ACTIVITY PERTEMUAN 3

Nama : Ferdy Agustian Prasetyo

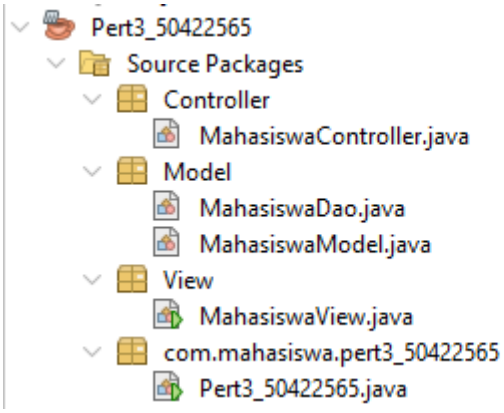
NPM 50422565

Kelas : 4IA28

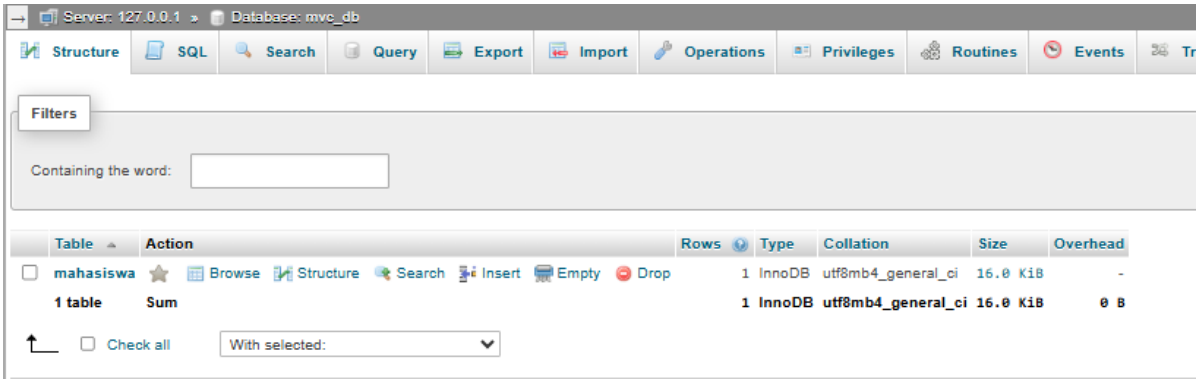
Materi : Model View Controller

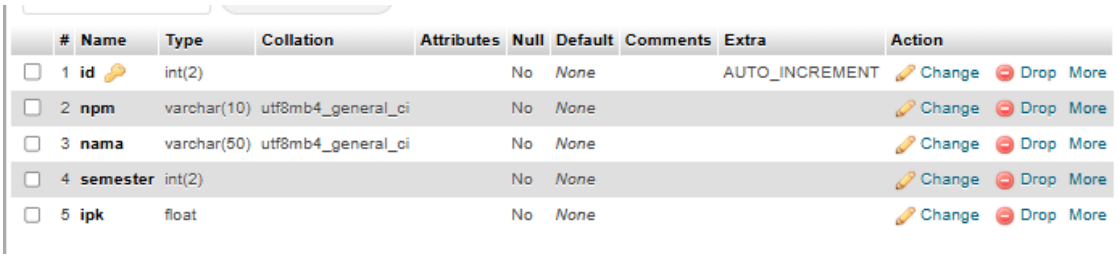
Mata Praktikum : Rekayasa Perangkat Lunak 2

Struktur Folder Program



Struktur Database





MahasiswaController.java

```java
package Controller;

import Model.MahasiswaDao;
import Model.MahasiswaModel;
import java.util.List;

/**
 *
 * @author ferdy
 */
public class MahasiswaController {
    private MahasiswaDao mahasiswaDao;

    public MahasiswaController(MahasiswaDao mahasiswaDAO) {
        this.mahasiswaDao = mahasiswaDAO;
    }

    public void checkDatabaseConnection() {
        boolean isConnected = mahasiswaDao.checkConnection();
        if (isConnected) {
            displayMessage("Koneksi ke db berhasil");
        } else {
            displayMessage("Koneksi DB Gagal");
        }
    }
public void displayAllMahasiswa() {
    List<MahasiswaModel> mahasiswaList = mahasiswaDao.getAllMahasiswa();
    displayMahasiswaList(mahasiswaList);
}

public void displayMahasiswaList(List<MahasiswaModel> mahasiswaList) {
    if (mahasiswaList.isEmpty()) {
        System.out.println("Tidak ada data mahasiswa");
    } else {
        System.out.println("--------------------------------");
        for (MahasiswaModel m : mahasiswaList) {
            System.out.println("ID \t: " + m.getId());
            System.out.println("NPM \t: " + m.getNpm());
            System.out.println("NAMA \t: " + m.getNama());
            System.out.println("SEMESTER \t: " + m.getSemester());
            System.out.println("IPK \t: " + m.getIpk());
            System.out.println("--------------------------------");
        }
    }
}

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

