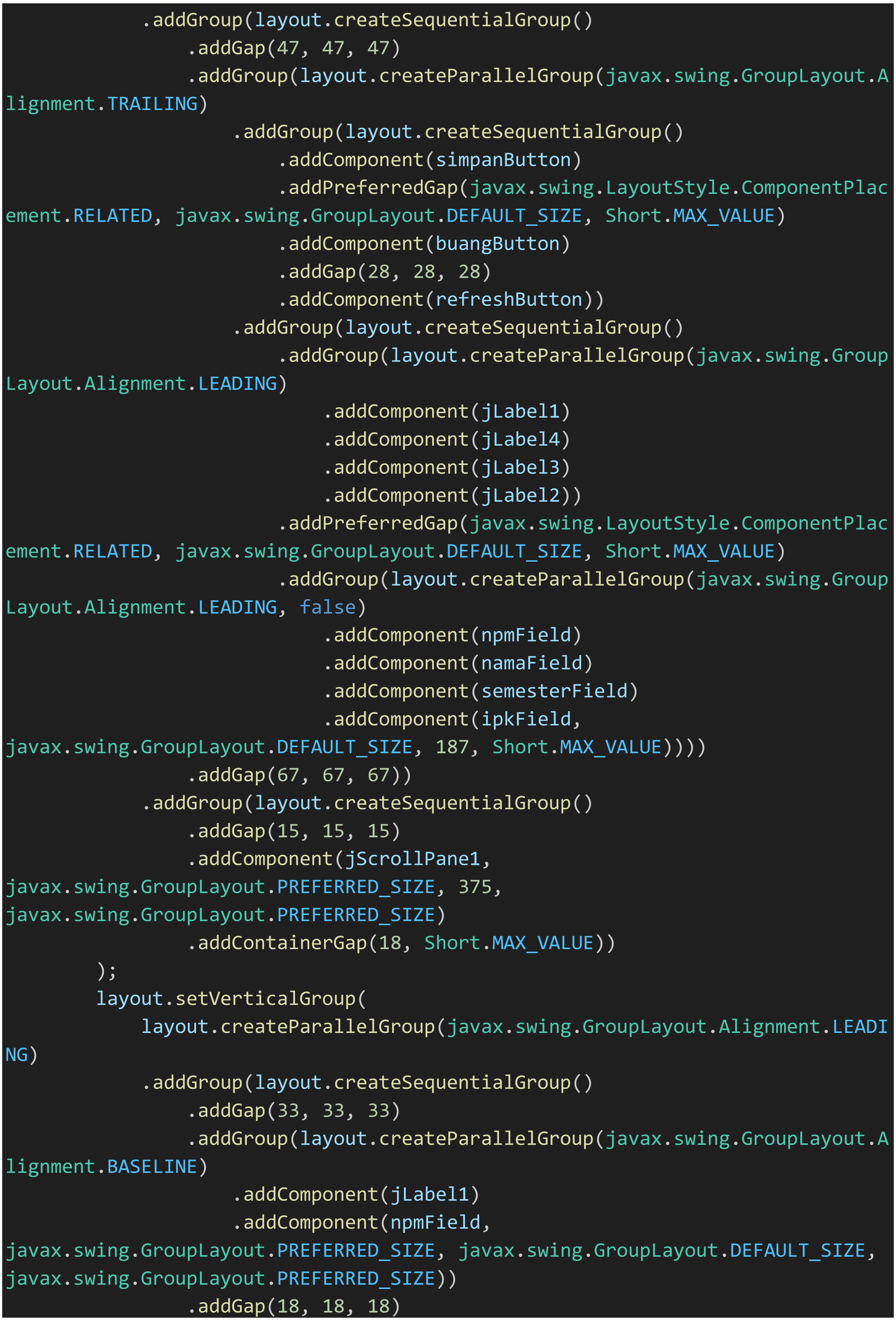
MahasiswaController.java

```
package me.reza.controller;

import me.reza.model.HibernateUtil;
import me.reza.model.ModelMahasiswa;
import java.util.List;
import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.query.Query;

/**
 *
 * @author WINDOWS 10
 */
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 Hapus");
            }
            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;
    }
}
```

