LAPORAN PRAKTIKUM
PRAKTIK PEMROGRAMAN

OLEH: HELGA ARYA PRAYOGA (24051130022)

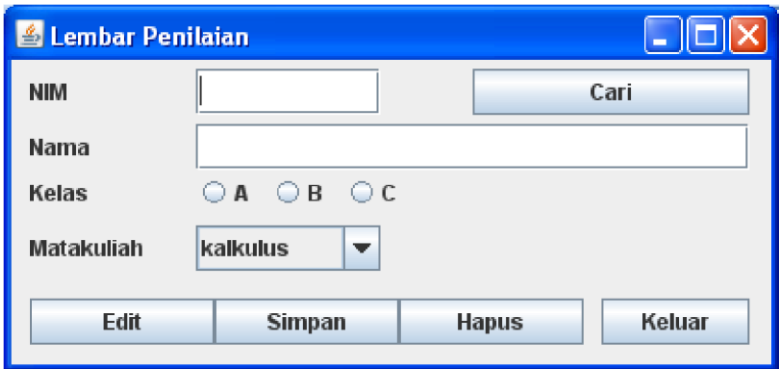
MODUL 11

TOPIK:
JAVA DAN DATABASE

TABLE OF CONTENTS

A. Penjelasan Tugas Praktikum

1. Penjelasan tentang database
2. Pengertian, tugas, serta kelebihan dari JDBC API
3. Perintah yang ada pada Data Manipulation Language (DML)
4. Tahapan menambahkan library MYSQL connector dalam program java
5. Buat program yang bisa menampilkan tampilan berikut ini, lengkap dengan Event handlingnya!



6. Buat database dan desain tabel yang sesuai dengan program diatas

B. Langkah-langkah dan Screenshot

1. Database merupakan suatu kumpulan data yang terstruktur dalam suatu sistem atau perangkat elektronik. Database digunakan agar dapat memudahkan dalam mencari informasi, menyimpan informasi, dan membuang informasi.
2. Java Database Connectivity (JDBC) merupakan sebuah API yang memungkinkan suatu program java dapat berinteraksi dengan database suatu sistem. JDBC API memungkinkan suatu program java untuk melakukan operasi database, seperti menambahkan data, mencari data, mengubah data, atau menghapus data.
   a. Tugas JDBC
      - Menghubungkan program java ke dalam database.
      - Memanajemen koneksi database.
      - Dapat mengeksekusi perintah SQL.
      - Memproses hasil query.
   b. Kelebihan JDBC API
      - JDBC bersifat independen terhadap database, artinya program java yang menggunakan JDBC dapat terhubung dengan seluruh database yang mendukung JDBC tanpa mengubah kode utama dari program java tersebut.

- JDBC dapat menghubungkan program java dengan berbagai sistem database.
- Dengan JDBC operasi database dapat dengan mudah dilakukan menggunakan perintah SQL dalam kode java.

3. Pada DML perintah yang sering dilakukan adalah insert (simpan) dan delete (hapus). Contoh dari perintah insert dan delete yaitu :
   a. Insert
      INSERT INTO nama_nim (nama, nim, kelas, matkul)
      VALUES (‘Helga Arya Prayoga', 24051130022, ‘A’, ‘Pemrograman’);
   b. Delete
      DELETE FROM nama_nim WHERE nim = 24051130022;
4. Tahapan menambahkan library MYSQL connector ke dalam program java
   a. Download MYSQL connector
   b. Buat proyek java dalam visual studio code
   c. Buat folder lib untuk menyimpan library eksternal
   d. Masukkan file connector .jar yang telah didownload ke dalam folder lib
   e. Tambahkan library ke dalam classpath proyek dengan cara membuka file settings.json kemudian masukkan kode :
      "java.project.referencedLibraries": [
          "lib/**/*.jar",
          "c:\\Users\\helga\\Downloads\\mysql-connector-j-8.0.31.jar"
      ],
   f. Buat kode untuk verifikasi koneksi ke database MYSQL
   g. Jalankan program
5. Program java menampilan data mahasiswa yang terhubung dalam database
   Kode Program:

```
import javax.swing.*;
import java.sql.*;

public class LembarPenilaian extends JFrame {
    private JTextField NIM, Nama;
    private JRadioButton btnA, btnB, btnC;
    private JComboBox<String> Matkul;
    private JButton btnCari, btnEdit, btnSimpan, btnHapus,
btnKeluar;
```

