PRAKTIKUM PEMROGRAMAN II

Latihan Pertemuan 7

(Java Database Connectivity (JDBC))

Dosen Pengajar :

WANDA GUSDYA PURNAMA, ST., MT.

DISUSUN OLEH :

Diaz Alfiari Rachmad

223040024

TEKNIK INFORMATIKA

UNIVERSITAS

PASUNDAN BANDUNG

2024

1. Langkah – langkah Pengerjaan

1.1 Persiapan Database MySQL

- Membuat database baru dengan nama pp2_biodata dan tabel biodata untuk menyimpan data pengguna dengan atribut

```
CREATE DATABASE pp2_biodata;
USE pp2_biodata;

CREATE TABLE `biodata` (
  `id` VARCHAR(255) NOT NULL,
  `nama` VARCHAR(255) DEFAULT NULL,
  `alamat` VARCHAR(255) DEFAULT NULL,
  `telepon` VARCHAR(15) DEFAULT NULL,
  PRIMARY KEY (`id`)
);
```

1.2 Pembuatan Struktur Kode

Membuat Folder Biodata yang didalam nya terdapat Folder libs dan src

- Didalam Folder libs masukan file JDBC untuk MySQL yang telah di download
- Didalam src buat Folder dao, db, model, dan view

1.3 Pembuatan Kelas Biodata

Kelas Biodata dibuat di dalam package model untuk merepresentasikan data biodata dengan atribut id, nama, alamat, dan telepon.

Kode :

```java
package model;

public class Biodata {
    private String id;
    private String nama;
    private String alamat;
    private String telepon;

    public Biodata(String id, String nama, String alamat, String telepon) {
        this.id = id;
        this.nama = nama;
        this.alamat = alamat;
        this.telepon = telepon;
    }

    // Getter and Setter
    public String getId() {
        return id;
    }

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

    public String getNama() {
        return nama;
    }

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

    public String getAlamat() {
        return alamat;
    }

    public void setAlamat(String alamat) {
        this.alamat = alamat;
    }

    public String getTelepon() {
        return telepon;
    }

    public void setTelepon(String telepon) {
        this.telepon = telepon;
    }
}

```

1.4 Pembuatan Kelas MySqlConnection

Kelas MySqlConnection dibuat di dalam package db untuk menghubungkan aplikasi Java dengan MySQL.

Kode :

```java
package db;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class MySqlConnection {
    private static final String DB_URL = "jdbc:mysql://localhost:3306/pp2_biodata";
    private static final String DB_USER = "root";
    private static final String DB_PASS = "";

    private static MySqlConnection instance;

    public static MySqlConnection getInstance() {
        if (instance == null) {
            instance = new MySqlConnection();
        }
        return instance;
    }

    public static Connection getConnection() {
        Connection connection = null;
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            connection = DriverManager.getConnection(
                    DB_URL, DB_USER, DB_PASS);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return connection;
    }
}
```

1.5 Pembuatan Kelas BiodataDao

Kelas BiodataDao dibuat di dalam package dao untuk menangani operasi CRUD pada database.

Kode :

```java
package dao;

import db.MySqlConnection;
import model.Biodata;

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

public class BiodataDao {
    public void save(Biodata biodata) {
        String query = "INSERT INTO biodata (id, nama, alamat, telepon) VALUES (?, ?, ?, ?)";
        try (Connection connection = MySqlConnection.getConnection();
                PreparedStatement ps = connection.prepareStatement(query)) {
            ps.setString(1, biodata.getId());
            ps.setString(2, biodata.getNama());
            ps.setString(3, biodata.getAlamat());
            ps.setString(4, biodata.getTelepon());
            ps.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public void update(Biodata biodata) {
        String query = "UPDATE biodata SET nama = ?, alamat = ?, telepon = ? WHERE id = ?";
        try (Connection connection = MySqlConnection.getConnection();
                PreparedStatement ps = connection.prepareStatement(query)) {
            ps.setString(1, biodata.getNama());
            ps.setString(2, biodata.getAlamat());
            ps.setString(3, biodata.getTelepon());
            ps.setString(4, biodata.getId());
            ps.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public void delete(String id) {
        String query = "DELETE FROM biodata WHERE id = ?";
        try (Connection connection = MySqlConnection.getConnection();
                PreparedStatement ps = connection.prepareStatement(query)) {
            ps.setString(1, id);
            ps.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public List<Biodata> getAll() {
        List<Biodata> list = new ArrayList<>();
        String query = "SELECT * FROM biodata";
        try (Connection connection = MySqlConnection.getConnection();
                Statement stmt = connection.createStatement();
                ResultSet rs = stmt.executeQuery(query)) {
            while (rs.next()) {
                Biodata biodata = new Biodata(
                        rs.getString("id"),
                        rs.getString("nama"),
                        rs.getString("alamat"),
                        rs.getString("telepon"));
                list.add(biodata);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return list;
    }
}

```

1.6 Pembuatan Kelas BiodataFrame

Kelas BiodataFrame dibuat di dalam package view untuk antarmuka pengguna, memungkinkan pengguna melakukan operasi CRUD.

