ACTVITY M3

Nama : Akmal Fauzan Octavian

Kelas :4IA14

NPM :50421093

Mata Praktikum : Rekayasa Perangkat Lunak

---

- Kode mahasiswa Model
  - Mahasiswa DAO

```
import java.sql.*;
import java.util.ArrayList;
import java.util.List;
/**
 *
 * @author Octa
 */
public class MahasiswaDAO {
    private Connection connection;

    public MahasiswaDAO() {
        try{
            Class.forName(className:"com.mysql.cj.jdbc.Driver");
            connection = DriverManager.getConnection(url:"jdbc:mysql://localhost:3306/mvc_5042109
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

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

```
    public void closeConnection() {
        try {
            if(connection != null) {
                connection.close();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

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

    public List<ModelMahasiswa> getAllMahasiswa(){
        List<ModelMahasiswa> mahasiswaList = new ArrayList<>();
        String sql = "SELECT * FROM mahasiswa";
        try{
```

```
public List<ModelMahasiswa> getAllMahasiswa(){
    List<ModelMahasiswa> mahasiswaList = new ArrayList<>();
    String sql = "SELECT * FROM mahasiswa";
    try{
        Statement stmt = connection.createStatement();
        ResultSet rs = stmt.executeQuery(string:sql);
        while(rs.next()){
            mahasiswaList.add(new ModelMahasiswa(
                    id: rs.getInt(string:"id"),
                    nama: rs.getString(string:"npm"),
                    npm:rs.getString(string:"nama"),
                    semester: rs.getInt(string:"semester"),
                    ipk:rs.getFloat(string:"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(string:sql);
        pstmt.setString(i: 1, string:mahasiswa.getNpm());
        pstmt.setString(i: 2, string:mahasiswa.getNama());
        pstmt.setInt(i: 3, il: mahasiswa.getSemester());
        pstmt.setFloat(i: 4, f: mahasiswa.getIpk());
```

```
    public void updateMahasiswa(ModelMahasiswa mahasiswa){
        String sql = "UPDATE mahasiswa SET npm = ?, nama = ?, semester = ?, ipk = ? WHERE id = ?
        try{
            PreparedStatement pstmt = connection.prepareStatement(string:sql);
            pstmt.setString(i: 1, string:mahasiswa.getNpm());
            pstmt.setString(i: 2, string:mahasiswa.getNama());
            pstmt.setInt(i: 3, il: mahasiswa.getSemester());
            pstmt.setFloat(i: 4, f: mahasiswa.getIpk());
            pstmt.setInt(i: 5, il: mahasiswa.getId());
        } catch(SQLException e) {
            e.printStackTrace();
        }
    }

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

- Model Mahasiswa

```
package me.akmal.mahasiswa.model;

/**
 *
 * @author Octa
 */
public class ModelMahasiswa {
    private int id;
    private String nama;
    private String npm;
    private int semester;
    private float ipk;

    public ModelMahasiswa(int id, String nama, String npm, int semester, float ipk) {
        this.id = id;
        this.nama = nama;
        this.npm = npm;
        this.semester = semester;
        this.ipk = ipk;
    }

    public int getId() {
        return id;
    }
```

```
    public void setId(int id) {
        this.id = id;
    }

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public String getNpm() {
        return npm;
    }

    public void setNpm(String npm) {
        this.npm = npm;
    }

    public int getSemester() {
        return semester;
    }

    public void setSemester(int semester) {
        this.semester = semester;
    }
```