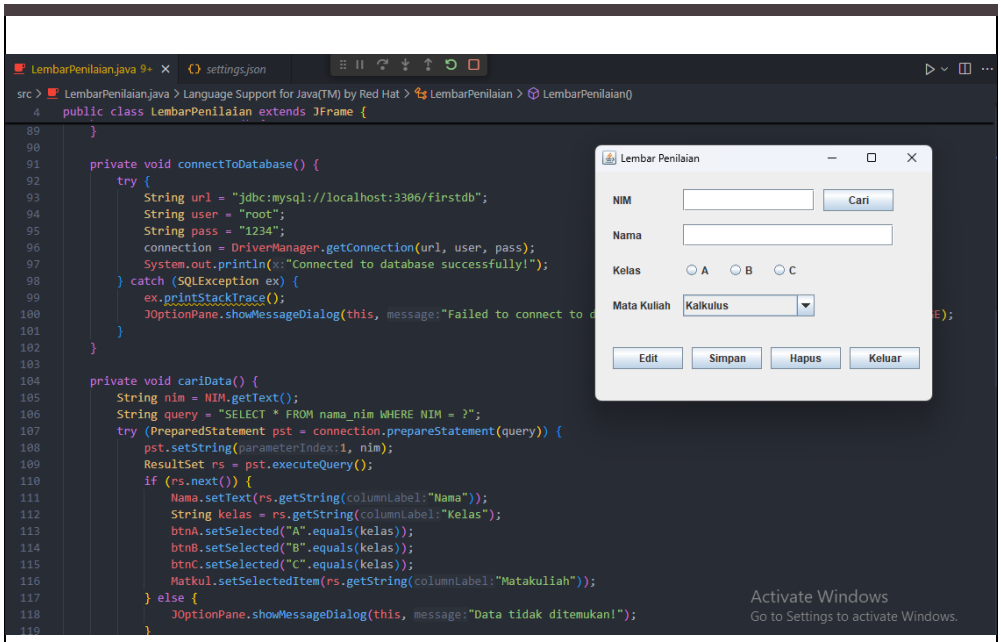
```
    private Connection connection;

    public LembarPenilaian() {
        setTitle("Lembar Penilaian");
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(null);

        JLabel lblNIM = new JLabel("NIM");
        lblNIM.setBounds(20, 20, 80, 25);
        add(lblNIM);

        NIM = new JTextField();
        NIM.setBounds(100, 20, 150, 25);
        add(NIM);

        btnCari = new JButton("Cari");
        btnCari.setBounds(260, 20, 80, 25);
        add(btnCari);

        JLabel lblNama = new JLabel("Nama");
        lblNama.setBounds(20, 60, 80, 25);
        add(lblNama);

        Nama = new JTextField();
        Nama.setBounds(100, 60, 240, 25);
        add(Nama);

        JLabel lblKelas = new JLabel("Kelas");
        lblKelas.setBounds(20, 100, 80, 25);
        add(lblKelas);

        btnA = new JRadioButton("A");
        btnA.setBounds(100, 100, 50, 25);
        add(btnA);

        btnB = new JRadioButton("B");
        btnB.setBounds(150, 100, 50, 25);
        add(btnB);

        btnC = new JRadioButton("C");
        btnC.setBounds(200, 100, 50, 25);
        add(btnC);

        ButtonGroup groupKelas = new ButtonGroup();
        groupKelas.add(btnA);
        groupKelas.add(btnB);
        groupKelas.add(btnC);

        JLabel lblMatakuliah = new JLabel("Mata Kuliah");
        lblMatakuliah.setBounds(20, 140, 80, 25);
        add(lblMatakuliah);

        Matkul = new JComboBox<>(new String[]{"Kalkulus",
"Fisika", "Pemrograman", "Agama", "Teknik Digital"});
```

