

Nama : Muhammad Izzuddin Almansyur

NPM : 51422084

Kelas : 4IA07

Praktikum : Rekayasa Perangkat Lunak 2, ACT 4

1. Screenshoot 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  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
19  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
20  */
21 package com.mycompany.rpl_pert4.controller;
22
23 import com.mycompany.rpl_pert4.model.ModelMahasiswa;
24 import java.util.List;
25
26 /**
27  *
28  * @author ASUS
29  */
30 public class MahasiswaControllerImpl implements MahasiswaController {
31
32     @Override
33     public void addMhs (ModelMahasiswa mhs) {
34         Transaction trx = null;
35
36         try (Session session = HibernateUtil.getSessionFactory().openSession()) {
37             trx = session.beginTransaction();
38             session.save(mhs);
39             trx.commit();
40         } catch (Exception e) {
41             if (trx != null) {
42                 trx.rollback();
43             }
44             e.printStackTrace();
45         }
46     }
47
48     @Override
49     public void updateMhs (ModelMahasiswa mhs) {
50         Transaction trx = null;
51
52         try (Session session = HibernateUtil.getSessionFactory().openSession()) {
53             trx = session.beginTransaction();
54             session.update(mhs);
55             trx.commit();
56         } catch (Exception e) {
57             if (trx != null) {
58                 trx.rollback();
59             }
60             e.printStackTrace();
61         }
62     }
63
64     @Override
65     public void deleteMhs (int id) {
66         Transaction trx = null;
67
68         try (Session session = HibernateUtil.getSessionFactory().openSession()) {
69             trx = session.beginTransaction();
70             ModelMahasiswa mhs = session.get(ModelMahasiswa.class, id);
71             if (mhs != null) {
72                 session.delete(mhs);
73                 System.out.println("Berhasil hapus");
74             }
75             trx.commit();
76         } catch (Exception e) {
77             if (trx != null) {
78                 trx.rollback();
79             }
80             e.printStackTrace();
81         }
82     }
83
84     @Override
85     public List<ModelMahasiswa> getAllMahasiswa() {
86         Transaction trx = null;
87         List<ModelMahasiswa> listMhs = null;
88
89         try (Session session = HibernateUtil.getSessionFactory().openSession()) {
90             trx = session.beginTransaction();
91             // Using HQL (Hibernate Query Language) to fetch all records
92             Query<ModelMahasiswa> query = session.createQuery("FROM ModelMahasiswa");
93             listMhs = query.list();
94         } catch (Exception e) {
95             e.printStackTrace();
96         }
97     }
98 }
```

HibernateUtil.java

```
MahasiswaController.java x MahasiswaControllerImpl.java x HibernateUtil.java x
Source History

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

```
MahasiswaController.java x MahasiswaControllerImpl.java x HibernateUtil.java x ModelMahasiswa.java x
Source History

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
47      public int getId() {
48          return id;
49      }
50
51      public void setId(int id) {
52          this.id = id;
53      }
54
55      public String getNpm() {
56          return npm;
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.getName();
            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

MahasiswaView.java ×

Source Design History

The Preview Design button (in the toolbar) enables you to test the design of the form.

Title 1	Title 2	Title 3	Title 4
---------	---------	---------	---------

Pom.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <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">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.mycompany</groupId>
5   <artifactId>rpl_pert4</artifactId>
6   <version>1.0-SNAPSHOT</version>
7   <packaging>jar</packaging>
8   <properties>
9     <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
10    <maven.compiler.release>22</maven.compiler.release>
11    <exec.mainClass>com.mycompany.rpl_pert4.Rpl_pert4</exec.mainClass>
12  </properties>
13  <dependencies>
14    <dependency>
15      <groupId>org.hibernate.orm</groupId>
16      <artifactId>hibernate-core</artifactId>
17      <version>6.6.0.Final</version>
18    </dependency>
19
20    <!-- MySQL Connector -->
21    <dependency>
22      <groupId>mysql</groupId>
23      <artifactId>mysql-connector-java</artifactId>
24      <version>8.0.33</version>
25    </dependency>
26  </dependencies>
27  <build>
28    <resources>
29      <resource>
30        <directory>src/main/resources</directory>
31        <filtering>>false</filtering>
32      </resource>
33    </resources>
34  </build>
35</project>
```

Output:

NPM

51422321

NAMA

Jeta

SEMESTER

6

IPK

2.5

Simpan

Refresh

Hapus

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

51422321

NAMA

Jeta

SEMESTER

6

IPK

2.5

Simpan

Refresh

Hapus

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

i

Data berhasil dihapus.

OK

NPM

51422321

NAMA

Jeta

SEMESTER

6

IPK

2.5

Simpan

Refresh

Hapus

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