Nama : **Muhammad Izzuddin Almansyur**

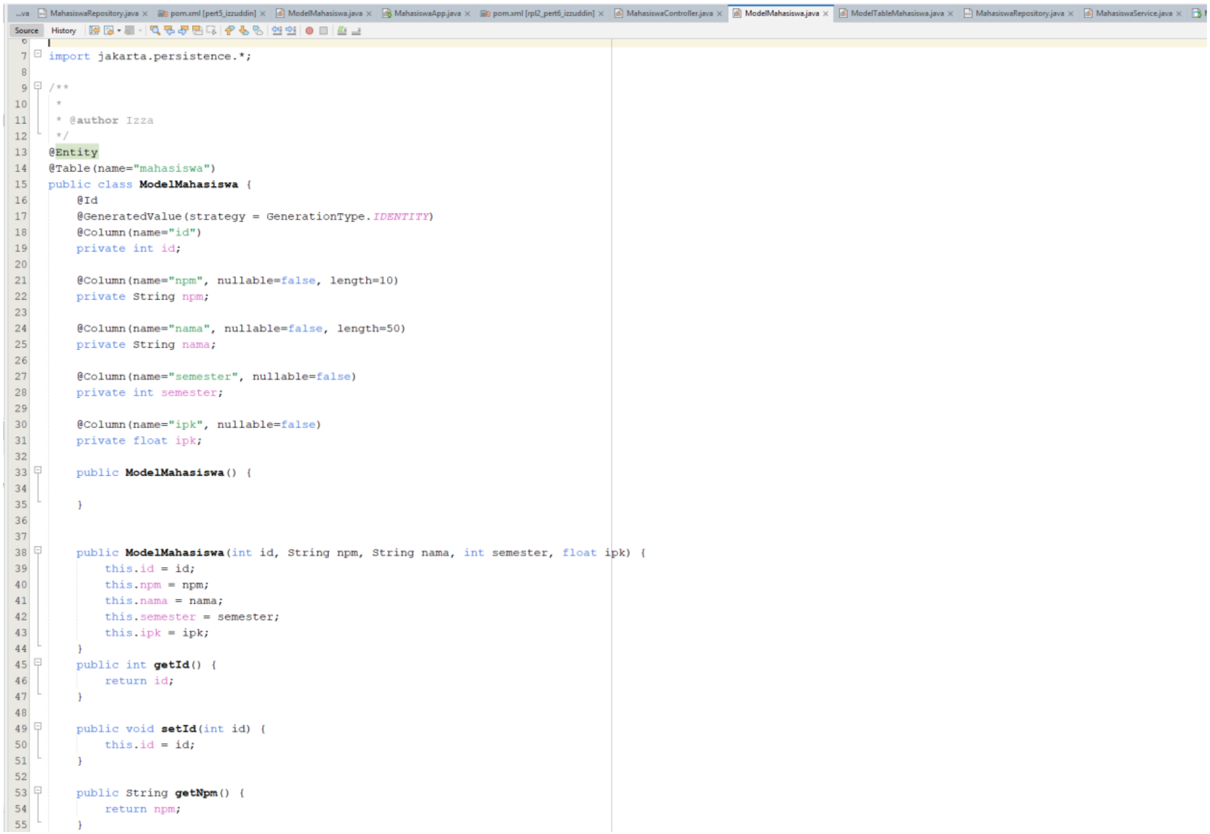
NPM : **51422084**

Kelas : **4IA07**

Praktikum : **Rekayasa perangkat Lunak 2, ACT6**

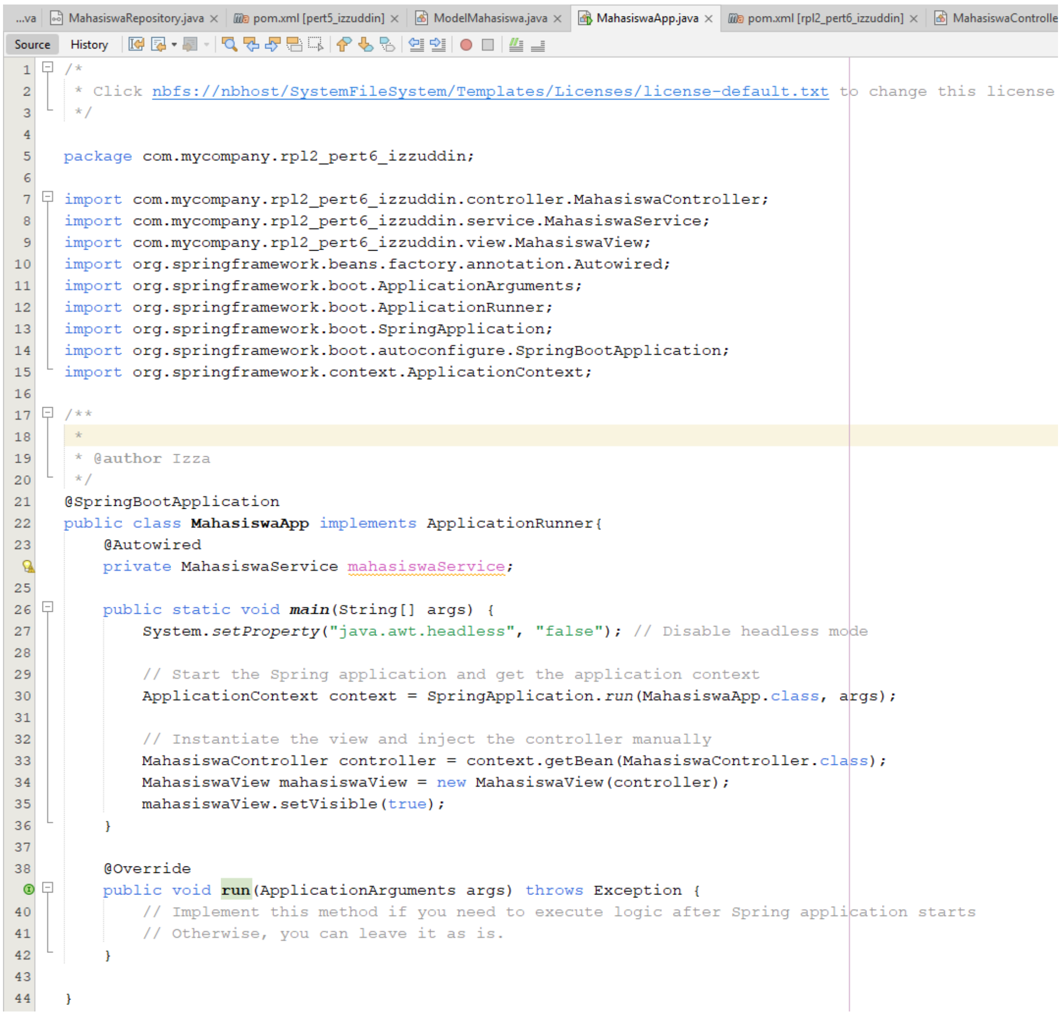
1. Screenshoot kode dan output program

- ModelMahasiswa.java



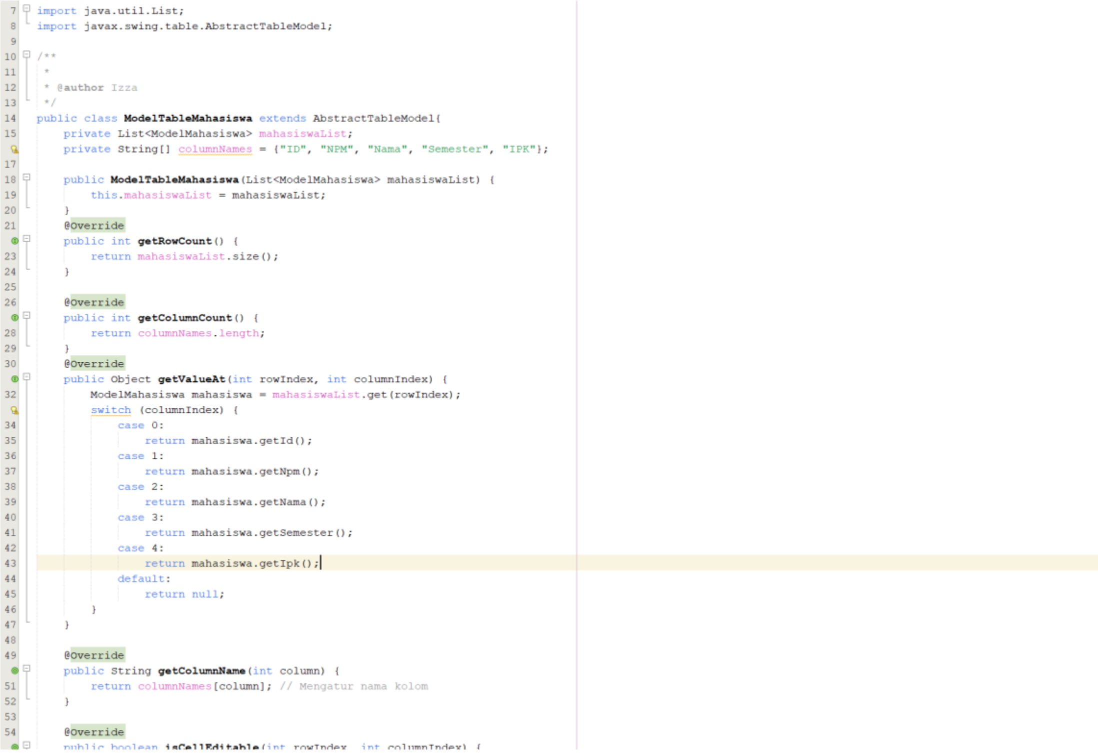
```java

import jakarta.persistence.*;

/**
 *
 * @author Izza
 */
