Nama : Dika Dwipati

NPM : 50421378

Kelas : 4IA28

ACT PERT 3

Cek koneksi

```
PILIH OPSI: 5
Koneksi ke db berhasil
Menu:
```

Tambah 3 data

```
Masukkan NPM:
50421378
Masukkan Nama:
dika
Masukkan Semester:
7
Masukkan IPK:
3
50421378dika73.0
Controller Data:   50421378dika73.0
com.mahasiswa.model.ModelMahasiswa@1990a65e
Mahasiswa berhasil ditambahkan!
Masukkan NPM:
5445179938
Masukkan Nama:
BCA
Masukkan Semester:
2
Masukkan IPK:
3
5445179938BCA23.0
Controller Data:   5445179938BCA23.0
com.mahasiswa.model.ModelMahasiswa@68ceda24
Mahasiswa berhasil ditambahkan!
```

```
Masukkan NPM:
57412533
Masukkan Nama:
auuu
Masukkan Semester:
4
Masukkan IPK:
3
57412533auuu43.0
Controller Data:   57412533auuu43.0
com.mahasiswa.model.ModelMahasiswa@281e3708
Mahasiswa berhasil ditambahkan!
Menu:
```

Update

```
0. Keluar
PILIH OPSI: 3
Masukkan ID mahasiswa: 2
Masukkan NPM:
5445179938
Masukkan Nama:
kita
Masukkan Semester:
4
Masukkan IPK:
4
Mahasiswa berhasil diperbarui!
Menu:
```

PERT 4

SQL CONNECTOR

```
        <!-- 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>
```

MAHASISWA CONTROLLER

```
7   import com.mahasiswa.model.ModelMahasiswa;
8   import java.util.List;
9
10  /**
11   *
12   * @author DIKA DWIPATI
13   */
    public interface MahasiswaController {
        public void addMhs(ModelMahasiswa mhs);
        public ModelMahasiswa getMhs(int id);
        public void updateMhs(ModelMahasiswa mhs);
        public void deleteMhs(int id );
        public List<ModelMahasiswa> getAllMahasiswa();
```

MAHASISWA CONTORLLER IMPL

```java
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;

/**
 *
 * @author DIKA DWIPATI
 */
public class MahasiswaControllerImpl implements MahasiswaController {

    @Override
    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();
        }
    }
```

```
    @Override
    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();
        }

    }

    @Override
    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();
        }
```

```
    @Override
    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;
    }

    @Override
    public ModelMahasiswa getMhs(int id) {
        throw new UnsupportedOperationException("Not supported yet.");
    }
}
```

HIBERNATE UTIL

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

/**
 *
 * @author DIKA DWIPATI
 */
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();
        }
    }
```

MODEL MAHASISWA

```
*/
package com.mahasiswa.model;
import jakarta.persistence.*;

/**
 *
 * @author DIKA DWIPATI
 */
@Entity
@Table(name = "mahasiswa")
public class ModelMahasiswa {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @Column(name="npm",nullable = false, length = 8)
    private String npm;
    @Column(name="nama",nullable = false, length = 8)
    private String nama;
    @Column(name="semester")
    private int semester;
    @Column(name="ipk")
    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 ModelMahasiswa(){
```

MODEL TABEL MAHASISWA

```
package com.mahasiswa.model;

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

/**
 *
 * @author DIKA DWIPATI
 */
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
    }

}
```

MAHASISWA VIEW