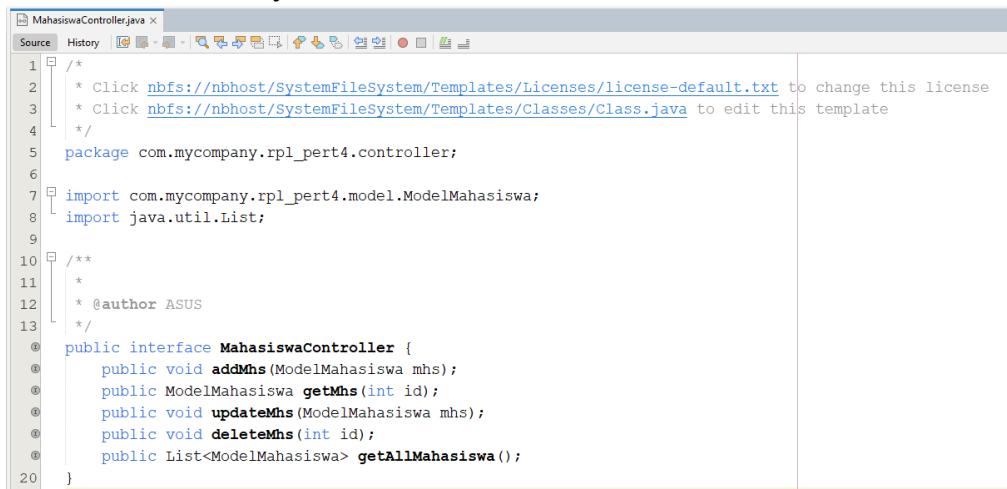


Nama : Muhammad Izzuddin Almansyur  
 NPM : 51422084  
 Kelas : 4IA07  
 Praktikum : Rekayasa Perangkat Lunak 2, ACT 4

---

## 1. Screenshot kode dan output program

### MahasiswaController.java

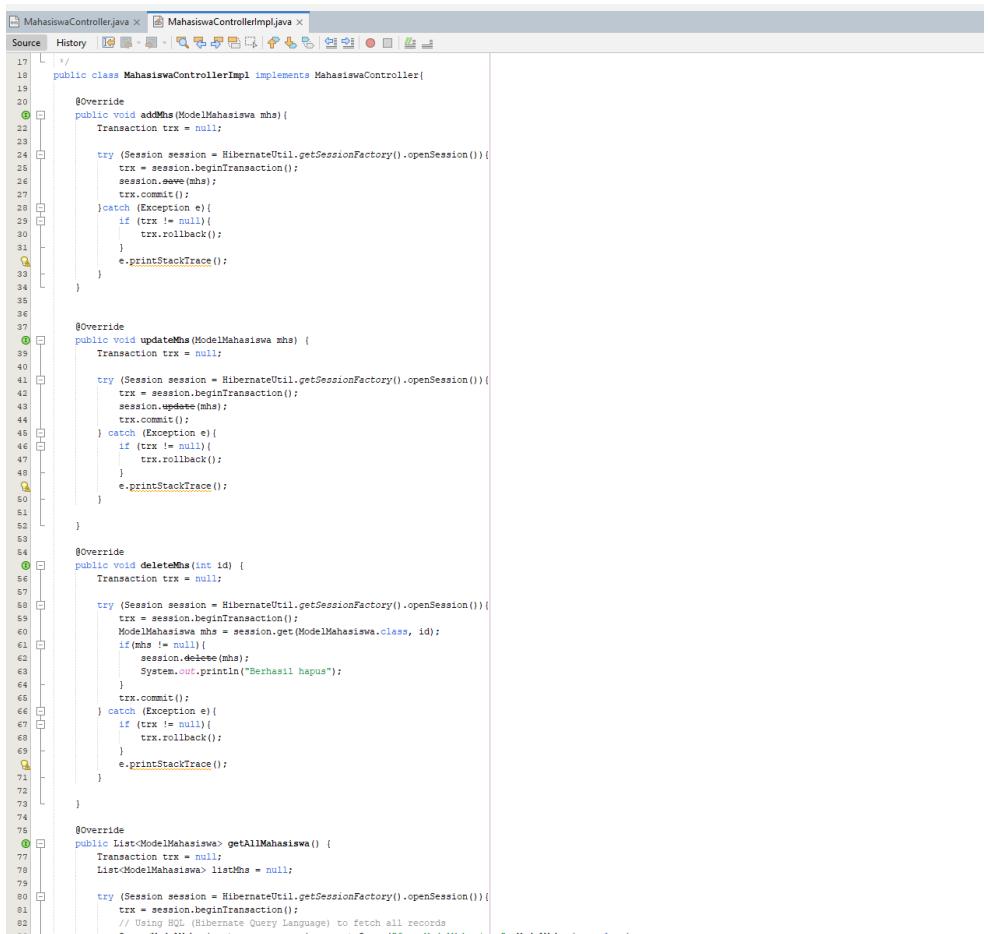


```

1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5  package com.mycompany.rpl_pert4.controller;
6
7  import com.mycompany.rpl_pert4.model.ModelMahasiswa;
8  import java.util.List;
9
10 /**
11  *
12  * @author ASUS
13  */
14 public interface MahasiswaController {
15     public void addMhs(ModelMahasiswa mhs);
16     public ModelMahasiswa getMhs(int id);
17     public void updateMhs(ModelMahasiswa mhs);
18     public void deleteMhs(int id);
19     public List<ModelMahasiswa> getAllMahasiswa();
20 }