@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=50)
    private String nama;

    @Column(name="semester", nullable=false)
    private int semester;

    @Column(name="ipk", nullable=false)
    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;
    }
```

- MasiswaApp.java



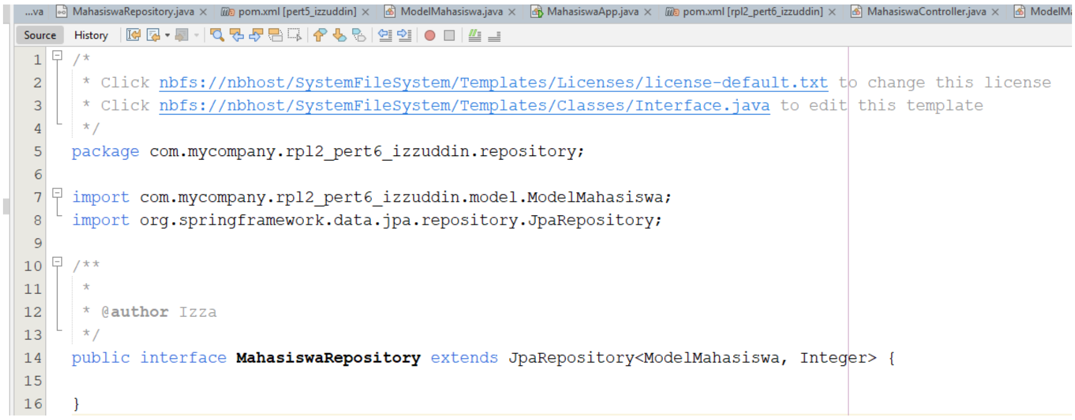
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */

package com.mycompany.rpl2_pert6_izzuddin;

import com.mycompany.rpl2_pert6_izzuddin.controller.MahasiswaController;
import com.mycompany.rpl2_pert6_izzuddin.service.MahasiswaService;
import com.mycompany.rpl2_pert6_izzuddin.view.MahasiswaView;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.ApplicationArguments;
import org.springframework.boot.ApplicationRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;