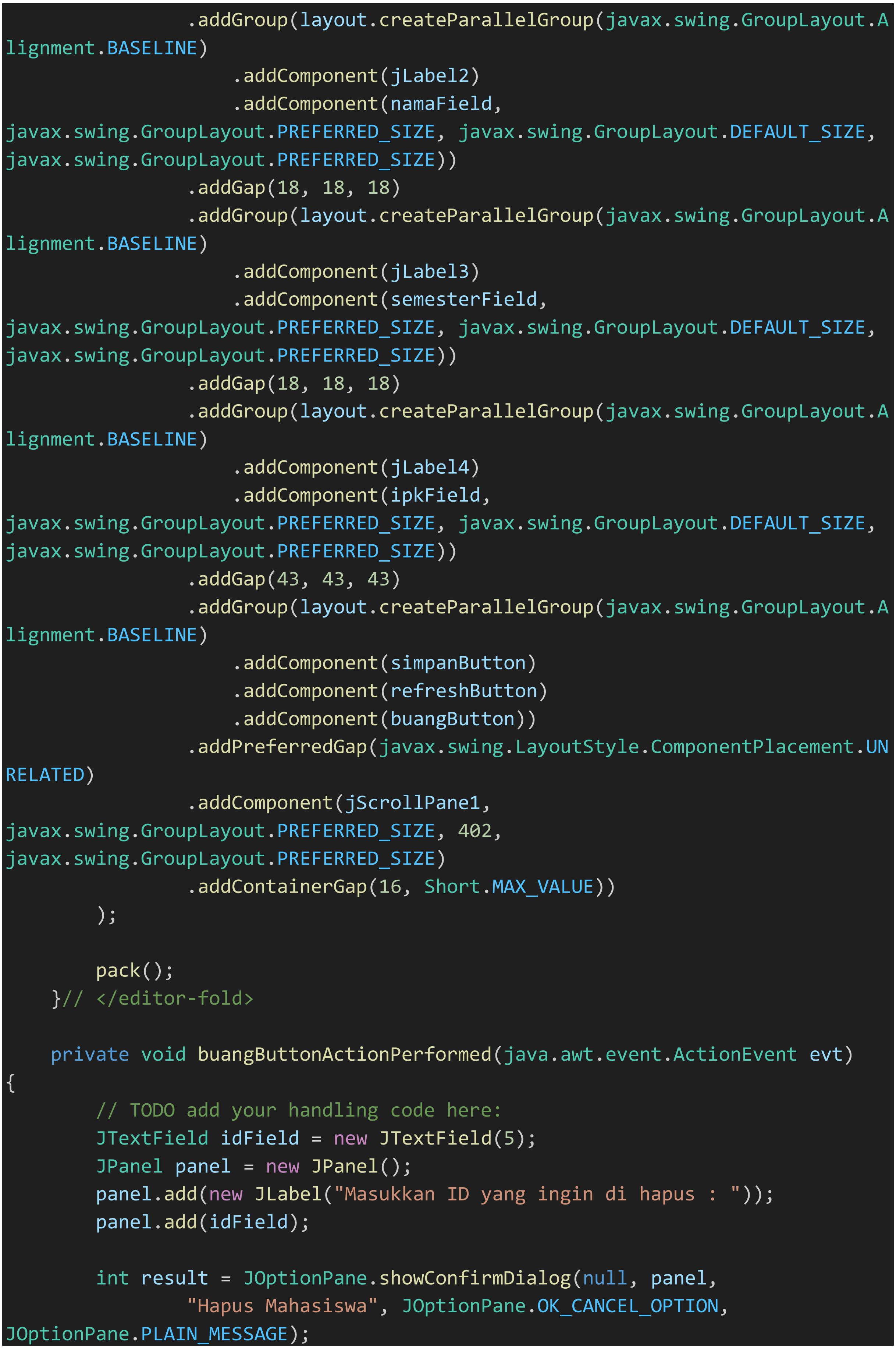
MahasiswaView.java



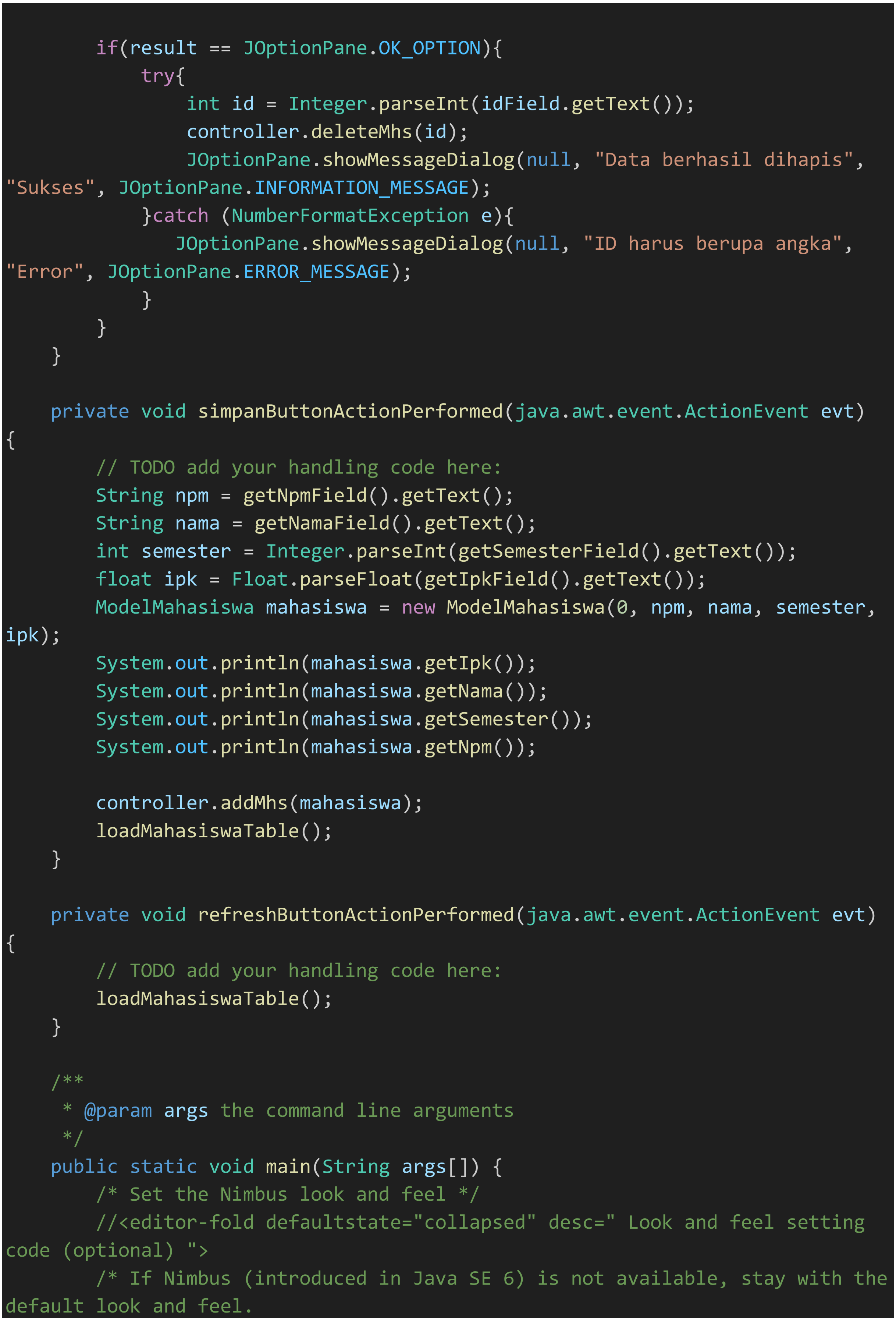
```
package me.reza.view;

/**
 *
 * @author WINDOWS 10
 */

import me.reza.controller.MahasiswaController;
import me.reza.model.HibernateUtil;
import me.reza.model.ModelMahasiswa;
import me.reza.model.ModelTabelMahasiswa;
import java.util.List;
import javax.swing.*;

public class MahasiswaView extends javax.swing.JFrame {
    /**
     * Creates new form MahasiswaView
     */
    private MahasiswaController controller;

    public MahasiswaView() {
        initComponents();
        HibernateUtil.testConnection();
        this.controller = new MahasiswaController();
        loadMahasiswaTable();
    }

    public void loadMahasiswaTable() {
        List<ModelMahasiswa> listMahasiswa = controller.getAllMahasiswa();

        ModelTabelMahasiswa tableModel = new
ModelTabelMahasiswa(listMahasiswa);
        dataTable.setModel(tableModel);
    }

    /**
     * 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();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        npmField = new javax.swing.JTextField();
        namaField = new javax.swing.JTextField();
        semesterField = new javax.swing.JTextField();
        ipkField = new javax.swing.JTextField();
        simpanButton = new javax.swing.JButton();
        buangButton = new javax.swing.JButton();
        refreshButton = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        dataTable = new javax.swing.JTable();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        jLabel1.setText("NPM");

        jLabel2.setText("Nama");

        jLabel3.setText("Semester");

        jLabel4.setText("IPK");

        simpanButton.setText("Simpan");
        simpanButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                simpanButtonActionPerformed(evt);
            }
        });

        buangButton.setText("Buang");
        buangButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                buangButtonActionPerformed(evt);
            }
        });

        refreshButton.setText("Refresh");
        refreshButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                refreshButtonActionPerformed(evt);
            }
        });

        dataTable.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null}
            },
            new String [] {
                "Title 1", "Title 2", "Title 3", "Title 4"
            }
        ));
        jScrollPane1.setViewportView(dataTable);

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
```