```
    public void addMahasiswa(String npm, String nama, int semester, float ipk) {
        MahasiswaModel mahasiswaBaru = new MahasiswaModel(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) {
        MahasiswaModel mahasiswaBaru = new MahasiswaModel(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

```
package Model;

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

/**
 *
 * @author Ferdy
 */
public class MahasiswaDao {

    private Connection connection;

    public MahasiswaDao() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            connection = DriverManager.getConnection("jdbc:mysql://localhost/mvc_db", "root", "");
        } 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 addMahasiswa(MahasiswaModel mahasiswa) {
    String sql = "INSERT INTO mahasiswa (npm, nama, semester, ipk) VALUES (?, ?, ?, ?)'
    try (Connection conn = this.connection;
            PreparedStatement pstmt = conn.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<MahasiswaModel> getAllMahasiswa() {
    List<MahasiswaModel> mahasiswaList = new ArrayList<>();
    String sql = "SELECT * FROM mahasiswa";

    try (Connection conn = this.connection;
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery(sql)) {

        while (rs.next()) {
            MahasiswaModel mhs = new MahasiswaModel(
                    rs.getInt("id"),
                    rs.getString("npm"),
                    rs.getString("nama"),
                    rs.getInt("semester"),
                    rs.getFloat("ipk"));
            mahasiswaList.add(mhs);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }

    return mahasiswaList;
}
public void updateMahasiswa(MahasiswaModel mahasiswa) {
    String sql = "UPDATE mahasiswa SET npm = ?, nama = ?, semester = ?, ipk = ? WHERE id = ?";

    try (Connection conn = this.connection;
            PreparedStatement pstmt = conn.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());

        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();
        }
    }

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

MahasiswaModel.java

```
package Model;

/**
 *
 * @author LENOVO LOQ
 */
public class MahasiswaModel {
    private int id;
    private String npm;
    private String nama;
    private int semester;
    private float ipk;

    public MahasiswaModel(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 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;
    }
}
```

MahasiswaView.java

```
package View;

import Controller.MahasiswaController;
import Model.MahasiswaDao;
import java.util.Scanner;

/**
 *
 * @author ferdy
 */
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("\nMenu:");
            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(); // membersihkan buffer
```

```
switch (pilihan) {
    case 1:
        mahasiswaController.displayAllMahasiswa();
        break;
    case 2:
        // tambah mhs
        System.out.print("Masukkan NPM: ");
        String npm = scanner.nextLine();
        System.out.print("Masukkan Nama: ");
        String nama = scanner.nextLine();
        System.out.print("Masukkan Semester: ");
        int semester = scanner.nextInt();

        // PERBAIKAN: Baca sebagai String untuk mengatasi InputMismatchException (koma vs titik)
        System.out.print("Masukkan IPK: ");
        String ipkString = scanner.next(); // Baca sebagai String
        float ipk = Float.parseFloat(ipkString.replace(',', '.')); // Ganti koma (,) jadi titik (.)

        scanner.nextLine(); // membersihkan buffer
        System.out.println(npm + nama + semester + ipk);
        mahasiswaController.addMahasiswa(npm, nama, semester, ipk);
        break;
```

```
    case 3:
        System.out.print("Masukkan ID mahasiswa: ");
        int id = scanner.nextInt();
        scanner.nextLine(); // membersihkan buffer
        System.out.print("Masukkan NPM: ");
        String npmBaru = scanner.nextLine();
        System.out.print("Masukkan Nama: ");
        String namaBaru = scanner.nextLine();
        System.out.print("Masukkan Semester: ");
        int semesterBaru = scanner.nextInt();

        System.out.print("Masukkan IPK: ");
        String ipkBaruString = scanner.next();
        float ipkBaru = Float.parseFloat(ipkBaruString.replace(',', '.'));

        scanner.nextLine();
        mahasiswaController.updateMahasiswa(id, npmBaru, namaBaru, semesterBaru, ipkBaru);
        break;
```

```
                case 4:
                    System.out.print("Masukkan ID Hapus: ");
                    int idHapus = scanner.nextInt();
                    scanner.nextLine(); // membersihkan buffer
                    mahasiswaController.deleteMahasiswa(idHapus);
                    break;
                case 5:
                    mahasiswaController.checkDatabaseConnection();
                    break;
                case 6:
                    // Keluar
                    mahasiswaController.closeConnection();
                    System.out.println("Program selesai.");
                    return;
                default:
                    System.out.println("Input Tidak valid");
            }
        }
    }
}
```

OUTPUT

Tambah Data 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: 50422565
Masukkan Nama: Ferdy agustian prastyo
Masukkan Semester: 7
Masukkan IPK: 3.8
50422565Ferdy agustian prastyo73.8
Controller Data: 50422565 Ferdy agustian prastyo 7 3.8
Mahasiswa berhasil ditambahkan!
```

Tampilkan Semua data 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              : 2
NPM             : 50422565
NAMA    : Ferdy agustian prastyo
SEMESTER : 7
IPK             : 3.8
--------------------------------------------
ID              : 3
NPM             : 50422523
NAMA    : Rusdiawan Renaldi
SEMESTER : 12
IPK             : 2.8
--------------------------------------------
```

Hapus data 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 Hapus: 2
Mahasiswa berhasil dihapus!
```

Cek koneksi ke DB

```
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
```

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.
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  09:24 min
Finished at: 2025-11-01T16:00:20+07:00
------------------------------------------------------------------------
```