/**
 *
 * @author Izza
 */
@SpringBootApplication
public class MahasiswaApp implements ApplicationRunner{
    @Autowired
    private MahasiswaService mahasiswaService;

    public static void main(String[] args) {
        System.setProperty("java.awt.headless", "false"); // Disable headless mode

        // Start the Spring application and get the application context
        ApplicationContext context = SpringApplication.run(MahasiswaApp.class, args);

        // Instantiate the view and inject the controller manually
        MahasiswaController controller = context.getBean(MahasiswaController.class);
        MahasiswaView mahasiswaView = new MahasiswaView(controller);
        mahasiswaView.setVisible(true);
    }

    @Override
    public void run(ApplicationArguments args) throws Exception {
        // Implement this method if you need to execute logic after Spring application starts
        // Otherwise, you can leave it as is.
    }

}
```

- ModelTableMahasiswa.java



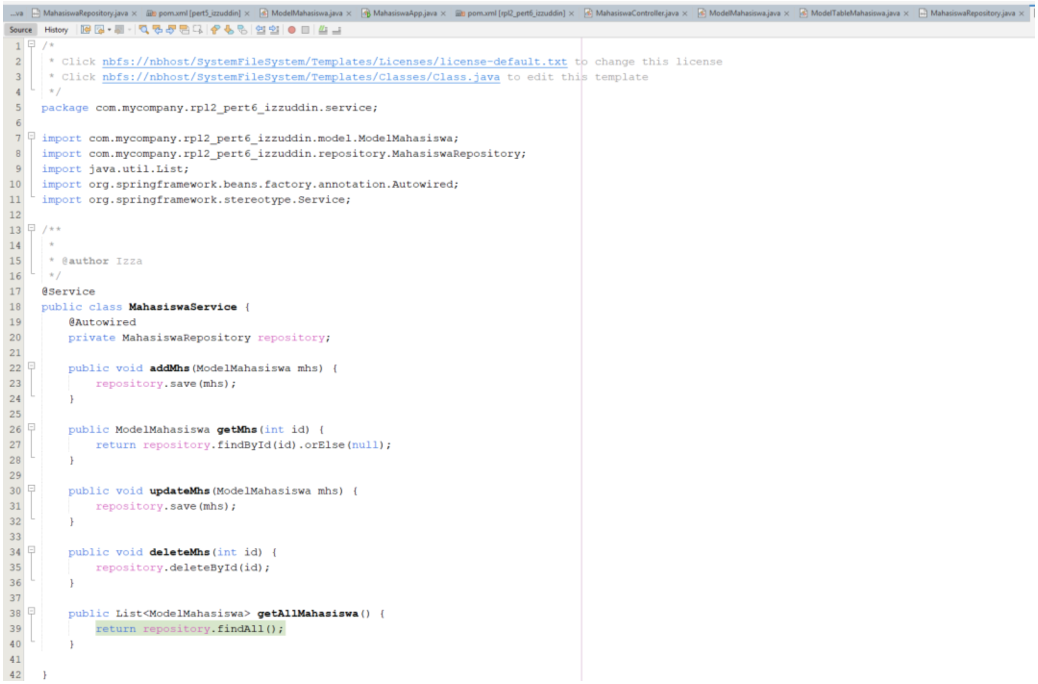
```java
import java.util.List;
import javax.swing.table.AbstractTableModel;

/**
 *
 * @author Izza
 */
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.getIpk();
            default:
                return null;
        }
    }

