Nama: Shaun Patrick Hendra

NPM: 51421416

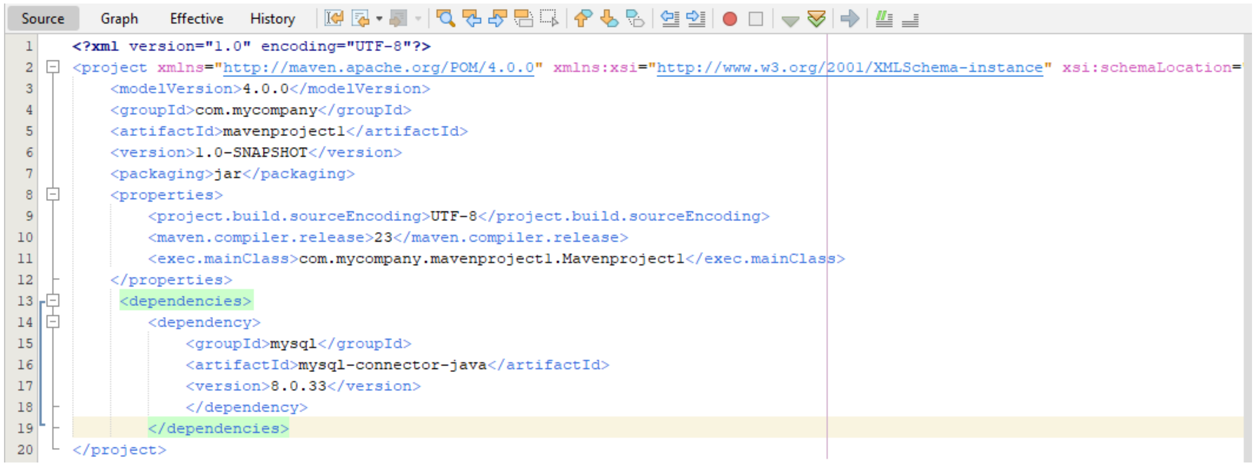
Kelas: 4IA28

Praktikum RPL 2

Pertemuan 3

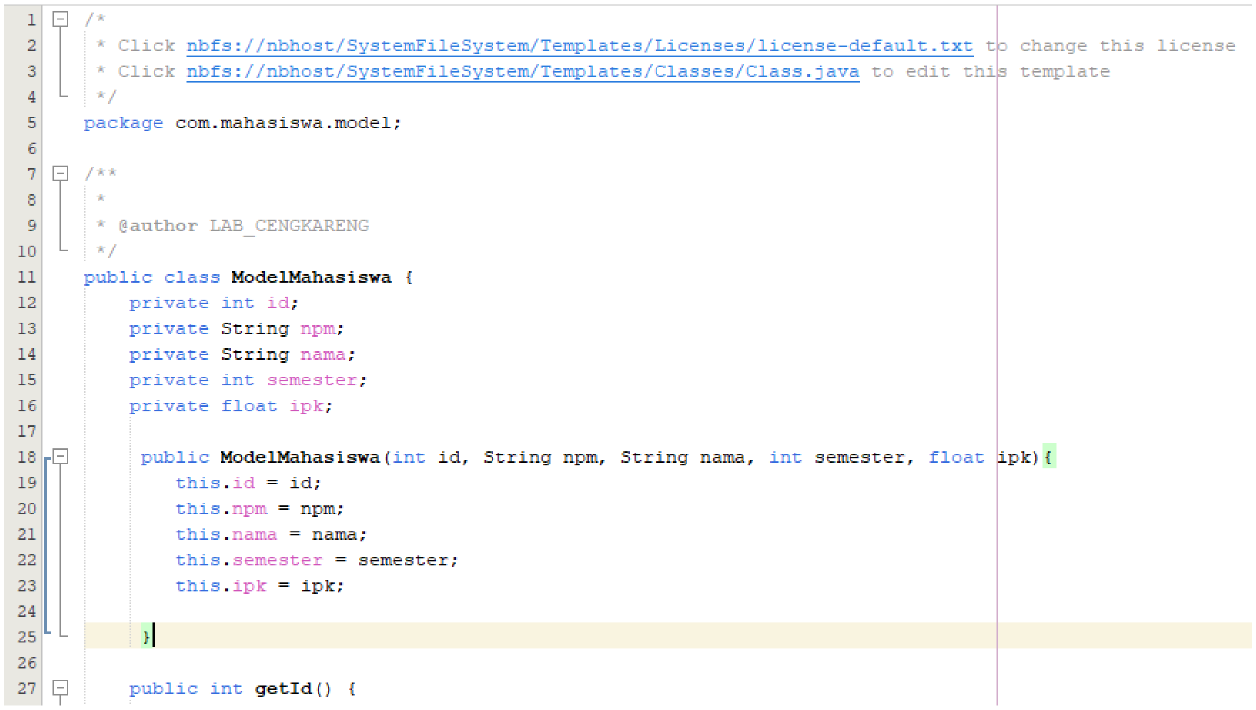
Source Code:

pom.xml

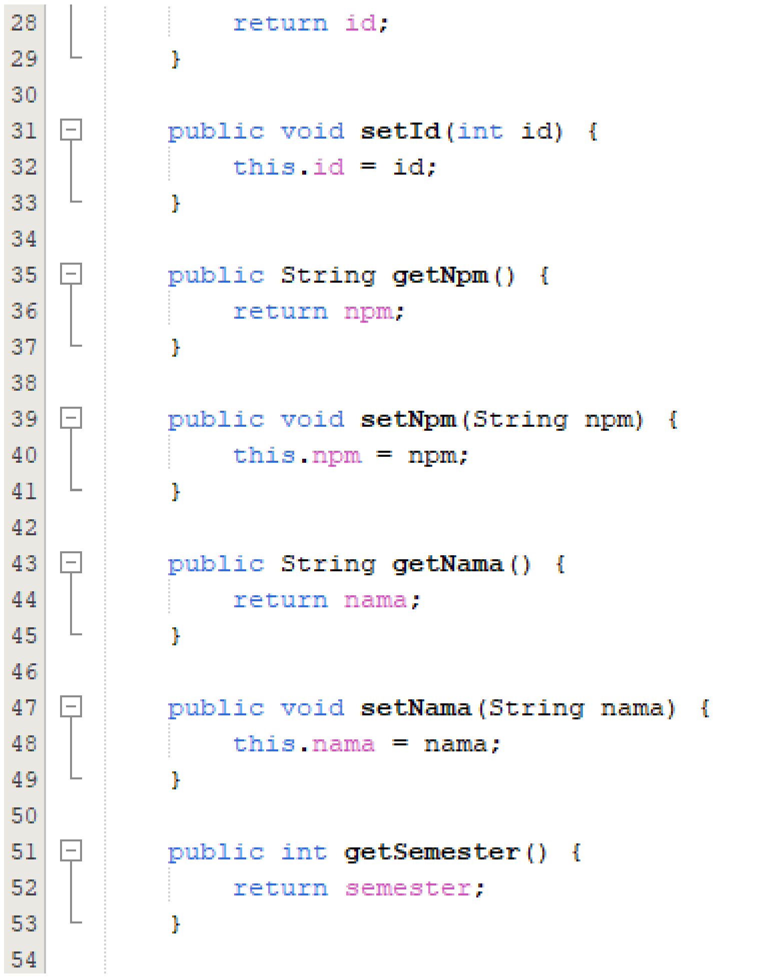


```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.mycompany</groupId>
    <artifactId>mavenproject1</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>jar</packaging>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.release>23</maven.compiler.release>
        <exec.mainClass>com.mycompany.mavenproject1.Mavenproject1</exec.mainClass>
    </properties>
     <dependencies>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.33</version>
            </dependency>
        </dependencies>
</project>
```

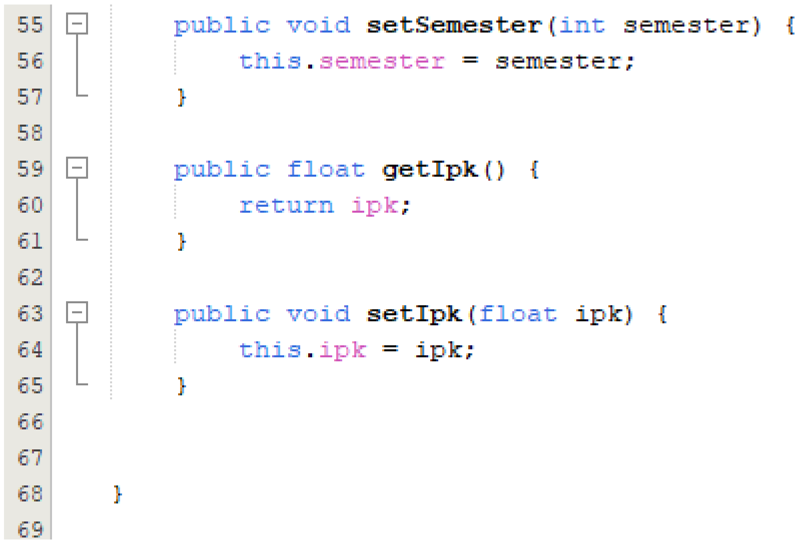
ModelMahasiswa



```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mahasiswa.model;

/**
 *
 * @author LAB_CENGKARENG
 */
public class ModelMahasiswa {
    private int id;
    private String npm;
    private String nama;
    private int semester;
    private float ipk;

     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() {
```

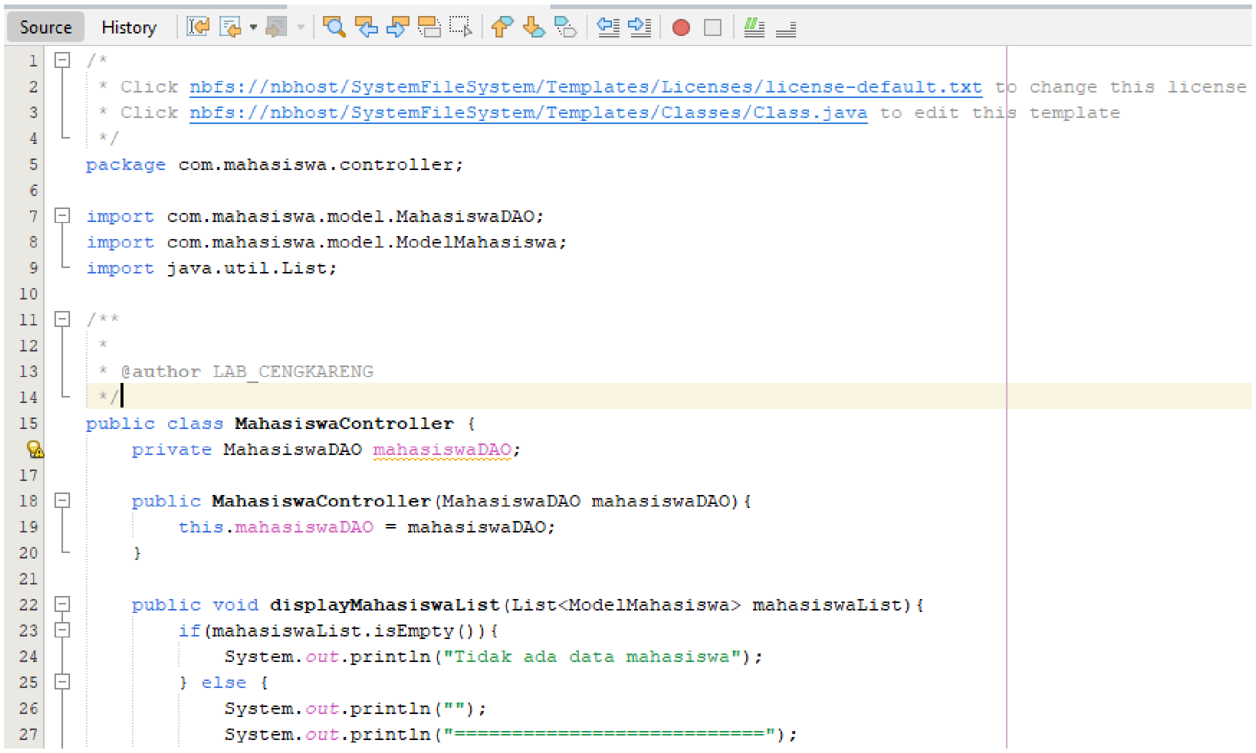


```java
        return id;
    }

    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;
    }

