

## ACTIVITY PERTEMUAN 3

NAMA : Muhammad Reza Rahman

NPM : 51421051

Kelas : 4IA14

Materi : MVC

Mata Praktikum : Rekayasa Perangkat Lunak 2

---

Input Kode:

- Model Mahasiswa

```
- package me.reza.mahasiswa.model;
-
- /**
-  *
-  * @author WINDOWS 10
-  */
- 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;
-     }
-
-     public float getIpk() {
-         return ipk;
-     }
-
-     public void setIpk(float ipk) {
-         this.ipk = ipk;
-     }
- }
-

```

- MahasiswaDAO

```

package me.reza.mahasiswa.model;

import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class MahasiswaDAO {
    private Connection connection;

    public MahasiswaDAO() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/mvc_51421051", "root",
""); // URL typo diperbaiki
        } 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(sql);
            pstmt.setString(1, mahasiswa.getNpm());
            pstmt.setString(2, mahasiswa.getNama());
            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("nama"),
                    rs.getString("npm"),

```

```

        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 = ?"; // Koma berlebih dihapus
    try {
        PreparedStatement pstmt = connection.prepareStatement(sql);
        pstmt.setString(1, mahasiswa.getNpm());
        pstmt.setString(2, mahasiswa.getNama());
        pstmt.setInt(3, mahasiswa.getSemester());
        pstmt.setFloat(4, mahasiswa.getIpk());
        pstmt.setInt(5, mahasiswa.getId()); // Parameter kelima dihapus
        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();
    }
}
}

```

- MahasiswaController

```
package me.reza.mahasiswa.controller;

import me.reza.mahasiswa.model.MahasiswaDAO;
import me.reza.mahasiswa.model.ModelMahasiswa;
import java.util.List;

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");
        }
    }

    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);
        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 deletMahasiswa(int id) {
        mahasiswaDAO.deleteMahasiswa(id);
        displayMessage("Mahasiswa Berhasil Dihapus");
    }

    public void closeConnection() {
        mahasiswaDAO.closeConnection();
    }
}

```

#### - MahasiswaView

```

package me.reza.mahasiswa.view;

import me.reza.mahasiswa.controller.MahasiswaController;
import me.reza.mahasiswa.model.MahasiswaDAO;
import java.util.Scanner;

public class MahasiswaView {

    public static void main(String[] args) {
        MahasiswaDAO mahasiswaDAO = new MahasiswaDAO();
        MahasiswaController mahasiswaController = new
MahasiswaController(mahasiswaDAO);

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

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

```

```

System.out.println("6. Keluar");
System.out.print("PILIH OPSI: ");

pilihan = scanner.nextInt();
scanner.nextLine();

switch (pilihan) {
    case 1 -> mahasiswaController.displayAllMahasiswa();
    case 2 -> {
        System.out.println("Masukkan NPM: ");
        String npm = scanner.next();
        System.out.println("Masukkan Nama: ");
        String nama = scanner.next();
        System.out.println("Masukkan Semester: ");
        int semester = scanner.nextInt();
        System.out.println("Masukkan IPK: ");
        float ipk = scanner.nextFloat();
        mahasiswaController.addMahasiswa(npm, nama, semester, ipk);
    }
    case 3 -> {
        System.out.print("Masukkan ID Mahasiswa: ");
        int id = scanner.nextInt();
        scanner.nextLine();
        System.out.println("Masukkan NPM: ");
        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);
    }
    case 4 -> {
        System.out.print("Masukkan ID Mahasiswa: ");
        int idHapus = scanner.nextInt();
        mahasiswaController.deletMahasiswa(idHapus);
        // Break ditambahkan
    }
    case 5 -> mahasiswaController.checkDatabaseConnection();
    // Break ditambahkan
    case 6 -> {
        mahasiswaController.closeConnection();
        System.out.println("Program selesai.");
        return;
    }
    default -> System.out.println("Pilihan tidak valid.");
}

```

```
}  
  }  
}
```



## Output Kode :

1

```
--- exec:3.1.0:exec (default-cli) @ Pert_51421051
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:
51421051
Masukkan Nama:
Reza
Masukkan Semester:
3
Masukkan IPK:
3.2
Mahasiswa berhasil ditambahkan!
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
```

2

```
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 : Reza
NAMA : 51421051
SEMESTER : 3
IPK : 3.2
----
```

3

```
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: 1
Masukkan NPM:
51421051
Masukkan Nama:
Reza
Masukkan Semester:
4
Masukkan IPK:
3.4
Mahasiswa berhasil diperbarui!
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 : Reza
NAMA : 51421051
SEMESTER : 4
IPK : 3.4
```

4

```
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: 1
Mahasiswa Berhasil Dihapus
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 1
Tidak ada data mahasiswa
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI:
```

5

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

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.

-----