```

### MahasiswaControllerImpl.java



```

17 /**
18  * 
19  */
20 public class MahasiswaControllerImpl implements MahasiswaController{
21
22     @Override
23     public void addMhs(ModelMahasiswa mhs) {
24         Transaction trx = null;
25
26         try (Session session = HibernateUtil.getSessionFactory().openSession()) {
27             trx = session.beginTransaction();
28             session.persist(mhs);
29             trx.commit();
30         } catch (Exception e) {
31             if (trx != null){
32                 trx.rollback();
33             }
34             e.printStackTrace();
35         }
36
37     @Override
38     public void updateMhs(ModelMahasiswa mhs) {
39         Transaction trx = null;
40
41         try (Session session = HibernateUtil.getSessionFactory().openSession()) {
42             trx = session.beginTransaction();
43             session.update(mhs);
44             trx.commit();
45         } catch (Exception e) {
46             if (trx != null){
47                 trx.rollback();
48             }
49             e.printStackTrace();
50         }
51     }
52
53     @Override
54     public void deleteMhs(int id) {
55         Transaction trx = null;
56
57         try (Session session = HibernateUtil.getSessionFactory().openSession()) {
58             trx = session.beginTransaction();
59             ModelMahasiswa mhs = session.get(ModelMahasiswa.class, id);
60             if(mhs != null){
61                 session.delete(mhs);
62                 System.out.println("Berhasil hapus");
63             }
64             trx.commit();
65         } catch (Exception e) {
66             if (trx != null){
67                 trx.rollback();
68             }
69             e.printStackTrace();
70         }
71     }
72
73     @Override
74     public List<ModelMahasiswa> getAllMahasiswa() {
75         Transaction trx = null;
76         List<ModelMahasiswa> listMhs = null;
77
78         try (Session session = HibernateUtil.getSessionFactory().openSession()) {
79             trx = session.beginTransaction();
80             // Using HQL (Hibernate Query Language) to fetch all records
81             String hql = "FROM ModelMahasiswa";
82             Query query = session.createQuery(hql);
83             listMhs = query.list();
84         } catch (Exception e) {
85             if (trx != null){
86                 trx.rollback();
87             }
88             e.printStackTrace();
89         }
90     }
91 }