```

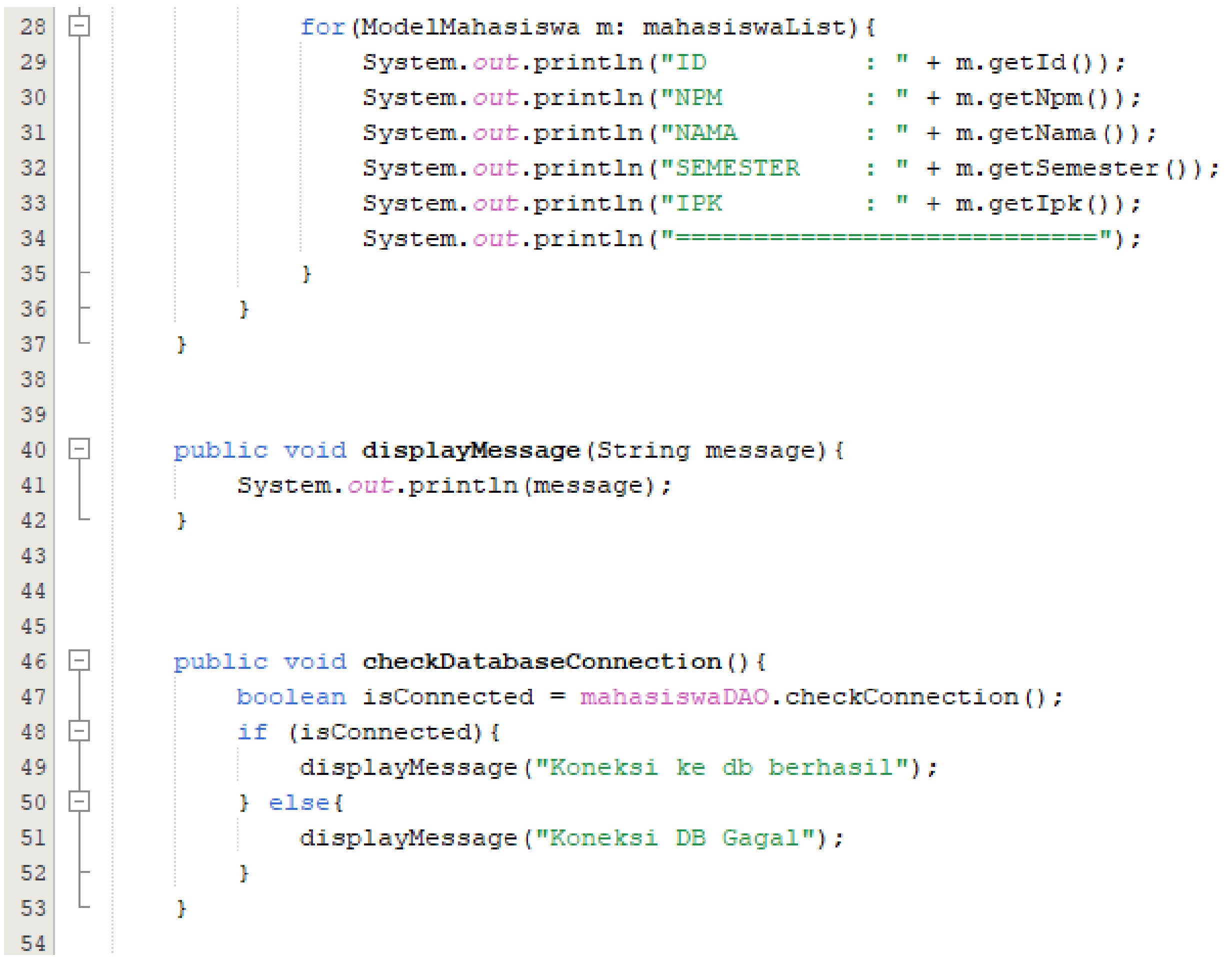


```java
    public void setSemester(int semester) {
        this.semester = semester;
    }

    public float getIpk() {
        return ipk;
    }

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


}

```

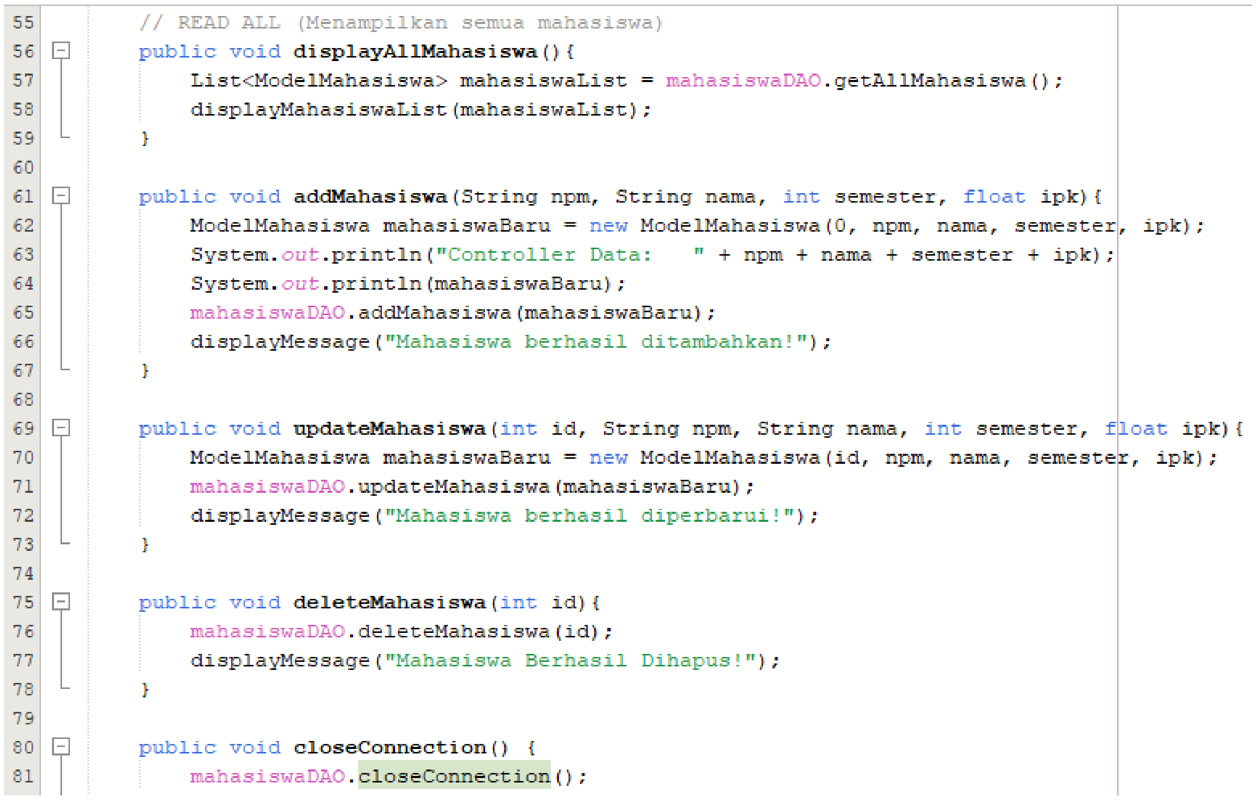
MahasiswaController



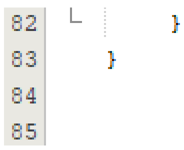
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mahasiswa.controller;

import com.mahasiswa.model.MahasiswaDAO;
import com.mahasiswa.model.ModelMahasiswa;
import java.util.List;

/**
 *
 * @author LAB_CENGKARENG
 */
public class MahasiswaController {
    private MahasiswaDAO mahasiswaDAO;

    public MahasiswaController(MahasiswaDAO mahasiswaDAO){
        this.mahasiswaDAO = mahasiswaDAO;
    }

    public void displayMahasiswaList(List<ModelMahasiswa> mahasiswaList){
        if(mahasiswaList.isEmpty()){
            System.out.println("Tidak ada data mahasiswa");
        } else {
            System.out.println("");
            System.out.println("===========================");
```



```java
            for(ModelMahasiswa m: mahasiswaList){
                System.out.println("ID          : " + m.getId());
                System.out.println("NPM         : " + m.getNpm());
                System.out.println("NAMA        : " + m.getNama());
                System.out.println("SEMESTER    : " + m.getSemester());
                System.out.println("IPK         : " + m.getIpk());
                System.out.println("===========================");
            }
        }
    }


    public void displayMessage(String message){
        System.out.println(message);
    }



    public void checkDatabaseConnection(){
        boolean isConnected = mahasiswaDAO.checkConnection();
        if (isConnected){
            displayMessage("Koneksi ke db berhasil");
        } else{
            displayMessage("Koneksi DB Gagal");
        }
    }

```

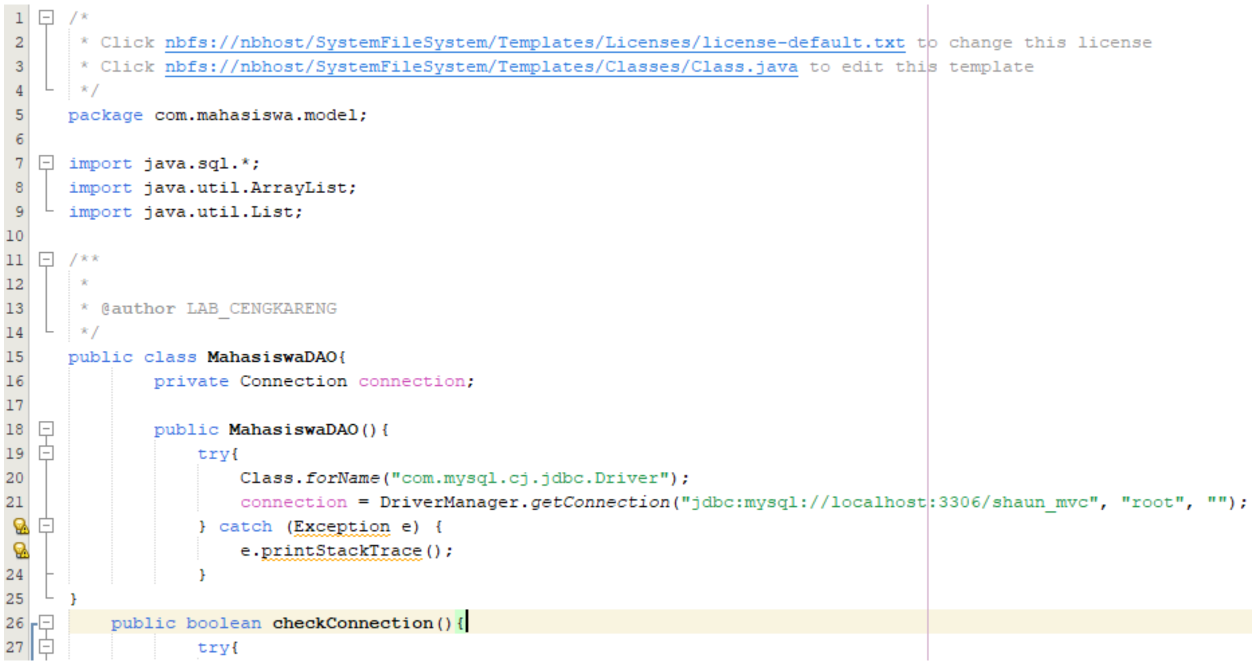


```java
    // READ ALL (Menampilkan semua mahasiswa)
    public void displayAllMahasiswa(){
        List<ModelMahasiswa> mahasiswaList = mahasiswaDAO.getAllMahasiswa();
        displayMahasiswaList(mahasiswaList);
    }

    public void addMahasiswa(String npm, String nama, int semester, float ipk){
        ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(0, npm, nama, semester, ipk);
        System.out.println("Controller Data:    " + npm + nama + semester + ipk);
        System.out.println(mahasiswaBaru);
        mahasiswaDAO.addMahasiswa(mahasiswaBaru);
        displayMessage("Mahasiswa berhasil ditambahkan!");
    }

    public void updateMahasiswa(int id, String npm, String nama, int semester, float ipk){
        ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(id, npm, nama, semester, ipk);
        mahasiswaDAO.updateMahasiswa(mahasiswaBaru);
        displayMessage("Mahasiswa berhasil diperbarui!");
    }

    public void deleteMahasiswa(int id){
        mahasiswaDAO.deleteMahasiswa(id);
        displayMessage("Mahasiswa Berhasil Dihapus!");
    }

    public void closeConnection() {
        mahasiswaDAO.closeConnection();
```

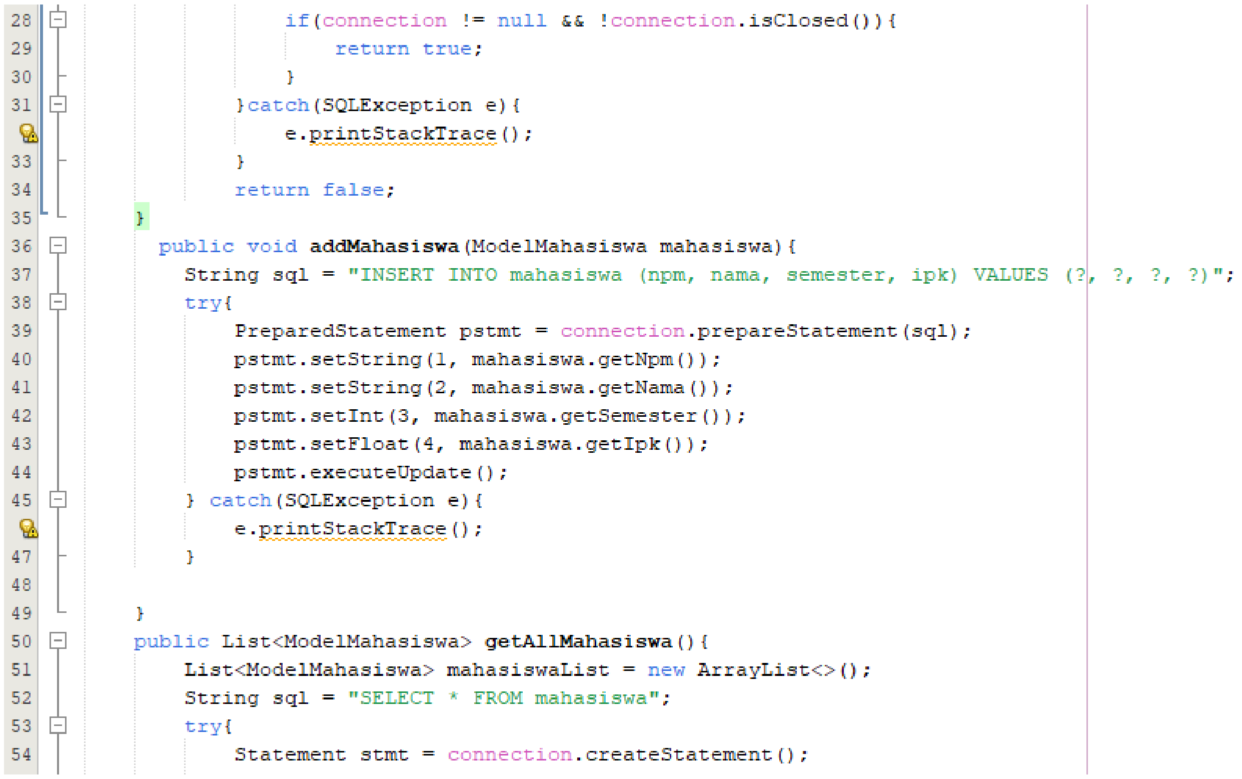