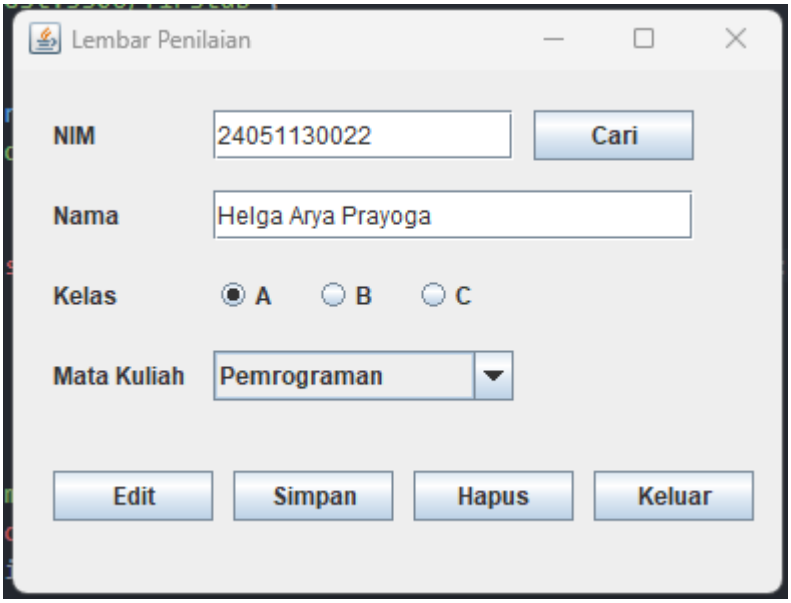
```
        Matkul.setBounds(100, 140, 150, 25);
        add(Matkul);

        btnEdit = new JButton("Edit");
        btnEdit.setBounds(20, 200, 80, 25);
        add(btnEdit);

        btnSimpan = new JButton("Simpan");
        btnSimpan.setBounds(110, 200, 80, 25);
        add(btnSimpan);

        btnHapus = new JButton("Hapus");
        btnHapus.setBounds(200, 200, 80, 25);
        add(btnHapus);

        btnKeluar = new JButton("Keluar");
        btnKeluar.setBounds(290, 200, 80, 25);
        add(btnKeluar);

        connectToDatabase();

        btnCari.addActionListener(e -> cariData());
        btnSimpan.addActionListener(e -> simpanData());
        btnEdit.addActionListener(e -> editData());
        btnHapus.addActionListener(e -> hapusData());
        btnKeluar.addActionListener(e -> System.exit(0));
    }

    private void connectToDatabase() {
        try {
            String url =
"jdbc:mysql://localhost:3306/firstdb";
            String user = "root";
            String pass = "1234";
            connection = DriverManager.getConnection(url,
user, pass);
            System.out.println("Connected to database
successfully!");
        } catch (SQLException ex) {
            ex.printStackTrace();
            JOptionPane.showMessageDialog(this, "Failed to
connect to database!", "Error", JOptionPane.ERROR_MESSAGE);
        }
    }

    private void cariData() {
        String nim = NIM.getText();
        String query = "SELECT * FROM nama_nim WHERE NIM =
?";
        try (PreparedStatement pst =
connection.prepareStatement(query)) {
            pst.setString(1, nim);
            ResultSet rs = pst.executeQuery();
            if (rs.next()) {
                Nama.setText(rs.getString("Nama"));
```

```
                String kelas = rs.getString("Kelas");
                btnA.setSelected("A".equals(kelas));
                btnB.setSelected("B".equals(kelas));
                btnC.setSelected("C".equals(kelas));

Matkul.setSelectedItem(rs.getString("Matakuliah"));
            } else {
                JOptionPane.showMessageDialog(this, "Data
tidak ditemukan!");
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }

    private void simpanData() {
        String nim = NIM.getText();
        String nama = Nama.getText();
        String kelas = btnA.isSelected() ? "A" :
btnB.isSelected() ? "B" : "C";
        String matakuliah = (String)
Matkul.getSelectedItem();

        String query = "INSERT INTO nama_nim (NIM, Nama,
Kelas, Matakuliah) VALUES (?, ?, ?, ?)";
        try (PreparedStatement pst =
connection.prepareStatement(query)) {
            pst.setString(1, nim);
            pst.setString(2, nama);
            pst.setString(3, kelas);
            pst.setString(4, matakuliah);
            pst.executeUpdate();
            JOptionPane.showMessageDialog(this, "Data
berhasil disimpan!");
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }

    private void editData() {
        String nim = NIM.getText();
        String nama = Nama.getText();
        String kelas = btnA.isSelected() ? "A" :
btnB.isSelected() ? "B" : "C";
        String matakuliah = (String)
Matkul.getSelectedItem();

        String query = "UPDATE nama_nim SET Nama = ?, Kelas =
?, Matakuliah = ? WHERE NIM = ?";
        try (PreparedStatement pst =
connection.prepareStatement(query)) {
            pst.setString(1, nama);
            pst.setString(2, kelas);
            pst.setString(3, matakuliah);
            pst.setString(4, nim);
```