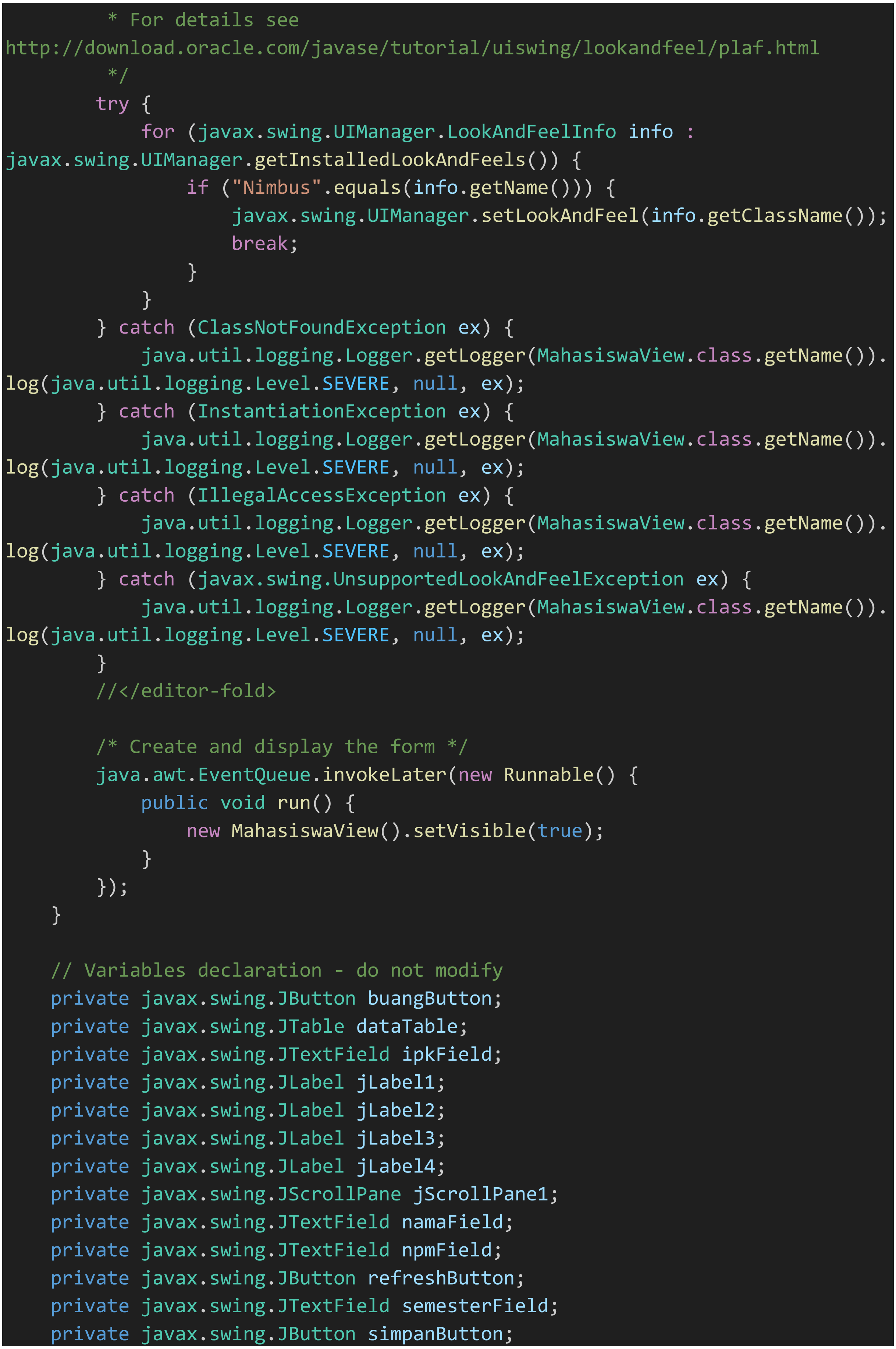
```
            .addGroup(layout.createSequentialGroup()
                .addGap(47, 47, 47)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.A
lignment.TRAILING)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(simpanButton)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlac
ement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(buangButton)
                        .addGap(28, 28, 28)
                        .addComponent(refreshButton))
                    .addGroup(layout.createSequentialGroup()
                        .addGroup(layout.createParallelGroup(javax.swing.Group
Layout.Alignment.LEADING)
                            .addComponent(jLabel1)
                            .addComponent(jLabel4)
                            .addComponent(jLabel3)
                            .addComponent(jLabel2))
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlac
ement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addGroup(layout.createParallelGroup(javax.swing.Group
Layout.Alignment.LEADING, false)
                            .addComponent(npmField)
                            .addComponent(namaField)
                            .addComponent(semesterField)
                            .addComponent(ipkField,
javax.swing.GroupLayout.DEFAULT_SIZE, 187, Short.MAX_VALUE))))
                .addGap(67, 67, 67))
            .addGroup(layout.createSequentialGroup()
                .addGap(15, 15, 15)
                .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 375,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(18, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
            .addGroup(layout.createSequentialGroup()
                .addGap(33, 33, 33)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.A
lignment.BASELINE)
                    .addComponent(jLabel1)
                    .addComponent(npmField,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)
```



```