```java
    }
}
```

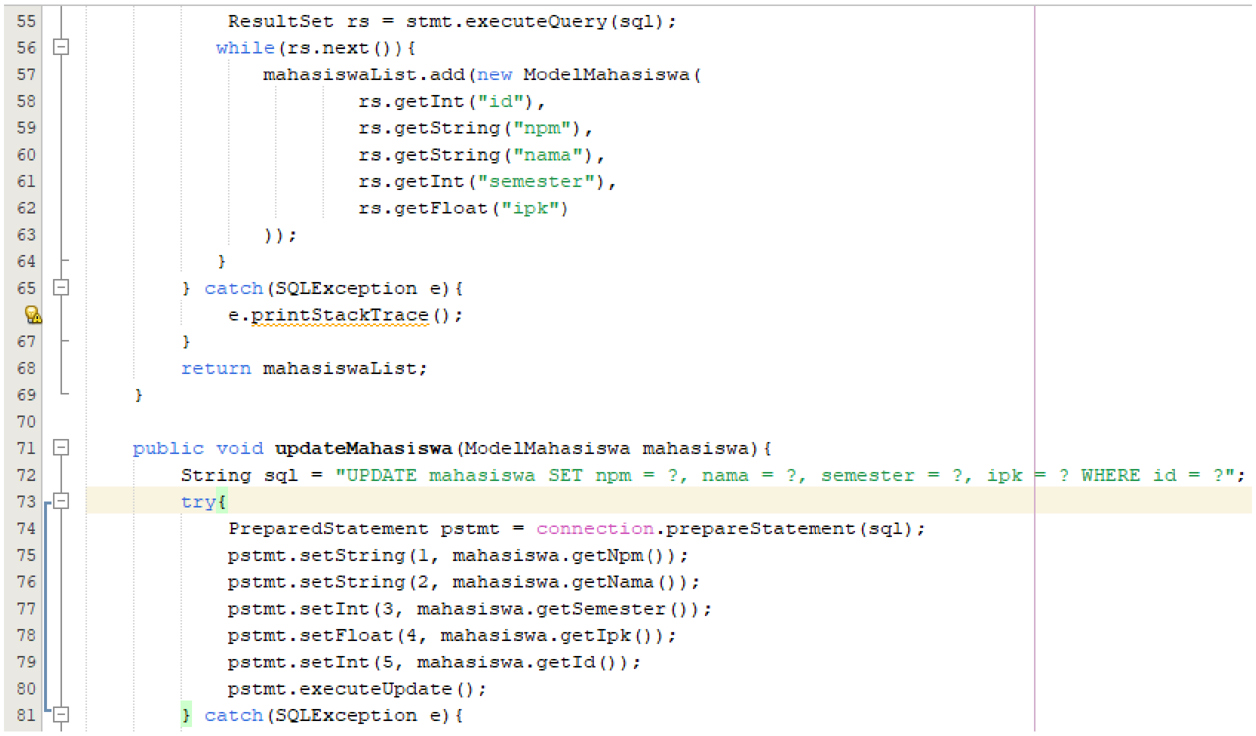
MahasiswaDAO



```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mahasiswa.model;

import java.sql.*;
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author LAB_CENGKARENG
 */
public class MahasiswaDAO{
        private Connection connection;

        public MahasiswaDAO(){
            try{
                Class.forName("com.mysql.cj.jdbc.Driver");
                connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/shaun_mvc", "root", "");
            } catch (Exception e) {
                e.printStackTrace();
            }
}
    public boolean checkConnection(){
            try{
```

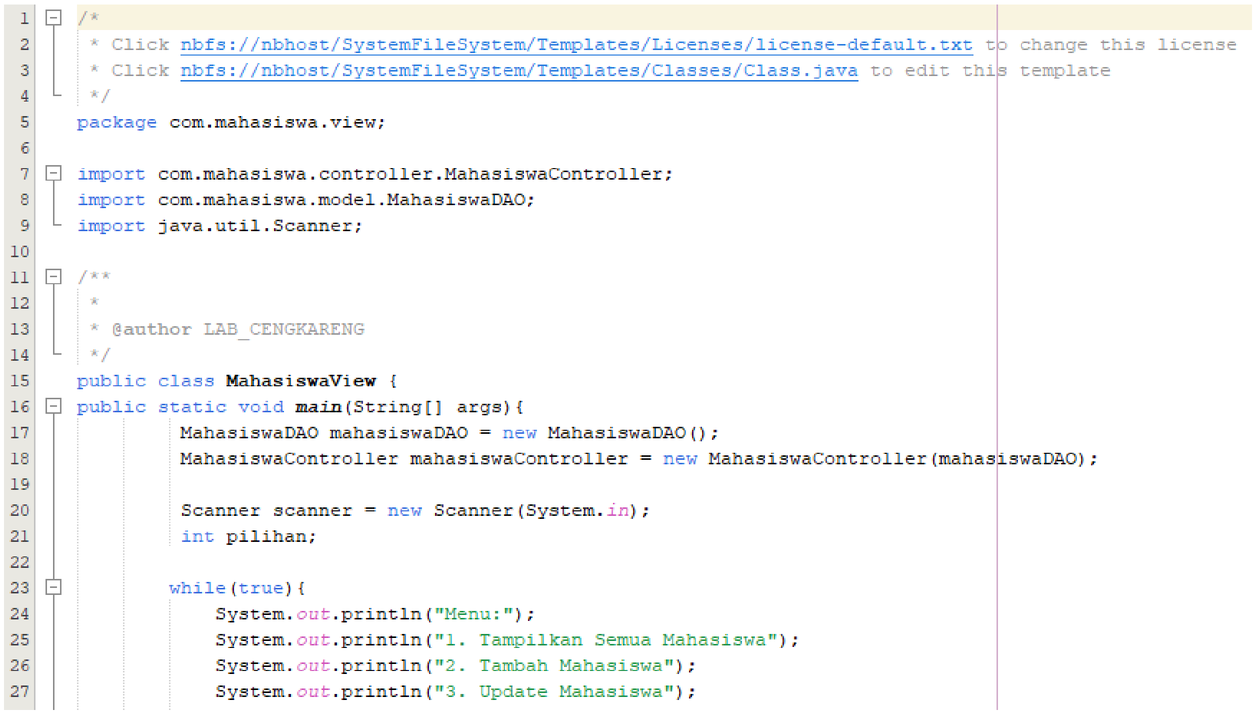


```java
                if(connection != null && !connection.isClosed()){
                    return true;
                }
            }catch(SQLException e){
                e.printStackTrace();
            }
            return false;
        }
        public void addMahasiswa(ModelMahasiswa mahasiswa){
          String sql = "INSERT INTO mahasiswa (npm, nama, semester, ipk) VALUES (?, ?, ?, ?)";
          try{
              PreparedStatement pstmt = connection.prepareStatement(sql);
              pstmt.setString(1, mahasiswa.getNpm());
              pstmt.setString(2, mahasiswa.getNama());
              pstmt.setInt(3, mahasiswa.getSemester());
              pstmt.setFloat(4, mahasiswa.getIpk());
              pstmt.executeUpdate();
          } catch(SQLException e){
              e.printStackTrace();
          }

        }
        public List<ModelMahasiswa> getAllMahasiswa(){
            List<ModelMahasiswa> mahasiswaList = new ArrayList<>();
            String sql = "SELECT * FROM mahasiswa";
            try{
                Statement stmt = connection.createStatement();
```

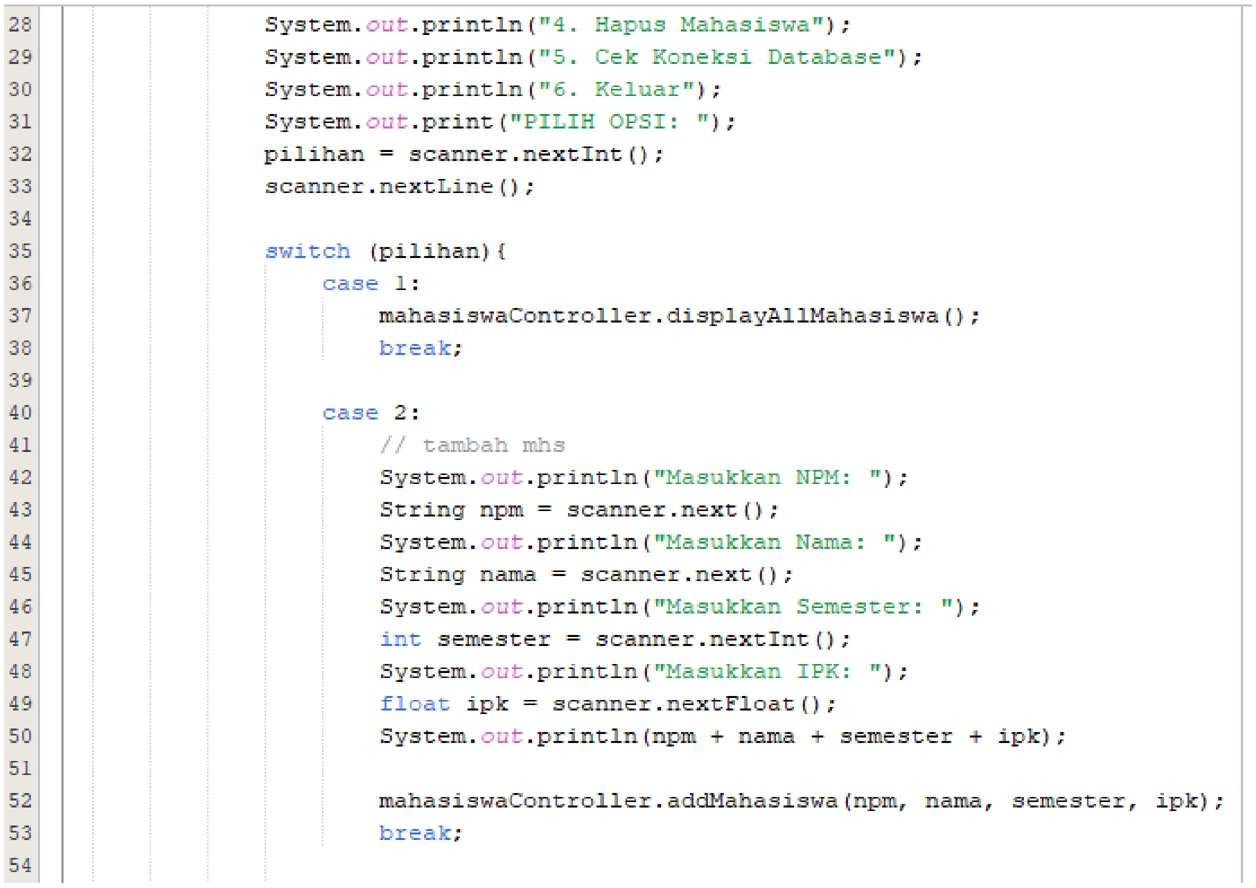


