

ACTIVITY PERTEMUAN 3

NAMA : Muhammad Rifqi Dwi Putra
NPM : 51422145
KELAS : 4IA08
MATERI : Konsep Model View Controller (MVC)
MATA PRAKTIKUM : Rekayasa Perangkat Lunak 2

(Screenshot langkah-langkah sesuai video pembelajaran dan jelaskan dengan ringkas)

Screenshot kode dan output nya

Jawaban :

Kode

- 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>rpl2_pertemuan3_51422145</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.rpl2_pertemuan3_51422145.Rpl2_pertemuan3_51422145</exec.mainClass>
    </properties>
    <dependencies>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.33</version>
        </dependency>
    </dependencies>
</project>
```

- 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.mycompany.rpl2_pertemuan3_51422145.controller;

import com.mycompany.rpl2_pertemuan3_51422145.model.MahasiswaDAO;
import com.mycompany.rpl2_pertemuan3_51422145.model.ModelMahasiswa;
import java.util.List;
```

```

/**
 *
 * @author mrifq
 */

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("=====");
            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");
        }
    }

    // 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();
    }

```

- 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.mycompany.rpl2_pertemuan3_51422145.model;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

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

    public MahasiswaDAO(){
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/pert3_db", "root", "");
        } catch (Exception e){
            e.printStackTrace();
        }
    }

    public boolean checkConnection(){
        try{
            if(connection != null && !connection.isClosed()){
                return true; //koneksi berhasil
            }
        } catch (SQLException e){
            e.printStackTrace();
        }
        return false;
    }

    public void addMahasiswa(ModelMahasiswa mahasiswa){
        String sql = "INSERT INTO mahasiswa (nama, npm, semester, ipk) VALUES (?, ?, ?, ?)";
        try{
            PreparedStatement pstmt = connection.prepareStatement(sql);
            pstmt.setString(1, mahasiswa.getNama());
            pstmt.setString(2, mahasiswa.getNpm());
            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();
            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.getNama());
            pstmt.setString(2, mahasiswa.getNpm());
            pstmt.setInt(3, mahasiswa.getSemester());
            pstmt.setFloat(4, mahasiswa.getIpk());
            pstmt.setInt(5, mahasiswa.getId());
            pstmt.executeUpdate();
        } catch(SQLException e){
            e.printStackTrace();
        }
    }

    public void deleteMahasiswa(int id){
        String sql = "DELETE FROM mahasiswa WHERE id = ?";
        try{
            PreparedStatement pstmt = connection.prepareStatement(sql);
            pstmt.setInt(1, id);
            pstmt.executeUpdate();
        } catch(SQLException e){
            e.printStackTrace();
        }
    }

    // Method untuk menutup koneksi database
    public void closeConnection() {
        try {
            if (connection != null) {
                connection.close();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```

- 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.mycompany.rpl2_pertemuan3_51422145.model;

/**
 *
 * @author mrifq
 */
public class ModelMahasiswa {
    private int id;
    private String npm;
    private String nama;
    private int semester;
    private float 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) {
        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;
    }

    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;
    }
}
```

- MahasiswaView.java

```
        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

1. Tampilkan Semua Mahasiswa

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

```
=====
ID          : 1
NPM         : 51422145
NAMA        : rifqi
SEMESTER    : 7
IPK         : 4.0
=====
```

2. Tambah Mahasiswa

Menu:

1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar

PILIH OPSI: 2

Masukkan NPM:

51422157

Masukkan Nama:

abay

Masukkan Semester:

7

Masukkan IPK:

4

51422157abay74.0

Controller Data: 51422157abay74.0

com.mycompany.rpl2_pertemuan3_51422145.model.ModelMahasiswa@133e16fd

Mahasiswa berhasil ditambahkan!

3. Update Mahasiswa

Menu:

1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar

PILIH OPSI: 3

Masukkan ID mahasiswa: 2

Masukkan NPM:

51422157

Masukkan Nama:

abay

Masukkan Semester:

7

Masukkan IPK:

4

Mahasiswa berhasil diperbarui!

4. Hapus Mahasiswa

Menu:

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

Masukkan ID Mahasiswa: 3

Mahasiswa Berhasil Dihapus!

Koneksi ke db berhasil

5. Cek koneksi database

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

6. Menu keluar

Menu:

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

Program selesai.