Kode :

```java
package view;

import dao.BiodataDao;
import model.Biodata;

import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class BiodataFrame extends JFrame {
    private JTextField txtId;
    private JTextField txtNama;
    private JTextField txtAlamat;
    private JTextField txtTelepon;
    private JButton btnSave;
    private JButton btnUpdate;
    private JButton btnDelete;
    private BiodataDao dao;

    public BiodataFrame() {
        dao = new BiodataDao();

        txtId = new JTextField(20);
        txtNama = new JTextField(20);
        txtAlamat = new JTextField(20);
        txtTelepon = new JTextField(20);

        btnSave = new JButton("Save");
        btnUpdate = new JButton("Update");
        btnDelete = new JButton("Delete");

        btnSave.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Biodata biodata = new Biodata(txtId.getText(), txtNama.getText(), txtAlamat.getText(),
                        txtTelepon.getText());
                dao.save(biodata);
                loadData();
            }
        });

        btnUpdate.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Biodata biodata = new Biodata(txtId.getText(), txtNama.getText(), txtAlamat.getText(),
                        txtTelepon.getText());
                dao.update(biodata);
                loadData();
            }
        });

        btnDelete.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                dao.delete(txtId.getText());
                loadData();
            }
        });

        JPanel panel = new JPanel();
        panel.add(new JLabel("ID:"));
        panel.add(txtId);
        panel.add(new JLabel("Nama:"));
        panel.add(txtNama);
        panel.add(new JLabel("Alamat:"));
        panel.add(txtAlamat);
        panel.add(new JLabel("Telepon:"));
        panel.add(txtTelepon);
        panel.add(btnSave);
        panel.add(btnUpdate);
        panel.add(btnDelete);

        add(panel);
        setTitle("Aplikasi Biodata");
        setSize(280, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
    }

    private void loadData() {
        // Refresh the data in the UI (e.g., reload the JTable)
        JOptionPane.showMessageDialog(this, "Data berhasil dimuat ulang!");
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new BiodataFrame().setVisible(true));
    }
}

```

1.7 Kompilasi dan Eksekusi Program

Setelah kode selesai, lakukan kompilasi dan eksekusi dari Command Prompt:

- Arahkan Terminal ke direktori tempat file java berada :

  "D:\Kuliah\semester 5\Praktikum Pemograman II\Praktikum Pemograman II - 223040024\Latihan\Sesi7\Biodata"

- Lakukan kompilasi :

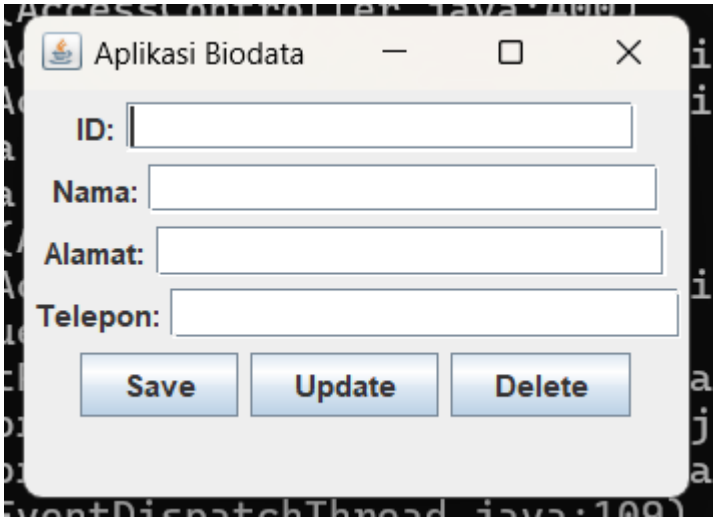
  javac -cp "libs/mysql-connector-j-9.1.0.jar" -d bin src\model\Biodata.java src\db\MySqlConnection.java src\dao\BiodataDao.java src\view\BiodataFrame.java