```java
                ResultSet rs = stmt.executeQuery(sql);
               while(rs.next()){
                   mahasiswaList.add(new ModelMahasiswa(
                           rs.getInt("id"),
                           rs.getString("npm"),
                           rs.getString("nama"),
                           rs.getInt("semester"),
                           rs.getFloat("ipk")
                   ));
               }
            } catch(SQLException e){
                e.printStackTrace();
            }
            return mahasiswaList;
        }

        public void updateMahasiswa(ModelMahasiswa mahasiswa){
            String sql = "UPDATE mahasiswa SET npm = ?, nama = ?, semester = ?, ipk = ? WHERE id = ?";
            try{
                PreparedStatement pstmt = connection.prepareStatement(sql);
                pstmt.setString(1, mahasiswa.getNpm());
                pstmt.setString(2, mahasiswa.getNama());
                pstmt.setInt(3, mahasiswa.getSemester());
                pstmt.setFloat(4, mahasiswa.getIpk());
                pstmt.setInt(5, mahasiswa.getId());
                pstmt.executeUpdate();
            } catch(SQLException e){
```

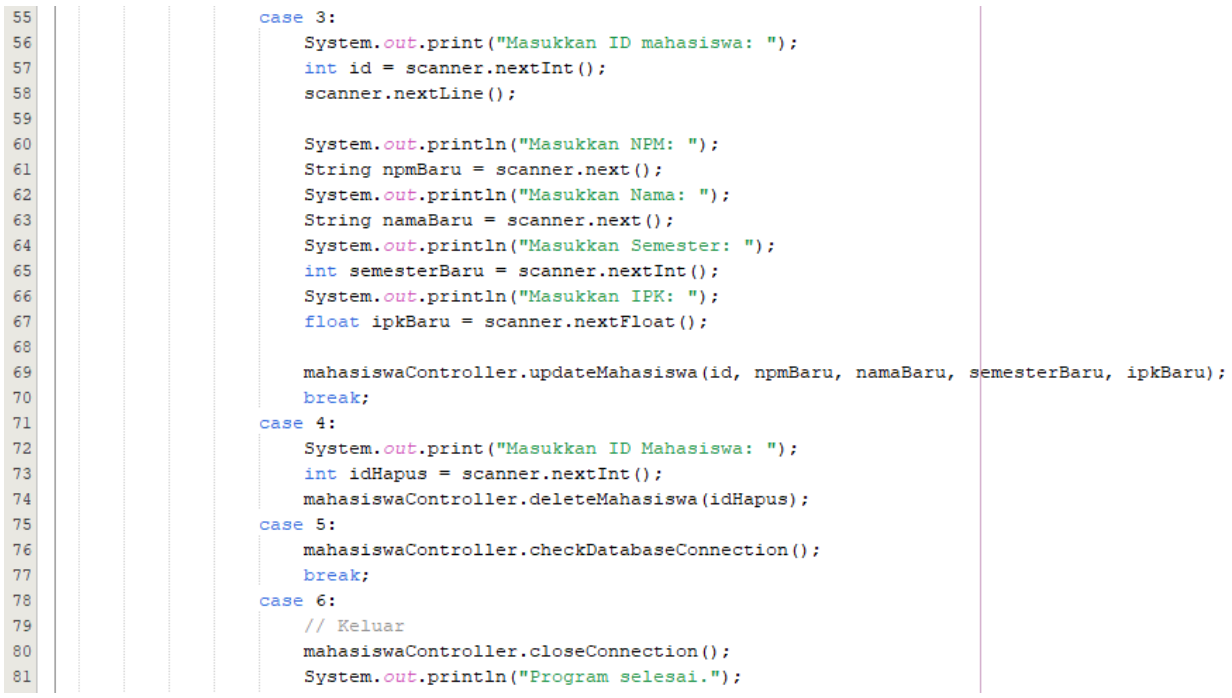
MahasiswaView



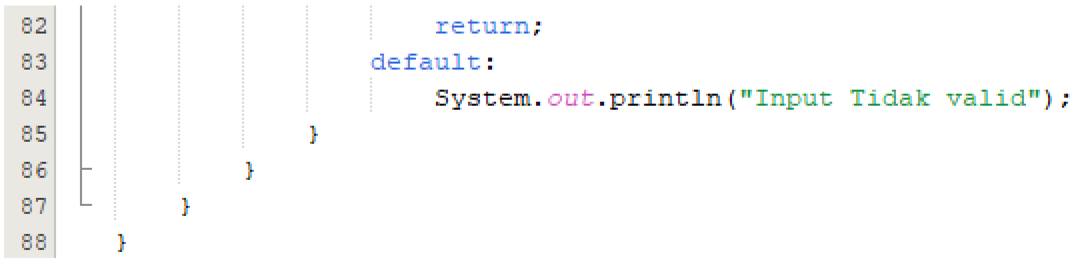
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mahasiswa.view;

import com.mahasiswa.controller.MahasiswaController;
import com.mahasiswa.model.MahasiswaDAO;
import java.util.Scanner;

/**
 *
 * @author LAB_CENGKARENG
 */
public class MahasiswaView {
public static void main(String[] args){
         MahasiswaDAO mahasiswaDAO = new MahasiswaDAO();
         MahasiswaController mahasiswaController = new MahasiswaController(mahasiswaDAO);

         Scanner scanner = new Scanner(System.in);
         int pilihan;

        while(true){
            System.out.println("Menu:");
            System.out.println("1. Tampilkan Semua Mahasiswa");
            System.out.println("2. Tambah Mahasiswa");
            System.out.println("3. Update Mahasiswa");
```



```java
            System.out.println("4. Hapus Mahasiswa");
            System.out.println("5. Cek Koneksi Database");
            System.out.println("6. Keluar");
            System.out.print("PILIH OPSI: ");
            pilihan = scanner.nextInt();
            scanner.nextLine();

            switch (pilihan){
                case 1:
                    mahasiswaController.displayAllMahasiswa();
                    break;

                case 2:
                    // tambah mhs
                    System.out.println("Masukkan NPM: ");
                    String npm = scanner.next();
                    System.out.println("Masukkan Nama: ");
                    String nama = scanner.next();
                    System.out.println("Masukkan Semester: ");
                    int semester = scanner.nextInt();
                    System.out.println("Masukkan IPK: ");
                    float ipk = scanner.nextFloat();
                    System.out.println(npm + nama + semester + ipk);

                    mahasiswaController.addMahasiswa(npm, nama, semester, ipk);
                    break;

```

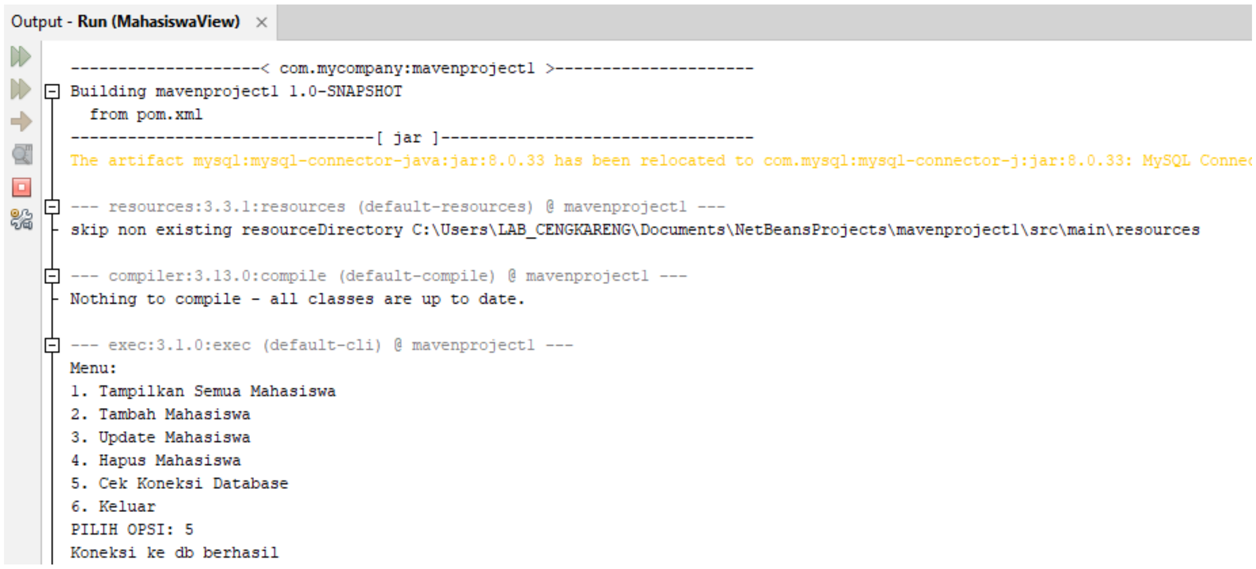


```java
                case 3:
                    System.out.print("Masukkan ID mahasiswa: ");
                    int id = scanner.nextInt();
                    scanner.nextLine();

                    System.out.println("Masukkan NPM: ");
                    String npmBaru = scanner.next();
                    System.out.println("Masukkan Nama: ");
                    String namaBaru = scanner.next();
                    System.out.println("Masukkan Semester: ");
                    int semesterBaru = scanner.nextInt();
                    System.out.println("Masukkan IPK: ");
                    float ipkBaru = scanner.nextFloat();

                    mahasiswaController.updateMahasiswa(id, npmBaru, namaBaru, semesterBaru, ipkBaru);
                    break;
                case 4:
                    System.out.print("Masukkan ID Mahasiswa: ");
                    int idHapus = scanner.nextInt();
                    mahasiswaController.deleteMahasiswa(idHapus);
                case 5:
                    mahasiswaController.checkDatabaseConnection();
                    break;
                case 6:
                    // Keluar
                    mahasiswaController.closeConnection();
                    System.out.println("Program selesai.");
```



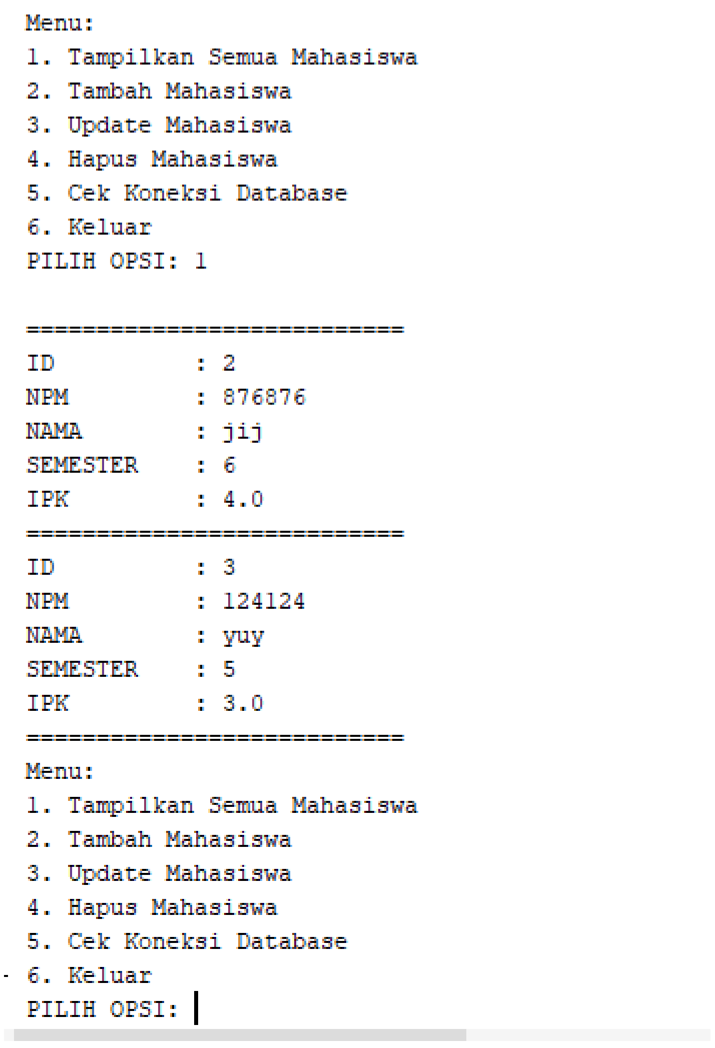
```
                    return;
                default:
                    System.out.println("Input Tidak valid");
            }
        }
    }
}
```

Output:



Output - Run (MahasiswaView)

```
--------------------< com.mycompany:mavenproject1 >---------------------
Building mavenproject1 1.0-SNAPSHOT
  from pom.xml
--------------------------------[ jar ]---------------------------------
The artifact mysql:mysql-connector-java:jar:8.0.33 has been relocated to com.mysql:mysql-connector-j:jar:8.0.33: MySQL Connec

--- resources:3.3.1:resources (default-resources) @ mavenproject1 ---
skip non existing resourceDirectory C:\Users\LAB_CENGKARENG\Documents\NetBeansProjects\mavenproject1\src\main\resources

--- compiler:3.13.0:compile (default-compile) @ mavenproject1 ---
Nothing to compile - all classes are up to date.

--- exec:3.1.0:exec (default-cli) @ mavenproject1 ---
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 5
Koneksi ke db berhasil
```

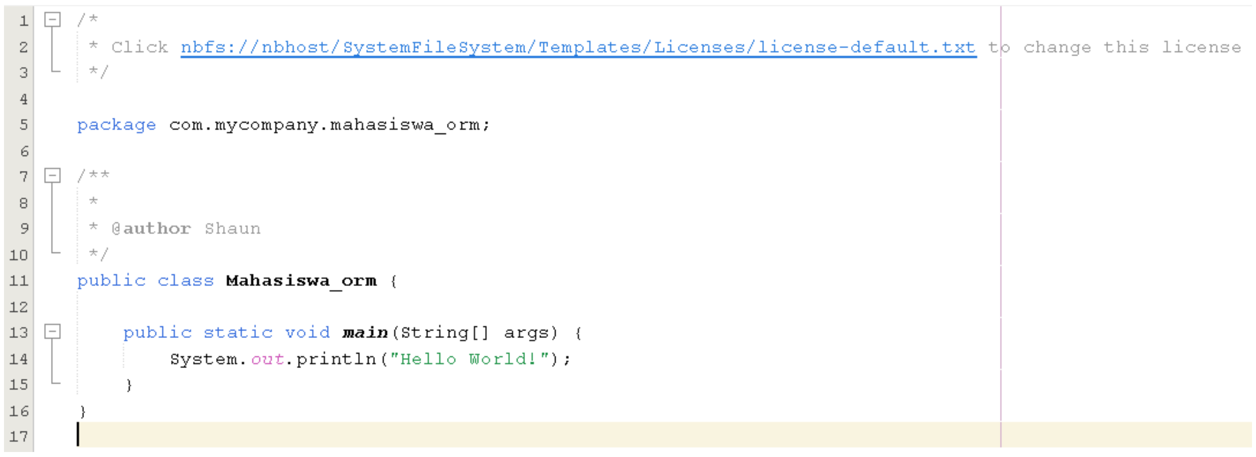


```
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 1

============================
ID          : 2
NPM         : 876876
NAMA        : jij
SEMESTER    : 6
IPK         : 4.0
============================
ID          : 3
NPM         : 124124
NAMA        : yuy
SEMESTER    : 5
IPK         : 3.0
============================
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI:
```