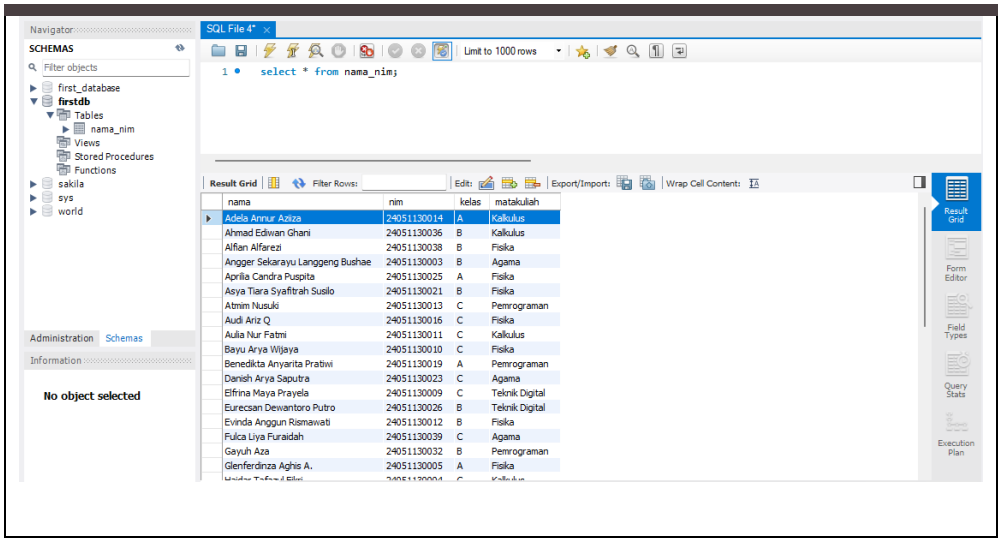
```
            pst.executeUpdate();
            JOptionPane.showMessageDialog(this, "Data
berhasil diperbarui!");
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }

    private void hapusData() {
        String nim = NIM.getText();
        String query = "DELETE FROM nama_nim WHERE NIM = ?";
        try (PreparedStatement pst =
connection.prepareStatement(query)) {
            pst.setString(1, nim);
            pst.executeUpdate();
            JOptionPane.showMessageDialog(this, "Data
berhasil dihapus!");
            clearFields();
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }

    private void clearFields() {
        NIM.setText("");
        Nama.setText("");
        btnA.setSelected(false);
        btnB.setSelected(false);
        btnC.setSelected(false);
        Matkul.setSelectedIndex(0);
    }

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

Screenshot:


LembarPenilaian.java 9+
settings.json
src > LembarPenilaian.java > Language Support for Java(TM) by Red Hat > LembarPenilaian > LembarPenilaian()
public class LembarPenilaian extends JFrame {
private void connectToDatabase() {
try {
String url = "jdbc:mysql://localhost:3306/firstdb";
String user = "root";
String pass = "1234";
connection = DriverManager.getConnection(url, user, pass);
System.out.println(x:"Connected to database successfully!");
} catch (SQLException ex) {
ex.printStackTrace();
JOptionPane.showMessageDialog(this, message:"Failed to connect to
}
}
private void cariData() {
String nim = NIM.getText();
String query = "SELECT * FROM nama_nim WHERE NIM = ?";
try (PreparedStatement pst = connection.prepareStatement(query)) {
pst.setString(parameterIndex:1, nim);
ResultSet rs = pst.executeQuery();
if (rs.next()) {
Nama.setText(rs.getString(columnLabel:"Nama"));
String kelas = rs.getString(columnLabel:"Kelas");
btnA.setSelected("A".equals(kelas));
btnB.setSelected("B".equals(kelas));
btnC.setSelected("C".equals(kelas));
Matkul.setSelectedItem(rs.getString(columnLabel:"Matakuliah"));
} else {
JOptionPane.showMessageDialog(this, message:"Data tidak ditemukan!");
Lembar Penilaian
NIM
Cari
Nama
Kelas
A
B
C
Mata Kuliah
Kalkulus
Edit
Simpan
Hapus
Keluar
Activate Windows
Go to Settings to activate Windows.


Lembar Penilaian
NIM
24051130022
Cari
Nama
Helga Arya Prayoga
Kelas
A
B
C
Mata Kuliah
Pemrograman
Edit
Simpan
Hapus
Keluar

6. Database dan desain tabel sesuai dengan program java diatas
   Screenshot:



C. Kendala yang Dialami

Kendala yang dialami adalah mengalami kesusahan disaat menginstal program MYSQL dan menghubungkan ke dalam visual studio code.

D. Kesimpulan

Praktikum ini memberikan pemahaman mengenai Java Database Connectivity (JDBC) API yang merupakan antarmuka penting untuk menghubungkan aplikasi Java dengan database. JDBC memudahkan dalam menjalankan perintah SQL dan mengelola data dalam database melalui aplikasi Java. Menambahkan MySQL Connector dalam proyek Java dilakukan melalui langkah-langkah seperti mengunduh file `.jar`, menambahkan ke dalam folder proyek, dan memperbarui classpath proyek. Hal ini memungkinkan aplikasi untuk terhubung ke database MySQL. Praktikum melibatkan pembuatan fitur untuk melakukan operasi CRUD (Create, Read, Update, Delete) pada data mahasiswa. Perintah INSERT, SELECT, UPDATE, dan DELETE digunakan untuk menambah, mencari, mengubah, dan menghapus data mahasiswa pada database.