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.A
lignment.BASELINE)
          .addComponent(jLabel2)
          .addComponent(namaField,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.A
lignment.BASELINE)
          .addComponent(jLabel3)
          .addComponent(semesterField,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.A
lignment.BASELINE)
          .addComponent(jLabel4)
          .addComponent(ipkField,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(43, 43, 43)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.A
lignment.BASELINE)
          .addComponent(simpanButton)
          .addComponent(refreshButton)
          .addComponent(buangButton))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UN
RELATED)
        .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 402,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(16, Short.MAX_VALUE))
    );

    pack();
  }// </editor-fold>

  private void buangButtonActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    JTextField idField = new JTextField(5);
    JPanel panel = new JPanel();
    panel.add(new JLabel("Masukkan ID yang ingin di hapus : "));
    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 dihapis", 
"Sukses", JOptionPane.INFORMATION_MESSAGE);
            }catch (NumberFormatException e){
               JOptionPane.showMessageDialog(null, "ID harus berupa angka", 
"Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    }

    private void simpanButtonActionPerformed(java.awt.event.ActionEvent evt) 
{
        // TODO add your handling code here:
        String npm = getNpmField().getText();
        String nama = getNamaField().getText();
        int semester = Integer.parseInt(getSemesterField().getText());
        float ipk = Float.parseFloat(getIpkField().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();
    }

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

    /**
     * @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.
```