Pertemuan 4

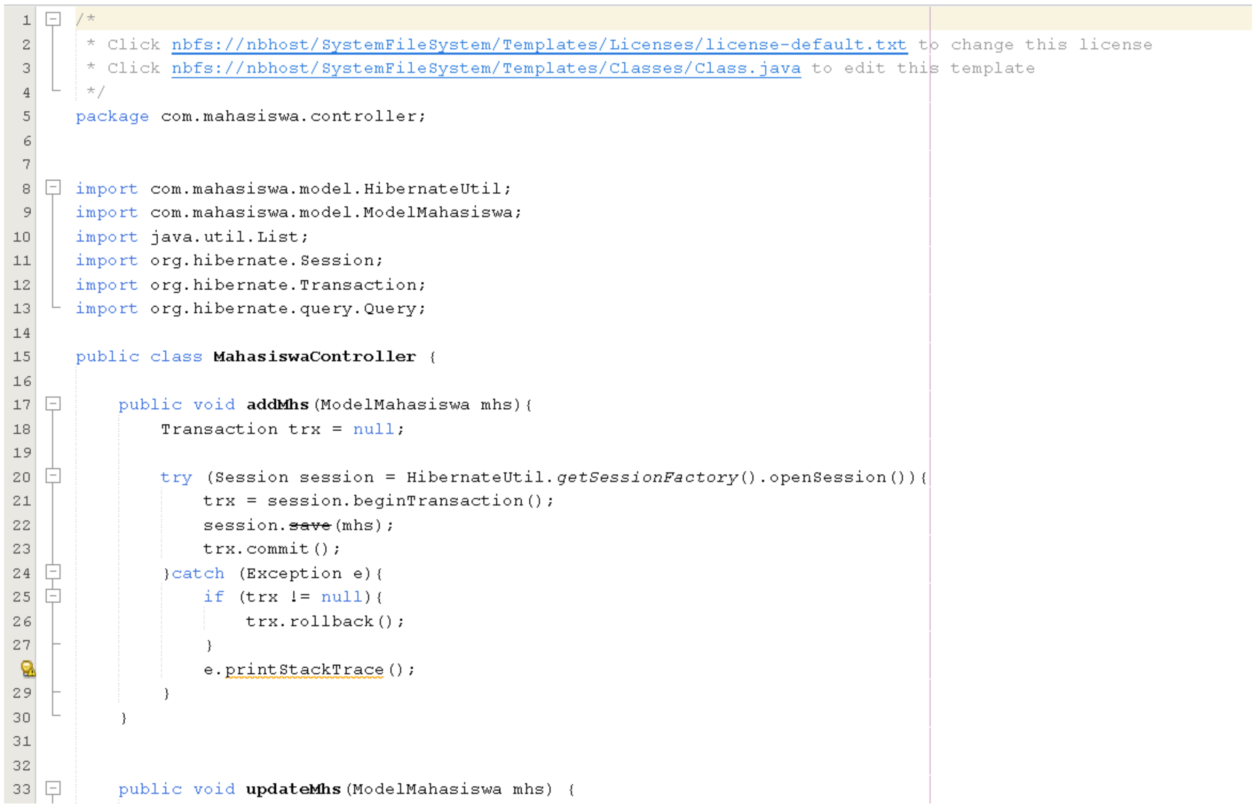
Source Code:

Mahasiswa_orm

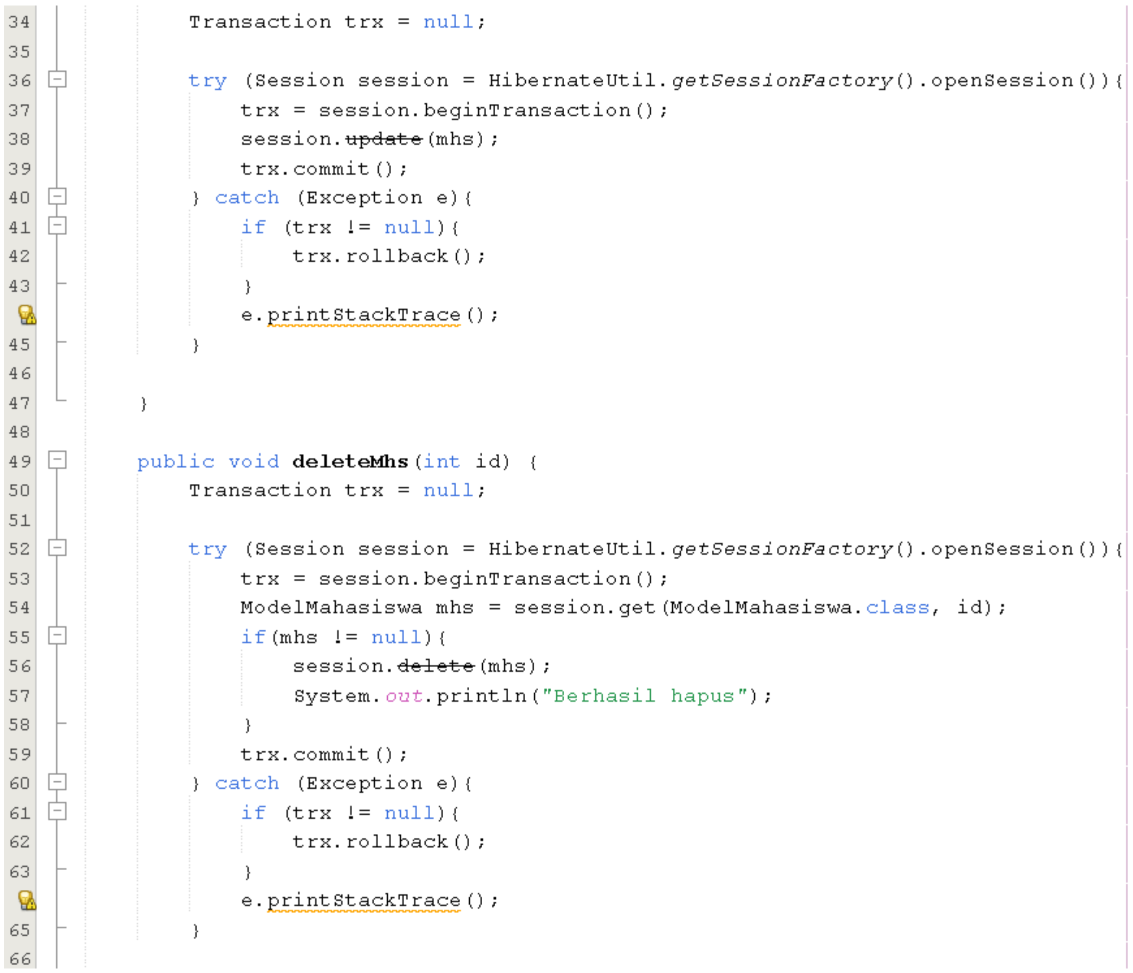


```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */

package com.mycompany.mahasiswa_orm;

/**
 *
 * @author Shaun
 */
public class Mahasiswa_orm {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}

```

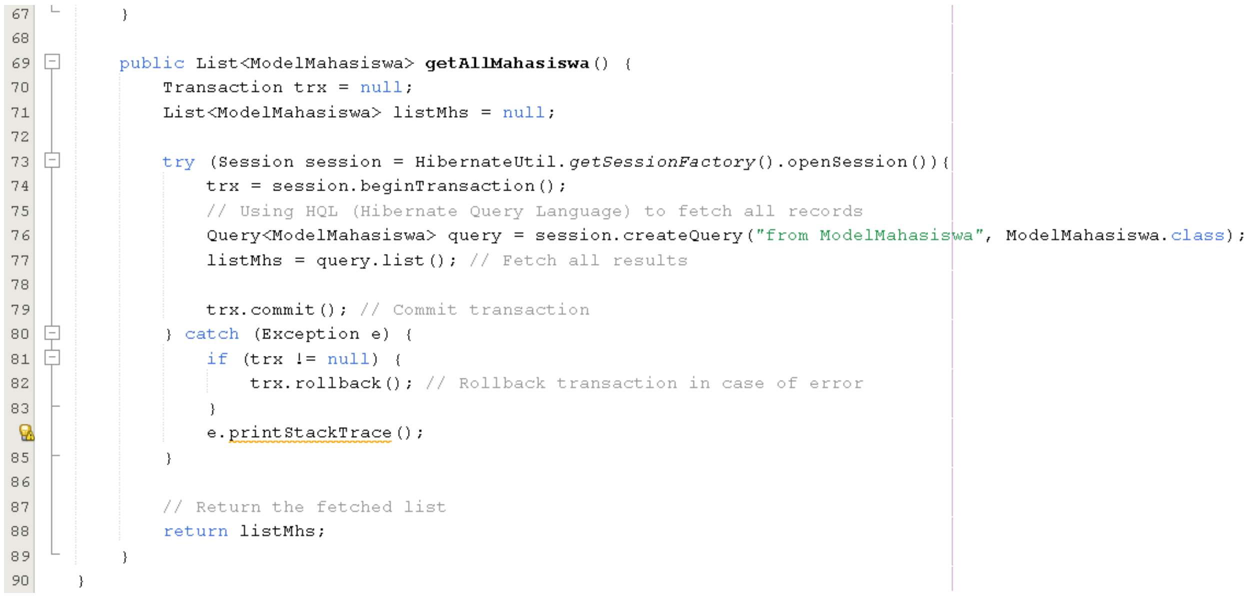
Mahasiswa Controller



```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mahasiswa.controller;


import com.mahasiswa.model.HibernateUtil;
import com.mahasiswa.model.ModelMahasiswa;
import java.util.List;
import org.hibernate.Session;
import org.hibernate.Transaction;
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) {
```

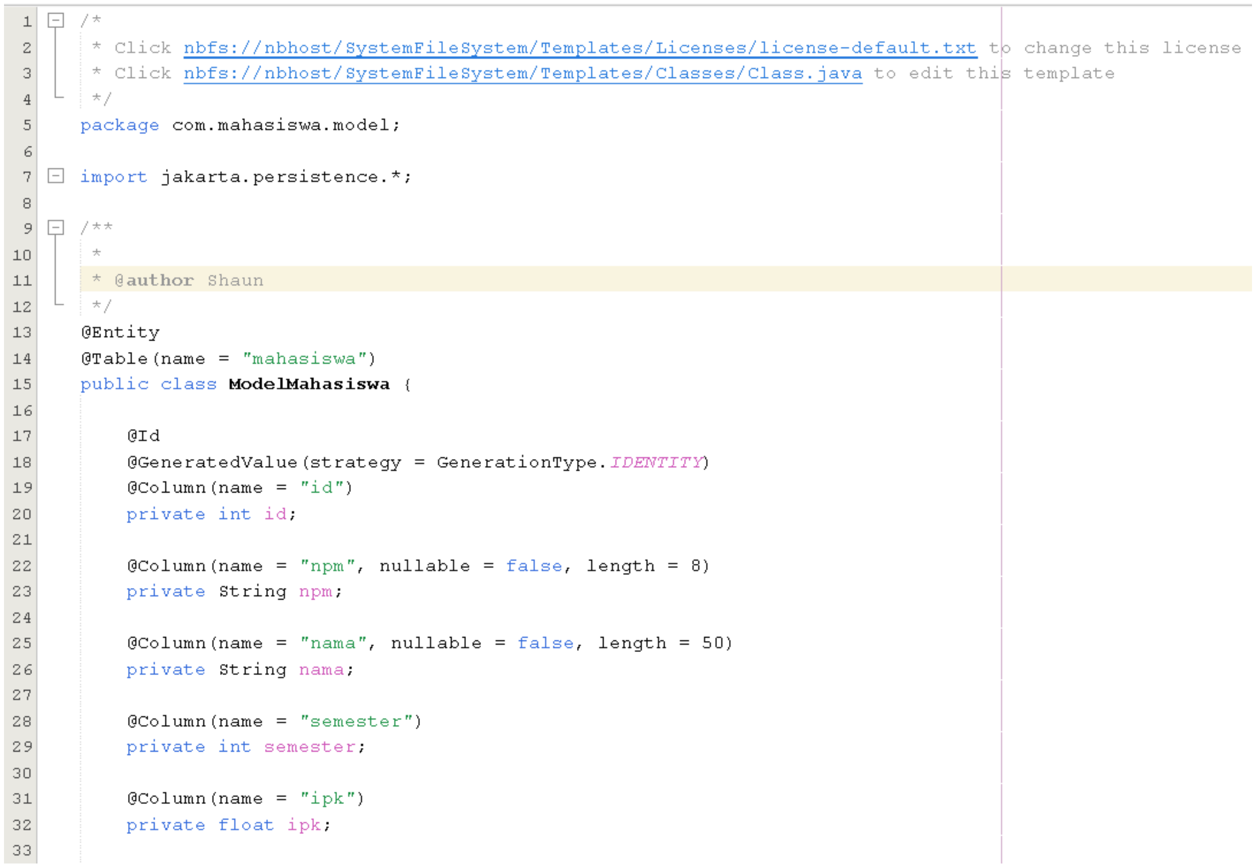


```java
        Transaction trx = null;

        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();
        }