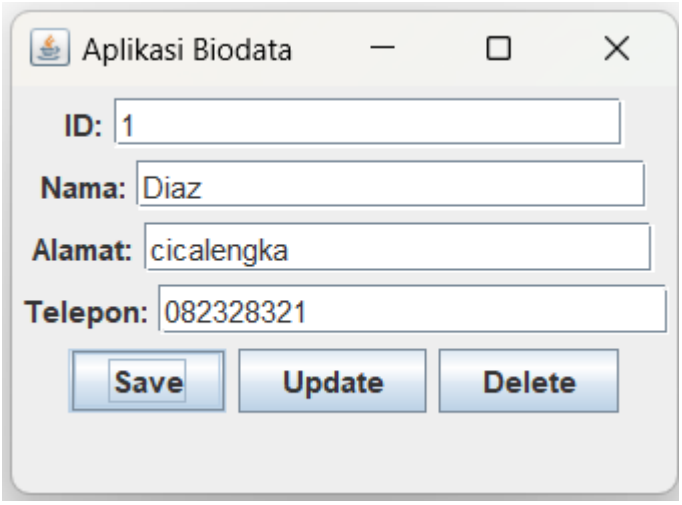
- Jalankan Program :

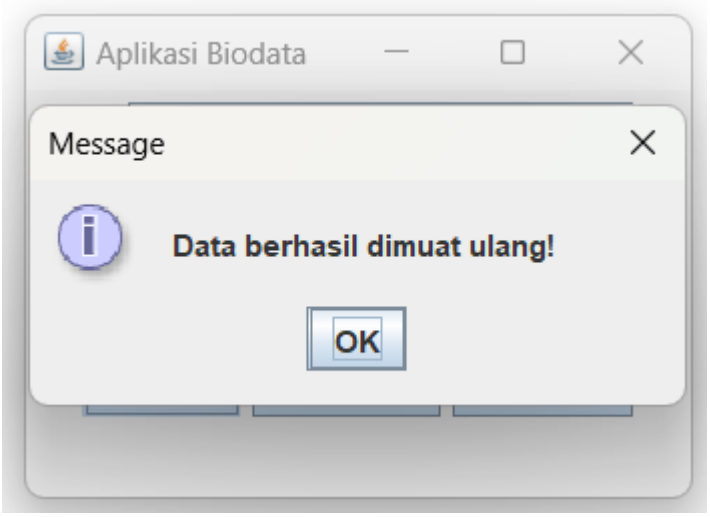
  java -cp "bin;libs/mysql-connector-j-9.1.0.jar" view.BiodataFrame

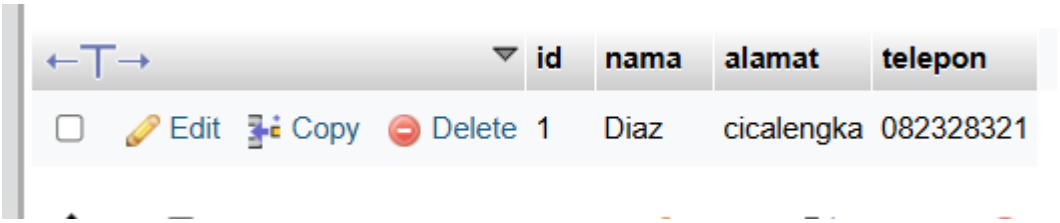
```
PS D:\> cd "D:\Kuliah\semester 5\Praktikum Pemograman II\Praktikum Pemograman II - 223040024\Latihan\Sesi7\Biodata"
PS D:\Kuliah\semester 5\Praktikum Pemograman II\Praktikum Pemograman II - 223040024\Latihan\Sesi7\Biodata> javac -cp "libs/mysql-connector-j-9.1.0.jar" -d b
in src\model\Biodata.java src\db\MySqlConnection.java src\dao\BiodataDao.java src\view\BiodataFrame.java
PS D:\Kuliah\semester 5\Praktikum Pemograman II\Praktikum Pemograman II - 223040024\Latihan\Sesi7\Biodata> java -cp "bin;libs/mysql-connector-j-9.1.0.jar" v
iew.BiodataFrame
```

2. Output




Aplikasi Biodata
ID: 1
Nama: Diaz
Alamat: cicalengka
Telepon: 082328321
Save
Update
Delete


Aplikasi Biodata
Message
Data berhasil dimuat ulang!
OK


id
nama
alamat
telepon
Edit
Copy
Delete
1
Diaz
cicalengka
082328321