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

    @Override
    public boolean isCellEditable(int rowIndex, int columnIndex) {
```

- MahasiswaRepository.java



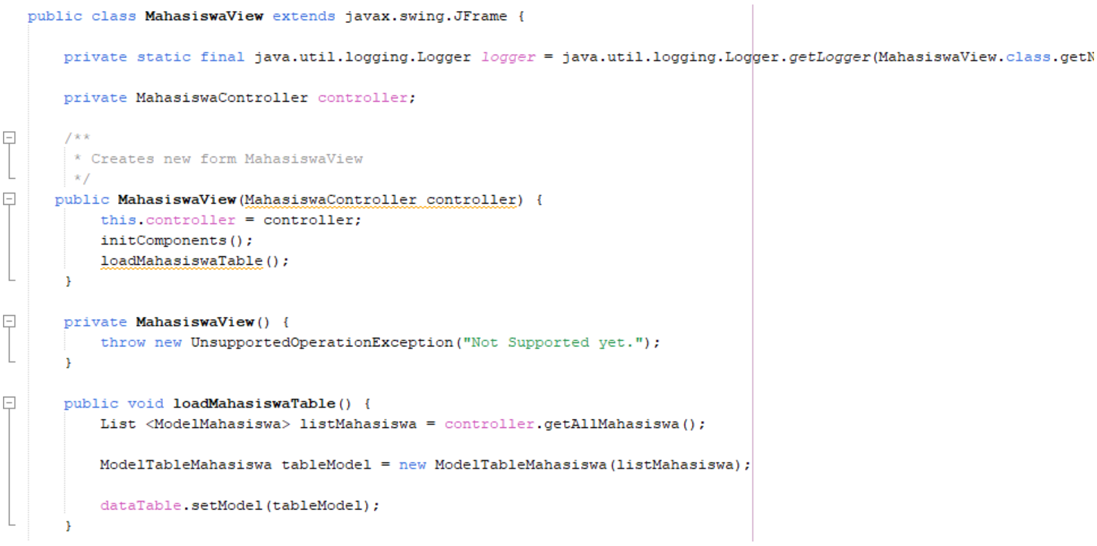
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Interface.java to edit this template
 */
package com.mycompany.rpl2_pert6_izzuddin.repository;

import com.mycompany.rpl2_pert6_izzuddin.model.ModelMahasiswa;
import org.springframework.data.jpa.repository.JpaRepository;

/**
 *
 * @author Izza
 */
public interface MahasiswaRepository extends JpaRepository<ModelMahasiswa, Integer> {

}
```

- MahasiswaService.java

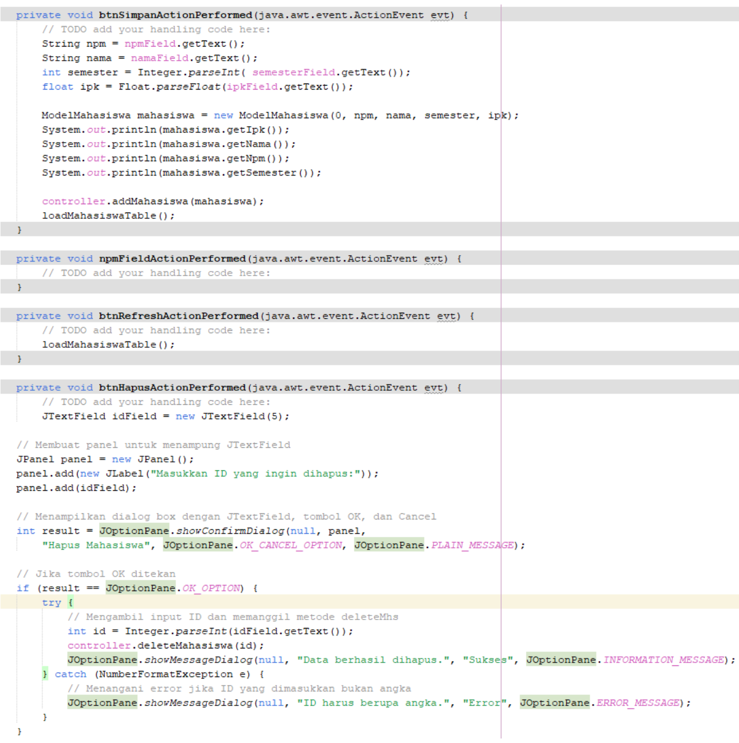


```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mycompany.rpl2_pert6_izzuddin.service;

import com.mycompany.rpl2_pert6_izzuddin.model.ModelMahasiswa;
import com.mycompany.rpl2_pert6_izzuddin.repository.MahasiswaRepository;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

/**
 *
 * @author Izza
 */
@Service
public class MahasiswaService {
    @Autowired
    private MahasiswaRepository repository;

    public void addMhs(ModelMahasiswa mhs) {
        repository.save(mhs);
    }

    public ModelMahasiswa getMhs(int id) {
        return repository.findById(id).orElse(null);
    }

    public void updateMhs(ModelMahasiswa mhs) {
        repository.save(mhs);
    }

    public void deleteMhs(int id) {
        repository.deleteById(id);
    }

    public List<ModelMahasiswa> getAllMahasiswa() {
        return repository.findAll();
    }

}
```