```

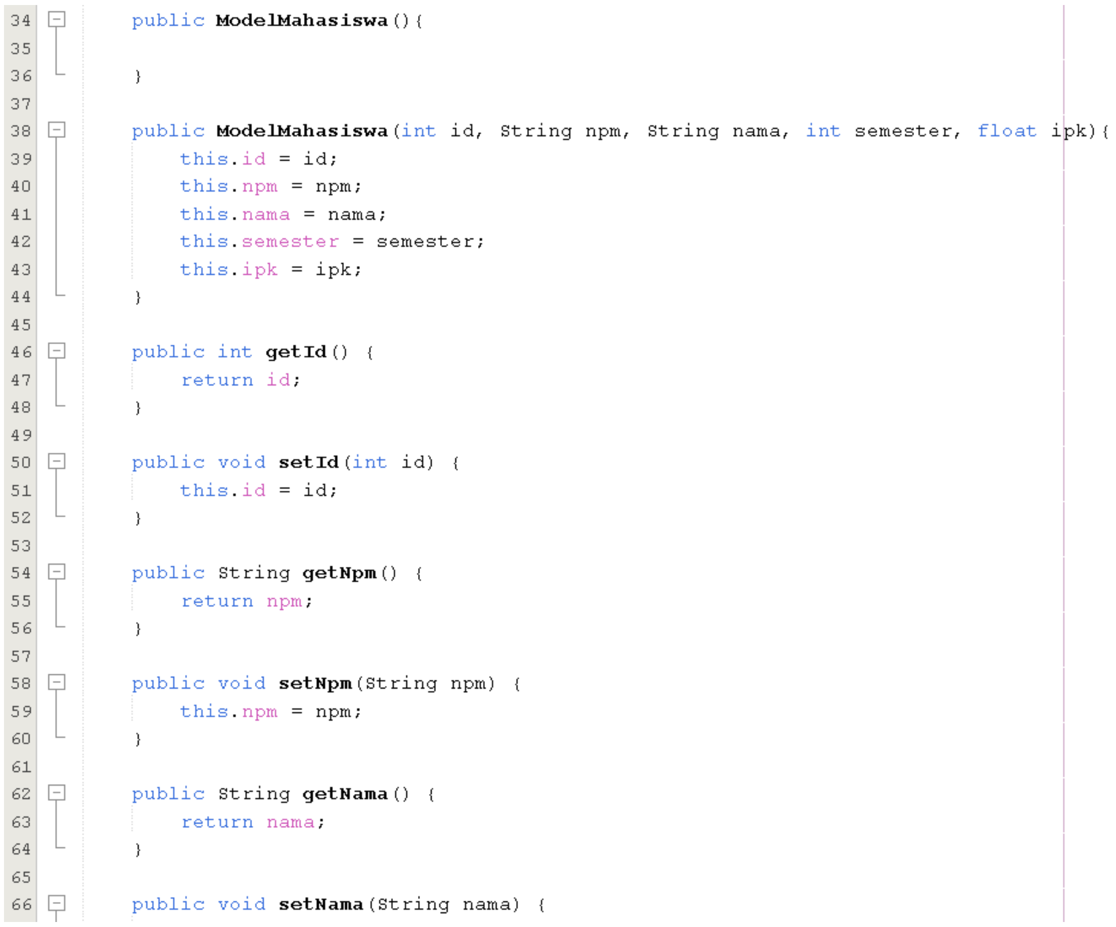


```java
    }

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

        try (Session session = HibernateUtil.getSessionFactory().openSession()){
            trx = session.beginTransaction();
            // Using HQL (Hibernate Query Language) to fetch all records
            Query<ModelMahasiswa> query = session.createQuery("from ModelMahasiswa", ModelMahasiswa.class);
            listMhs = query.list(); // Fetch all results

            trx.commit(); // Commit transaction
        } catch (Exception e) {
            if (trx != null) {
                trx.rollback(); // Rollback transaction in case of error
            }
            e.printStackTrace();
        }

        // Return the fetched list
        return listMhs;
    }
}
```

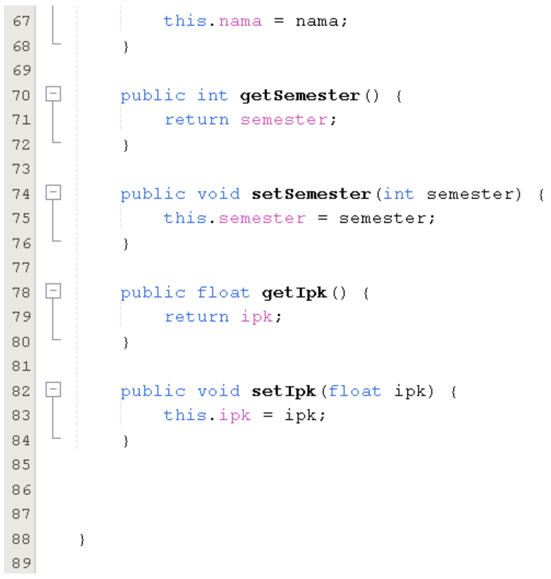
Model Mahasiswa



```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mahasiswa.model;

import jakarta.persistence.*;

/**
 *
 * @author Shaun
 */
@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;

```

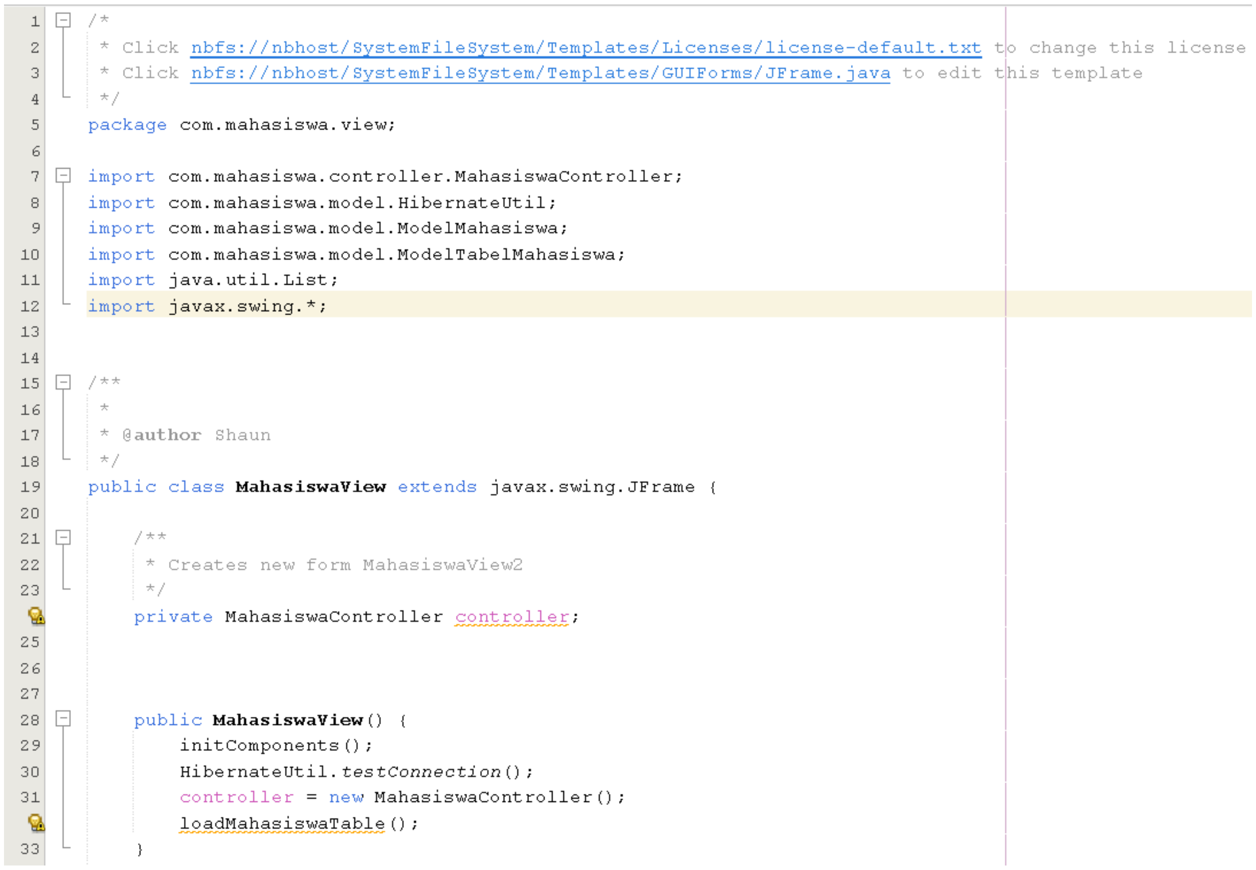


```java
    public ModelMahasiswa(){

    }

    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) {
```

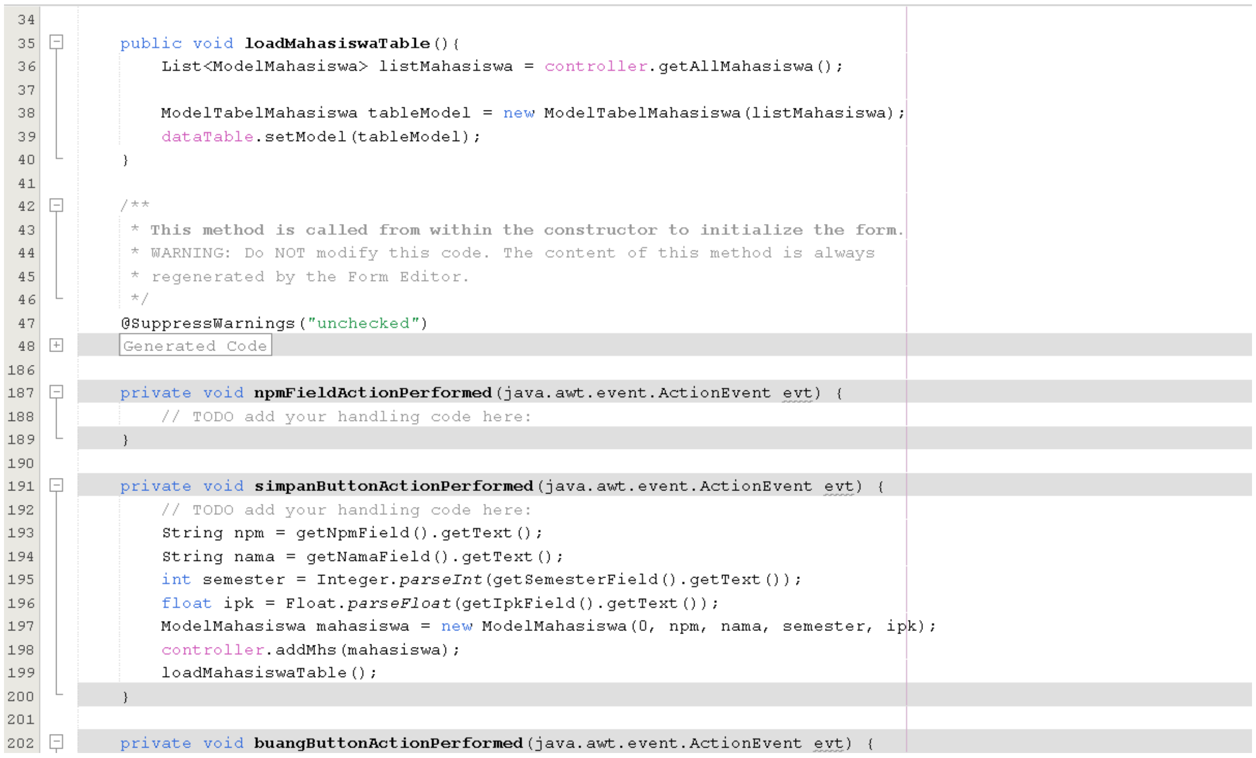


```java
        this.nama = nama;
    }

    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;
    }



}