```

## HibernateUtil.java

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package com.mycompany.rpl_pert4.model;
6
7  import org.hibernate.Session;
8  import org.hibernate.SessionFactory;
9  import org.hibernate.cfg.Configuration;
10
11 /**
12  *
13  * @author ASUS
14  */
15 public class HibernateUtil {
16     private static SessionFactory sessionFactory;
17
18     static {
19         try {
20             sessionFactory = new Configuration().configure().buildSessionFactory();
21         } catch(Throwable ex) {
22             System.err.println("Initial SessionFactory creation failed." + ex);
23             throw new ExceptionInInitializerError(ex);
24         }
25     }
26
27     public static SessionFactory getSessionFactory() {
28         return sessionFactory;
29     }
30
31     public static void testConnection() {
32         try (Session session = sessionFactory.openSession()) {
33             System.out.println("Connection to the database was successful");
34         } catch (Exception e) {
35             System.err.println("Failed to connect to database.");
36             e.printStackTrace();
37         }
38     }
39 }
```

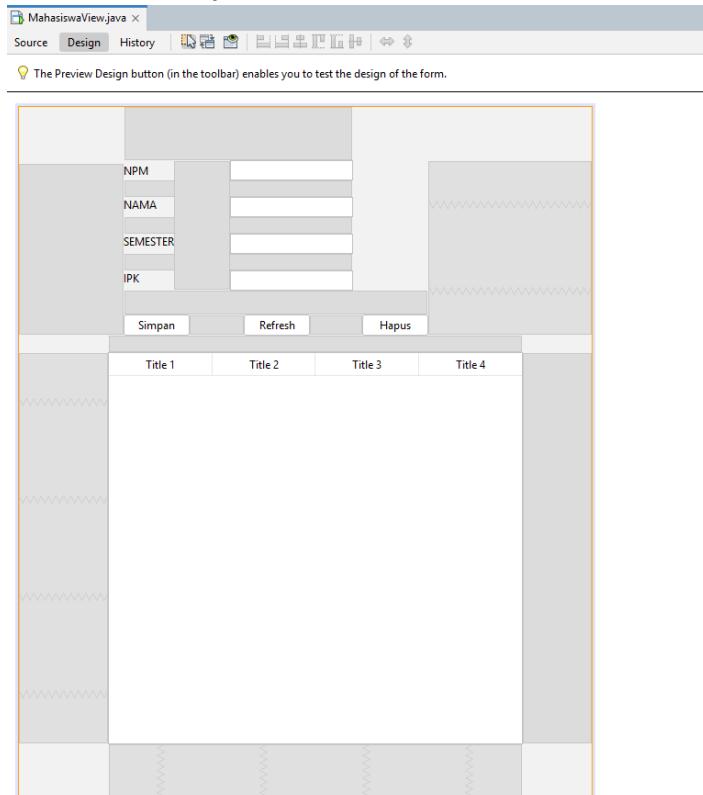
## ModelMahasiswa.java

```
6
7  import jakarta.persistence.*;
8
9 /**
10  *
11  * @author ASUS
12  */
13 @Entity
14 @Table(name= "mahasiswa")
15 public class ModelMahasiswa {
16
17     @Id
18     @GeneratedValue(strategy = GenerationType.IDENTITY)
19     @Column(name="id")
20     private int id;
21
22     @Column(name="npm", nullable=false, length=10)
23     private String npm;
24
25     @Column(name="nama", nullable=false, length=50)
26     private String nama;
27
28     @Column(name="semester", nullable=false)
29     private int semester;
30
31     @Column(name="ipk", nullable=false)
32     private float ipk;
33
34     public ModelMahasiswa() {
35
36     }
37
38
39     public ModelMahasiswa(int id, String npm, String nama, int semester, float ipk) {
40         this.id = id;
41         this.npm = npm;
42         this.nama = nama;
43         this.semester = semester;
44         this.ipk = ipk;
45     }
46     public int getId() {
47         return id;
48     }
49
50     public void setId(int id) {
51         this.id = id;
52     }
53
54     public String getNpm() {
55         return npm;
56     }
57 }
```

## ModelTabelMahasiswa.java

```
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);  
        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  
    }  
}
```

## MahasiswaView.java



## 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>com.mycompany</groupId>
    <artifactId>rpl_pert4</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>jar</packaging>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.release>22</maven.compiler.release>
        <exec.mainClass>com.mycompany.rpl_pert4.Rpl_pert4</exec.mainClass>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.hibernate.orm</groupId>
            <artifactId>hibernate-core</artifactId>
            <version>6.6.0.Final</version>
        </dependency>
        <!-- MySQL Connector -->
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.33</version>
        </dependency>
    </dependencies>
    <build>
        <resources>
            <resource>
                <directory>src/main/resources</directory>
                <filtering>false</filtering>
            </resource>
        </resources>
    </build>
</project>
```

## Output:

The screenshot shows a Java Swing application window titled "MahasiswaControllerImpl.java". The window contains four text input fields for NPM, NAMA, SEMESTER, and IPK, each with a corresponding label above it. Below these fields are three buttons: "Simpan", "Refresh", and "Hapus". At the bottom of the window is a table with columns labeled ID, NPM, Nama, Semester, and IPK. The table displays four rows of data.

ID	NPM	Nama	Semester	IPK
1	51422084	Izzuddin	7	3.85
2	51422123	Penaldo	12	3.9
3	51422122	Pessi	10	4.0
4	51422321	Jeta	6	2.5

NPM

NAMA

SEMESTER

IPK

ID	NPM	Nama	Semester	IPK
1	51422084	Izzuddin	7	3.85
2	51422123	Penaldo	12	3.9
3	51422122	Pessi	10	4.0
4	51422321	Jeta	6	2.5

**Sukses**

Data berhasil dihapus.

NPM

NAMA

SEMESTER

IPK

ID	NPM	Nama	Semester	IPK
1	51422084	Izzuddin	7	3.85
2	51422123	Penaldo	12	3.9
3	51422122	Pessi	10	4.0