- MahasiswaView.java

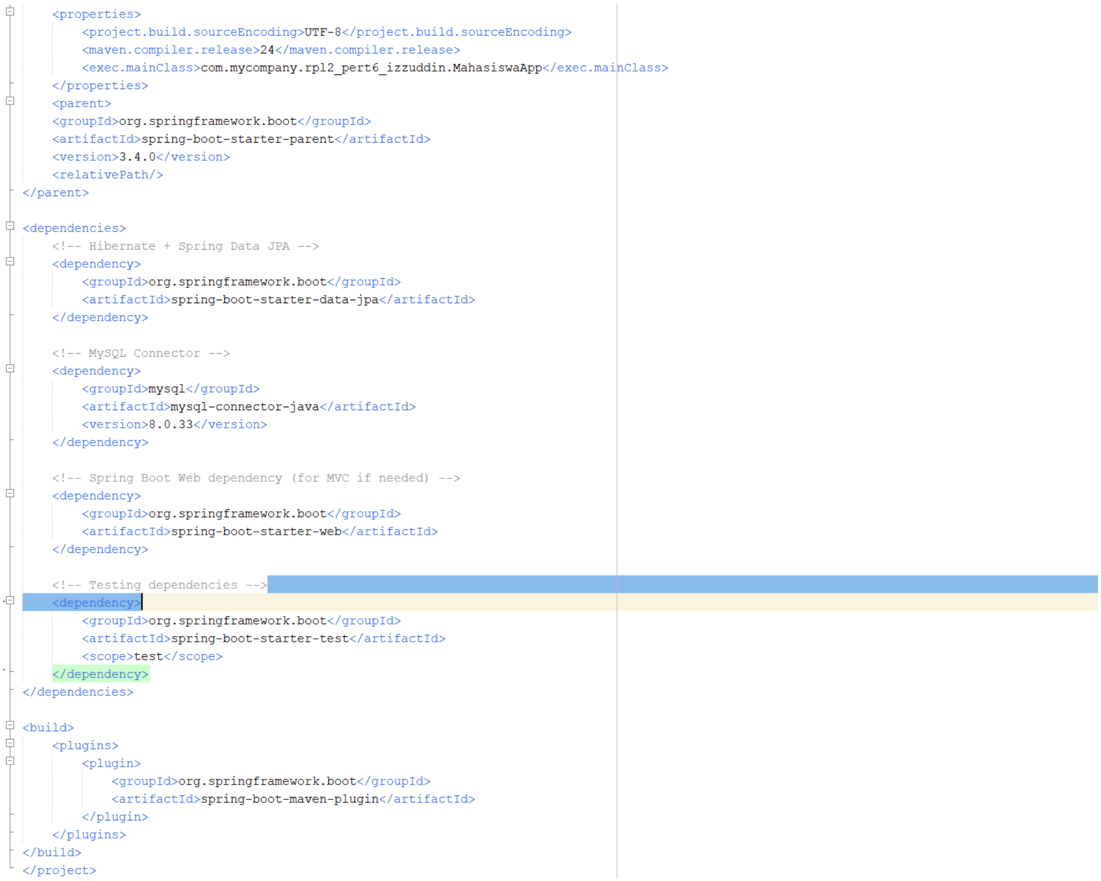


```java
public class MahasiswaView extends javax.swing.JFrame {

    private static final java.util.logging.Logger logger = java.util.logging.Logger.getLogger(MahasiswaView.class.getN

    private MahasiswaController controller;

    /**
     * Creates new form MahasiswaView
     */
   public MahasiswaView(MahasiswaController controller) {
        this.controller = controller;
        initComponents();
        loadMahasiswaTable();
    }

    private MahasiswaView() {
        throw new UnsupportedOperationException("Not Supported yet.");
    }

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

        ModelTableMahasiswa tableModel = new ModelTableMahasiswa(listMahasiswa);

        dataTable.setModel(tableModel);
    }
```