- Mahasiswa View

```
package me.akmal.mahasiswa.view;

import me.mahasiswa.controller.MahasiswaController;
import me.akmal.mahasiswa.model.MahasiswaDAO;
import java.util.Scanner;
/**
 *
 * @author Octa
 */
public class MahasiswaView {
    public static void main(String[] args) {
        MahasiswaDAO mahasiswaDAO = new MahasiswaDAO();
        MahasiswaController mahasiswaController = new MahasiswaController(mahasiswaDAO);

        Scanner scanner = new Scanner(source:System.in);
        int pilihan;

        while(true) {
            System.out.println(x: "Menu:");
            System.out.println(x: "1. Tampilkan Semua Mahasiswa");
            System.out.println(x: "2. Tambah Mahasiswa");
            System.out.println(x: "3. Update Mahasiswa");
            System.out.println(x: "4. Hapus Mahasiswa");
            System.out.println(x: "5. Cek Koneksi Database");
            System.out.println(x: "6. Keluar");
            System.out.println(x: "PILIH OPSI: ");
```

```
30                  System.out.println(x: "PILIH OPSI: ");
31                  pilihan = scanner.nextInt();
32                  scanner.nextLine();
33
34                  switch (pilihan){
35                      case 1:
36                          mahasiswaController.displayAllMahaiswa();
37                          break;
38
39                      case 2:
40                          // tambah mhs
41                          System.out.println(x: "Masukkan NPM: ");
42                          String npm = scanner.next();
43                          System.out.println(x: "Masukkan Nama: ");
44                          String nama = scanner.next();
45                          System.out.println(x: "Masukkan Semester: ");
46                          int semester = scanner.nextInt();
47                          System.out.println(x: "Masukkan IPK: ");
48                          float ipk = scanner.nextFloat();
49                          System.out.println(npm + nama + semester + ipk);
50
51                          mahasiswaController.addMahasiswa(npm, nama, semester, ipk);
52                          break;
53
54                      case 3:
55                          System.out.print(s: "Masukkan ID mahasiswa: ");
56                          int id = scanner.nextInt();
57                          scanner.nextLine();
58
```

```
62                          String namaBaru = scanner.next();
63                          System.out.println(x: "Masukkan Semester: ");
64                          int semesterBaru = scanner.nextInt();
65                          System.out.println(x: "Masukkan IPK: ");
66                          float ipkBaru = scanner.nextFloat();
67
68                          mahasiswaController.updateMahasiswa(id, npm:npmBaru, nama: namaBaru, semester: seme
69                          break;
70
71                      case 4:
72                          System.out.print(s: "Masukkan ID Mahasiswa: ");
73                          int idHapus = scanner.nextInt();
74                          mahasiswaController.deleteMahasiswa(id: idHapus);
75
76                      case 5:
77                          mahasiswaController.checkDatabaseConnection();
78                          break;
79
80                      case 6:
81                          // keluar
82                          mahasiswaController.closeConnection();
83                          System.out.println(x: "Program Selesai.");
84                          return;
85
86                      default:
87                          System.out.println(x: "Input Tidak Valid");
88                  }
89              }
90          }
```

- Mahasiswa Controller

```java
package me.mahasiswa.controller;

import me.akmal.mahasiswa.model.MahasiswaDAO;
import me.akmal.mahasiswa.model.ModelMahasiswa;
import java.util.List;
/**
 *
 * @author Octa
 */
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(x: "Tidak ada data mahasiswa");
        } else {
            System.out.println(x: "");
            System.out.println(x: "============================");
            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(x: "============================");
            }
        }
    }

    public void displayMessage(String message) {
        System.out.println(x: message);
    }

    public void checkDatabaseConnection() {
        boolean isConnected = mahasiswaDAO.checkConnection();
        if(isConnected){
            displayMessage(message: "Koneksi ke db berhasil");
        } else {
            displayMessage(message: "Koneksi DB Gagal");
        }
    }

    public void displayAllMahaiswa() {
        List<ModelMahasiswa> mahasiswaList = mahasiswaDAO.getAllMahasiswa();
        displayMahasiswaList(mahasiswaList);
    }

    public void addMahasiswa(String npm, String nama, int semester, float ipk) {
        ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(id: 0, nama: npm, npm:nama, semester, ipk);
        System.out.println("Controller Data:    " + npm + nama + semester + ipk);
        System.out.println(x: mahasiswaBaru);
        mahasiswaDAO.addMahasiswa(mahasiswa:mahasiswaBaru);
        displayMessage(message: "Mahasiswa berhasil ditambahkan!");
```

```
52              List<ModelMahasiswa> mahasiswaList = mahasiswaDAO.getAllMahasiswa();
53              displayMahasiswaList(mahasiswaList);
54          }
55
56          public void addMahasiswa(String npm, String nama, int semester, float ipk) {
57              ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(id: 0, nama: npm, npm: nama, semester, ipk);
58              System.out.println("Controller Data:     " + npm + nama + semester + ipk);
59              System.out.println(x: mahasiswaBaru);
60              mahasiswaDAO.addMahasiswa(mahasiswa:mahasiswaBaru);
61              displayMessage(message: "Mahasiswa berhasil ditambahkan!");
62          }
63
64          public void updateMahasiswa(int id, String npm, String nama, int semester, float ipk){
65              ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(id, nama: npm, npm: nama, semester, ipk);
66              mahasiswaDAO.updateMahasiswa(mahasiswa:mahasiswaBaru);
67              displayMessage(message: "Mahasiswaberhasil diperbarui!");
68          }
69
70          public void deleteMahasiswa(int id){
71              mahasiswaDAO.deleteMahasiswa(id);
72              displayMessage(message: "Mahasiswa Berhasil Dihapus!");
73          }
74
75          public void closeConnection() {
76              mahasiswaDAO.closeConnection();
77          }
78      }
79
```