```

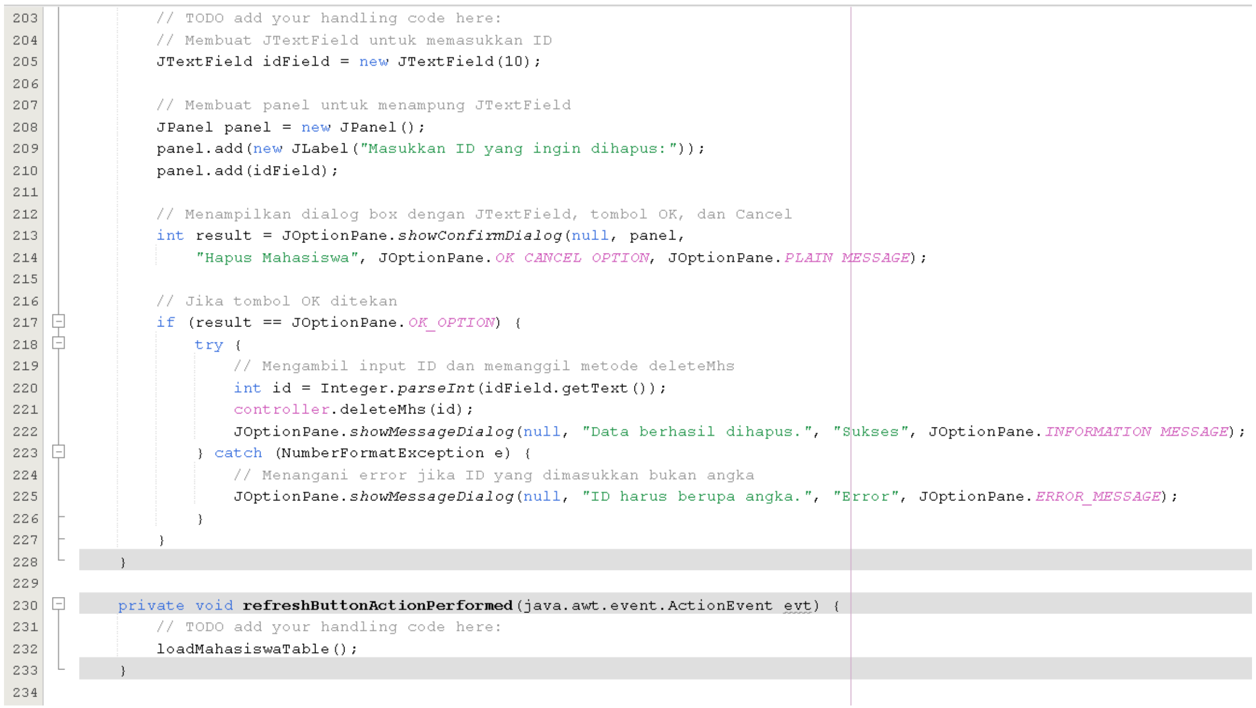
Mahasiswa VIew



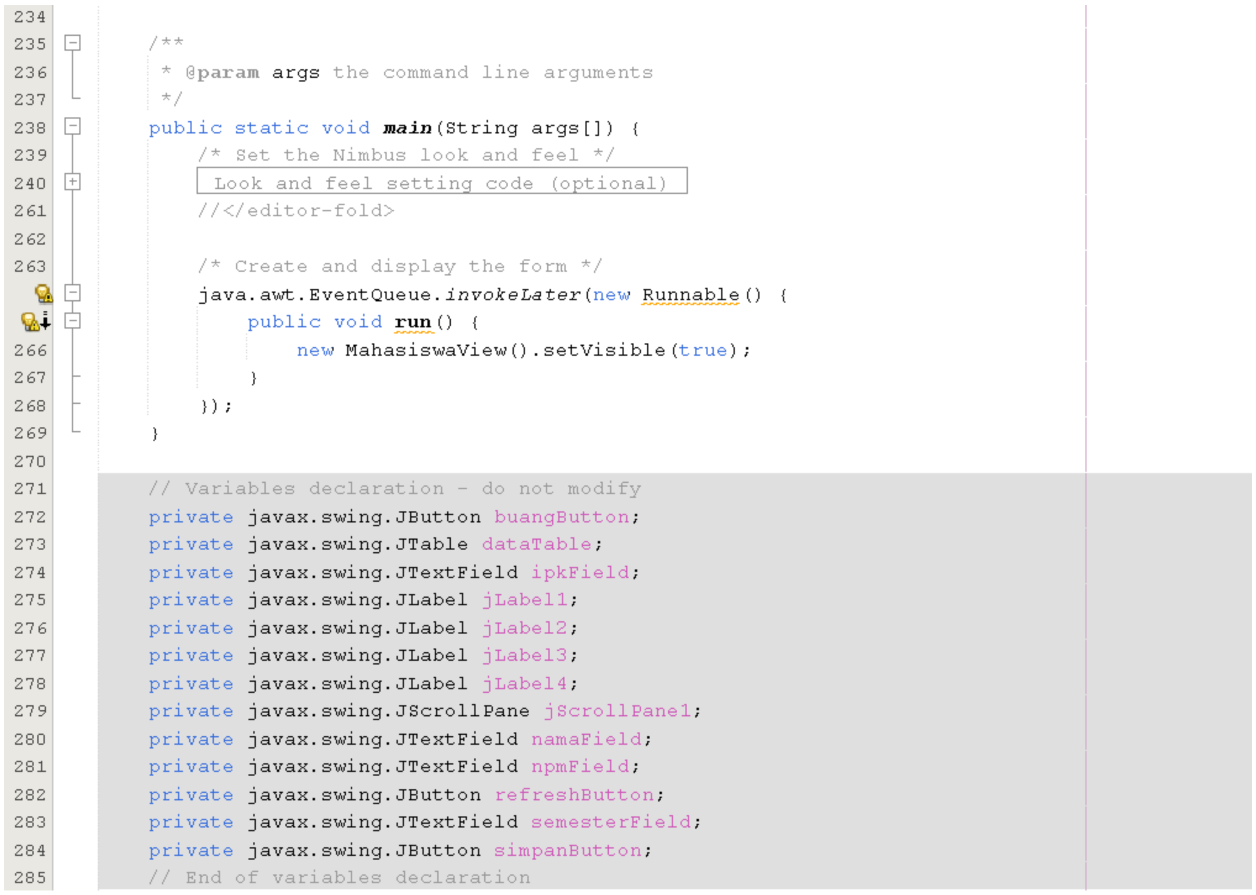
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
package com.mahasiswa.view;

import com.mahasiswa.controller.MahasiswaController;
import com.mahasiswa.model.HibernateUtil;
import com.mahasiswa.model.ModelMahasiswa;
import com.mahasiswa.model.ModelTabelMahasiswa;
import java.util.List;
import javax.swing.*;


/**
 *
 * @author Shaun
 */
public class MahasiswaView extends javax.swing.JFrame {

    /**
     * Creates new form MahasiswaView2
     */
    private MahasiswaController controller;



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



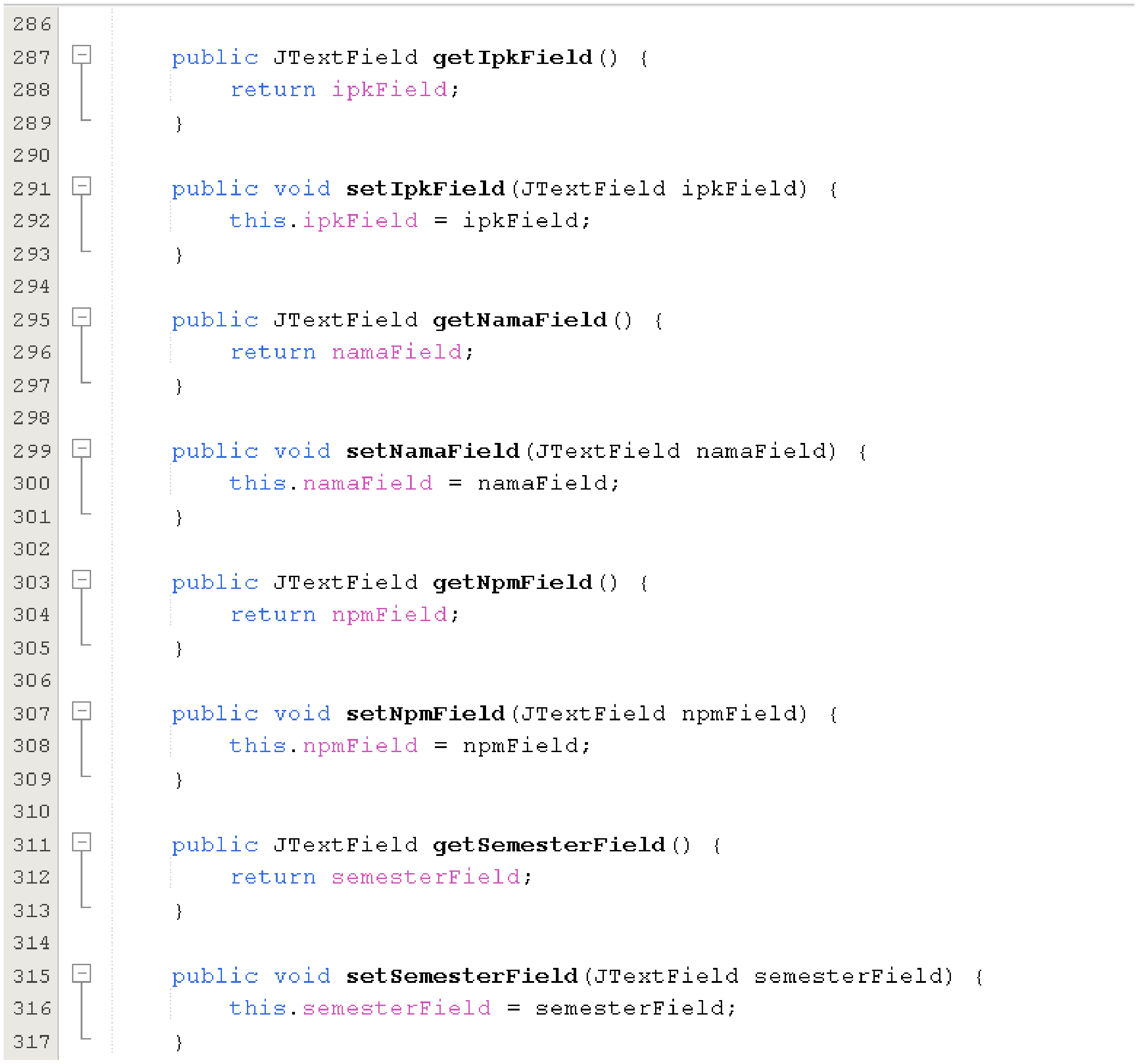
```

    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")
    Generated Code

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

    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);
        controller.addMhs(mahasiswa);
        loadMahasiswaTable();
    }

    private void buangButtonActionPerformed(java.awt.event.ActionEvent evt) {
```



```java
        // TODO add your handling code here:
        // Membuat JTextField untuk memasukkan ID
        JTextField idField = new JTextField(10);

        // Membuat panel untuk menampung JTextField
        JPanel panel = new JPanel();
        panel.add(new JLabel("Masukkan ID yang ingin dihapus:"));
        panel.add(idField);

        // Menampilkan dialog box dengan JTextField, tombol OK, dan Cancel
        int result = JOptionPane.showConfirmDialog(null, panel,
            "Hapus Mahasiswa", JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE);

        // Jika tombol OK ditekan
        if (result == JOptionPane.OK_OPTION) {
            try {
                // Mengambil input ID dan memanggil metode deleteMhs
                int id = Integer.parseInt(idField.getText());
                controller.deleteMhs(id);
                JOptionPane.showMessageDialog(null, "Data berhasil dihapus.", "Sukses", JOptionPane.INFORMATION_MESSAGE);
            } catch (NumberFormatException e) {
                // Menangani error jika ID yang dimasukkan bukan angka
                JOptionPane.showMessageDialog(null, "ID harus berupa angka.", "Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    }

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

```

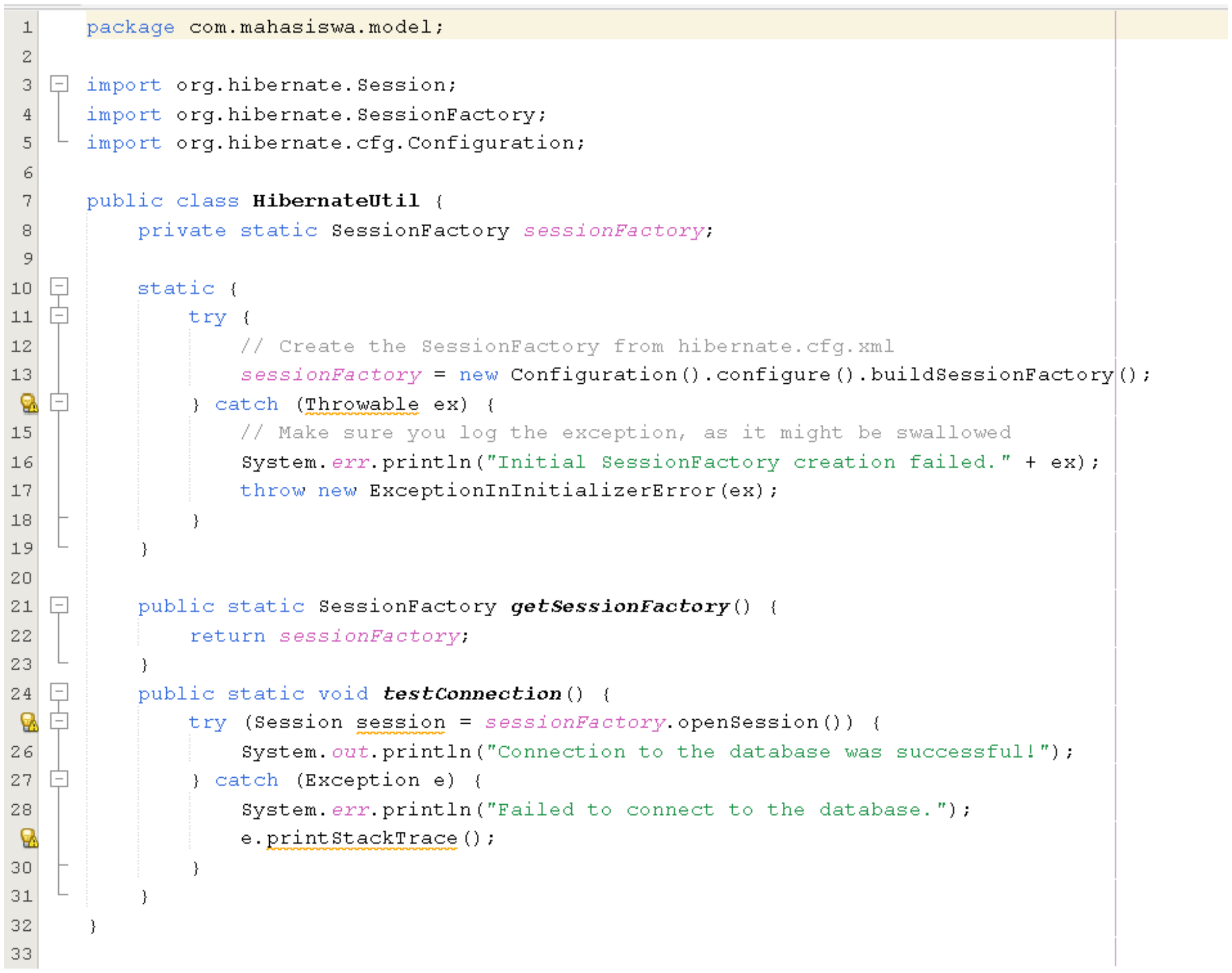


```java

    /**
     * @param args the command line arguments
     */
    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() {
                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
```