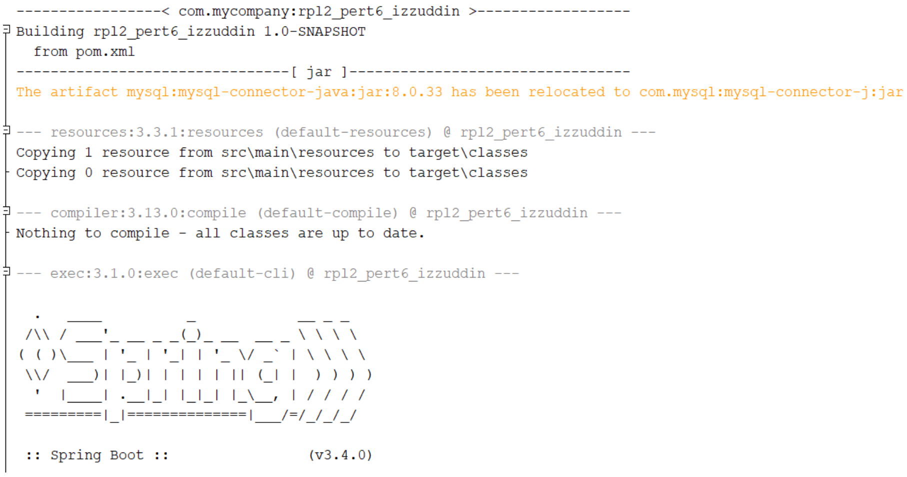
```java
private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String npm = npmField.getText();
    String nama = namaField.getText();
    int semester = Integer.parseInt( semesterField.getText());
    float ipk = Float.parseFloat(ipkField.getText());

    ModelMahasiswa mahasiswa = new ModelMahasiswa(0, npm, nama, semester, ipk);
    System.out.println(mahasiswa.getIpk());
    System.out.println(mahasiswa.getNama());
    System.out.println(mahasiswa.getNpm());
    System.out.println(mahasiswa.getSemester());

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

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

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

private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    JTextField idField = new JTextField(5);

// 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.deleteMahasiswa(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);
    }
}
```