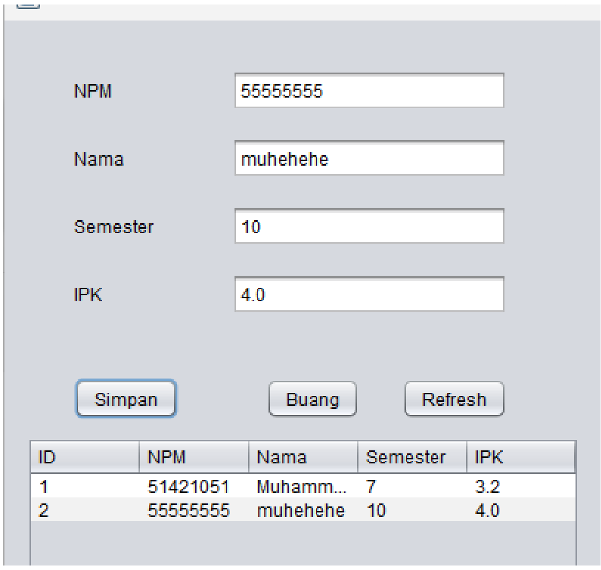
```
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(MahasiswaView.class.getName()).
log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(MahasiswaView.class.getName()).
log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(MahasiswaView.class.getName()).
log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(MahasiswaView.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 MahasiswaView().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton buangButton;
    private javax.swing.JTable dataTable;
    private javax.swing.JTextField ipkField;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTextField namaField;
    private javax.swing.JTextField npmField;
    private javax.swing.JButton refreshButton;
    private javax.swing.JTextField semesterField;
    private javax.swing.JButton simpanButton;
    // End of variables declaration

    public MahasiswaController getController() {
        return controller;
    }

    public void setController(MahasiswaController controller) {
        this.controller = controller;
    }

    public JButton getBuangButton() {
        return buangButton;
    }

    public void setBuangButton(JButton buangButton) {
        this.buangButton = buangButton;
    }

    public JTable getDataTable() {
        return dataTable;
    }

    public void setDataTable(JTable dataTable) {
        this.dataTable = dataTable;
    }

    public JTextField getIpkField() {
        return ipkField;
    }

    public void setIpkField(JTextField ipkField) {
        this.ipkField = ipkField;
    }

    public JLabel getjLabel1() {
        return jLabel1;
    }

    public void setjLabel1(JLabel jLabel1) {
        this.jLabel1 = jLabel1;
    }

    public JLabel getjLabel2() {
        return jLabel2;
    }

    public void setjLabel2(JLabel jLabel2) {
        this.jLabel2 = jLabel2;
    }

    public JLabel getjLabel3() {
        return jLabel3;
    }

    public void setjLabel3(JLabel jLabel3) {
        this.jLabel3 = jLabel3;
    }

    public JLabel getjLabel4() {
        return jLabel4;
    }

    public void setjLabel4(JLabel jLabel4) {
        this.jLabel4 = jLabel4;
    }

    public JScrollPane getjScrollPane1() {
        return jScrollPane1;
    }

    public void setjScrollPane1(JScrollPane jScrollPane1) {
        this.jScrollPane1 = jScrollPane1;
    }

    public JTextField getNamaField() {
        return namaField;
    }

    public void setNamaField(JTextField namaField) {
        this.namaField = namaField;
    }

    public JTextField getNpmField() {
        return npmField;
    }

    public void setNpmField(JTextField npmField) {
        this.npmField = npmField;
    }

    public JButton getRefreshButton() {
        return refreshButton;
    }

    public void setRefreshButton(JButton refreshButton) {
        this.refreshButton = refreshButton;
    }

    public JTextField getSemesterField() {
        return semesterField;
    }

    public void setSemesterField(JTextField semesterField) {
        this.semesterField = semesterField;
    }

    public JButton getSimpanButton() {
        return simpanButton;
    }

    public void setSimpanButton(JButton simpanButton) {
        this.simpanButton = simpanButton;
    }

}
```