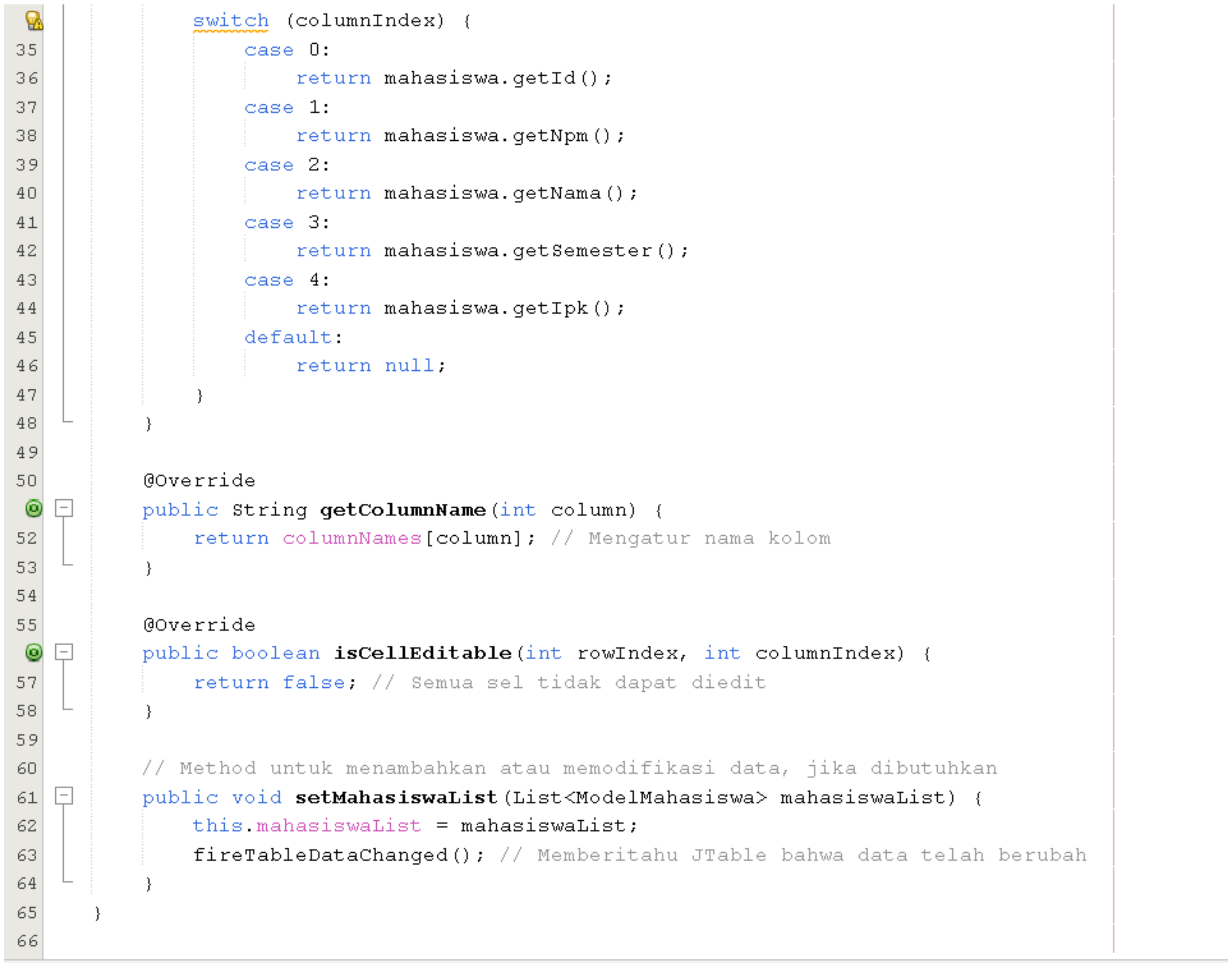


```java

    public JTextField getIpkField() {
        return ipkField;
    }

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

    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 JTextField getSemesterField() {
        return semesterField;
    }

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

Hibermate Util

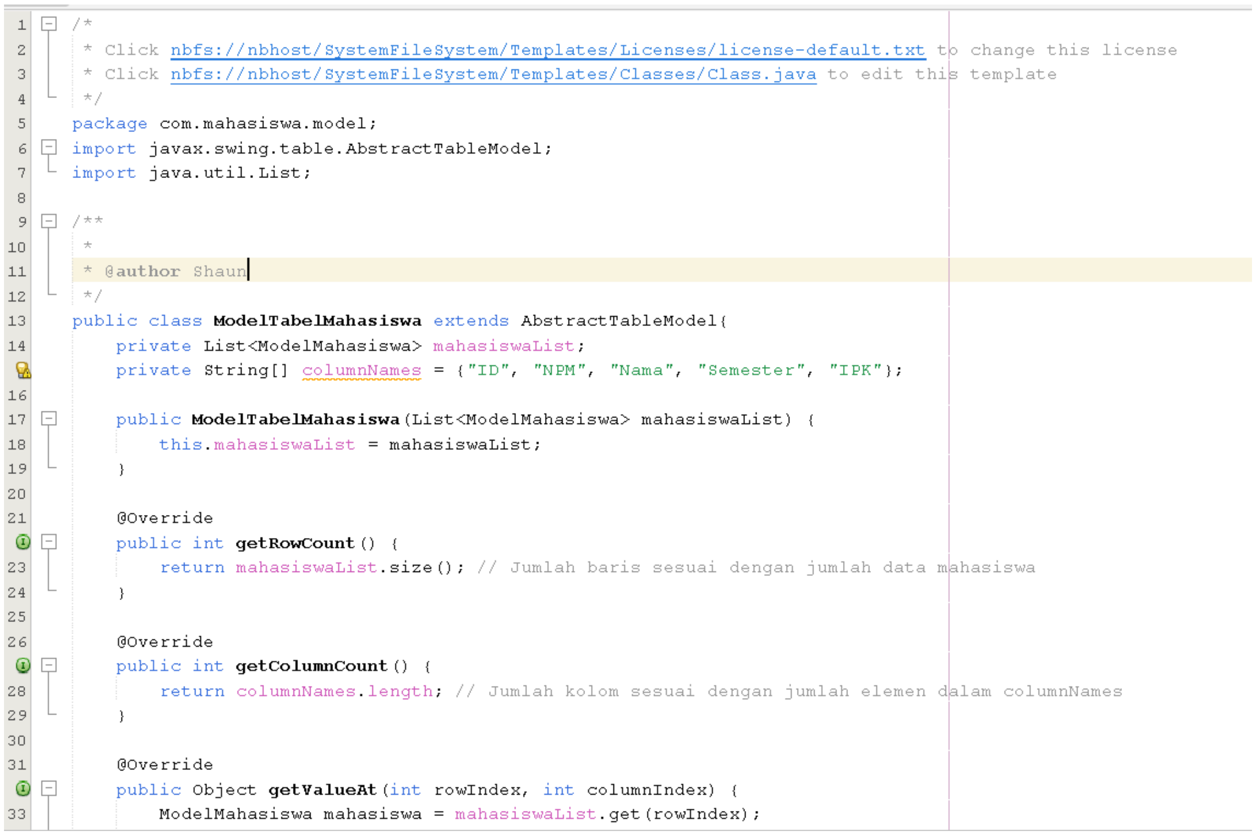


```java
package com.mahasiswa.model;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class HibernateUtil {
    private static SessionFactory sessionFactory;

    static {
        try {
            // Create the SessionFactory from hibernate.cfg.xml
            sessionFactory = new Configuration().configure().buildSessionFactory();
        } catch (Throwable ex) {
            // Make sure you log the exception, as it might be swallowed
            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();
        }
    }
}
```



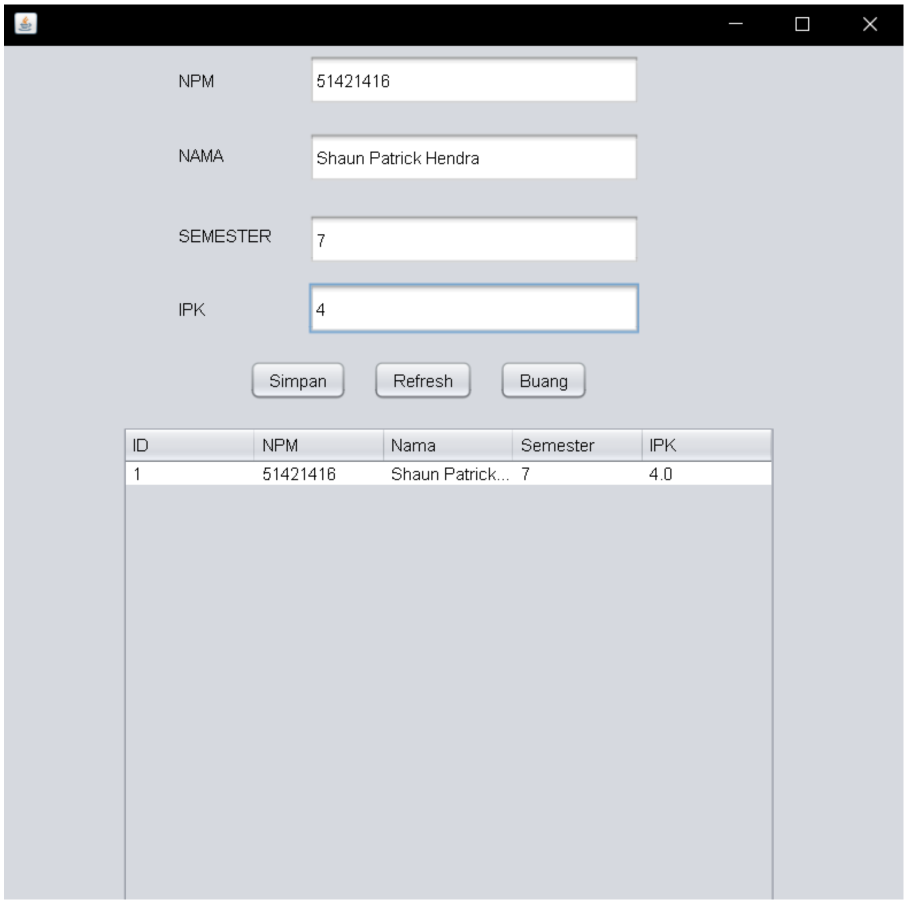
```
        switch (columnIndex) {
            case 0:
                return mahasiswa.getId();
            case 1:
                return mahasiswa.getNpm();
            case 2:
                return mahasiswa.getNama();
            case 3:
                return mahasiswa.getSemester();
            case 4:
                return mahasiswa.getIpk();
            default:
                return null;
        }
    }

    @Override
    public String getColumnName(int column) {
        return columnNames[column]; // Mengatur nama kolom
    }

    @Override
    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return false; // Semua sel tidak dapat diedit
    }

    // Method untuk menambahkan atau memodifikasi data, jika dibutuhkan
    public void setMahasiswaList(List<ModelMahasiswa> mahasiswaList) {
        this.mahasiswaList = mahasiswaList;
        fireTableDataChanged(); // Memberitahu JTable bahwa data telah berubah
    }
}
```

ModelTabelMahasiswa



```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mahasiswa.model;
import javax.swing.table.AbstractTableModel;
import java.util.List;

/**
 *
 * @author Shaun
 */
public class ModelTabelMahasiswa extends AbstractTableModel{
    private List<ModelMahasiswa> mahasiswaList;
    private String[] columnNames = {"ID", "NPM", "Nama", "Semester", "IPK"};

    public ModelTabelMahasiswa(List<ModelMahasiswa> mahasiswaList) {
        this.mahasiswaList = mahasiswaList;
    }

    @Override
    public int getRowCount() {
        return mahasiswaList.size(); // Jumlah baris sesuai dengan jumlah data mahasiswa
    }

    @Override
    public int getColumnCount() {
        return columnNames.length; // Jumlah kolom sesuai dengan jumlah elemen dalam columnNames
    }

    @Override
    public Object getValueAt(int rowIndex, int columnIndex) {
        ModelMahasiswa mahasiswa = mahasiswaList.get(rowIndex);
```

Output:



NPM 51421416

NAMA Shaun Patrick Hendra

SEMESTER 7

IPK 4

ID	NPM	Nama	Semester	IPK
1	51421416	Shaun Patrick...	7	4.0