- Pom.xml

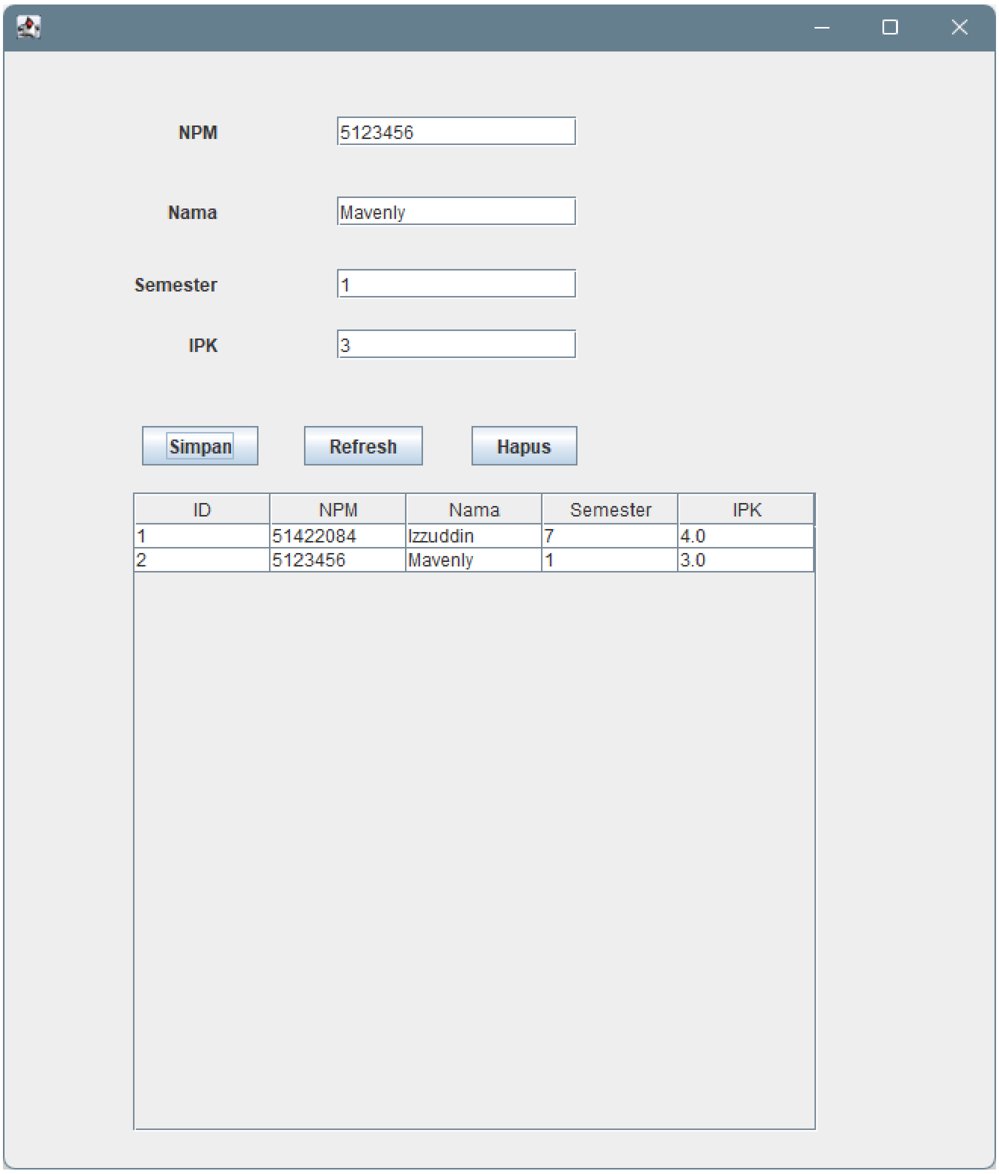


```xml
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.release>24</maven.compiler.release>
        <exec.mainClass>com.mycompany.rpl2_pert6_izzuddin.MahasiswaApp</exec.mainClass>
    </properties>
    <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.4.0</version>
    <relativePath/>
</parent>

<dependencies>
    <!-- Hibernate + Spring Data JPA -->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>

    <!-- MySQL Connector -->
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.33</version>
    </dependency>

    <!-- Spring Boot Web dependency (for MVC if needed) -->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <!-- Testing dependencies -->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>
</project>
```

Output:



```
-----------------< com.mycompany:rpl2_pert6_izzuddin >------------------
Building rpl2_pert6_izzuddin 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

--- resources:3.3.1:resources (default-resources) @ rpl2_pert6_izzuddin ---
Copying 1 resource from src\main\resources to target\classes
Copying 0 resource from src\main\resources to target\classes

--- compiler:3.13.0:compile (default-compile) @ rpl2_pert6_izzuddin ---
Nothing to compile - all classes are up to date.

--- exec:3.1.0:exec (default-cli) @ rpl2_pert6_izzuddin ---

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/

 :: Spring Boot ::                (v3.4.0)
```



NPM: 5123456

Nama: Mavenly

Semester: 1

IPK: 3

ID	NPM	Nama	Semester	IPK
1	51422084	Izzuddin	7	4.0
2	5123456	Mavenly	1	3.0