Hibernate.cfg.xml

```
<?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_51421051
</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_test_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="me.reza.model.ModelMahasiswa"/>
    </session-factory>
</hibernate-configuration>
```

OUTPUT :

Simpan



NPM 55555555

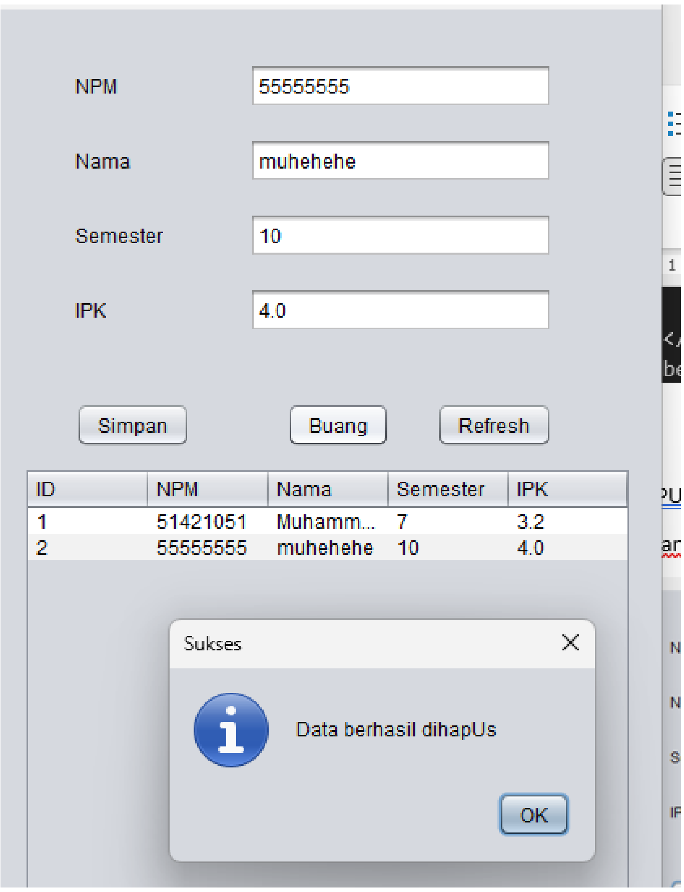
Nama muhehehe

Semester 10

IPK 4.0

ID	NPM	Nama	Semester	IPK
1	51421051	Muhamm...	7	3.2
2	55555555	muhehehe	10	4.0

Buang :



NPM 55555555

Nama muhehehe

Semester 10

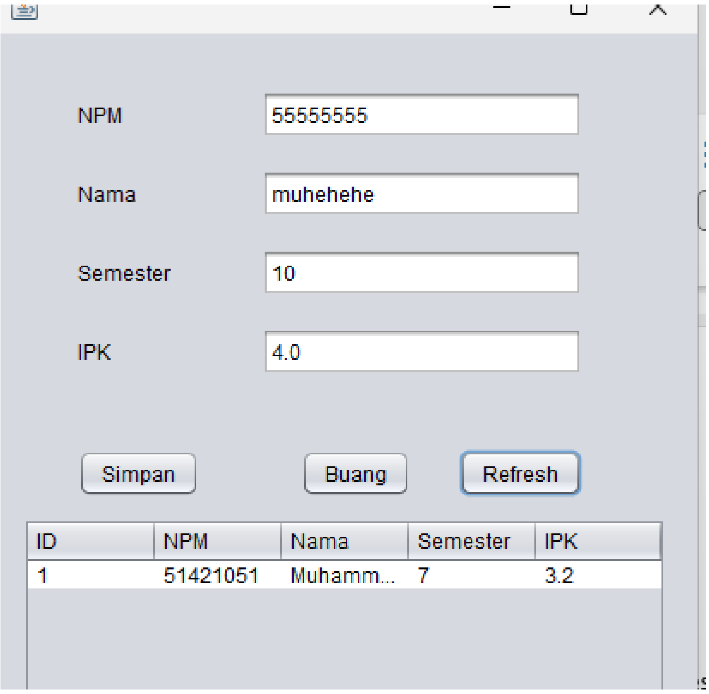
IPK 4.0

ID	NPM	Nama	Semester	IPK
1	51421051	Muhamm...	7	3.2
2	55555555	muhehehe	10	4.0

Sukses

Data berhasil dihapUs

Refresh :



NPM: 55555555

Nama: muhehehe

Semester: 10

IPK: 4.0

ID	NPM	Nama	Semester	IPK
1	51421051	Muhamm...	7	3.2