PRAKTIKUM 10
PEMROGRAMAN DATABASE MYSQL DAN JAVA

Anggota Kelompok:

RIKI MAULANA

MUTIARA CAHAYA

RAHMA LAILATUL ZAHRA

ADRIAN DWI SYACHPUTRA

1.Database

Database adalah koleksi data yang terorganisir yang disimpan dalam cara yang memungkinkan pengambilan dan manipulasi yang efisien. Ini adalah komponen yang sangat penting dalam komputasi modern dan digunakan dalam berbagai aplikasi, dari situs web sederhana hingga sistem enterprise yang kompleks.

2. JDBC API

(Aplication Program Interface).

Pengertian JDBC API

JDBC API adalah sebuah API yang memungkinkan aplikasi Java untuk berinteraksi dengan database relasional. Tugas utama JDBC API adalah menyediakan koneksi ke database, menjalankan query SQL, mengelola transaksi, dan menghandle error.

Tugas JDBC API:

- Menghubungkan aplikasi Java dengan database.
- Mengirim dan mengeksekusi SQL query.
- Mengambil data dari database.
- Mengelola koneksi dan transaksi database.

Kelebihan JDBC API:

- Portabel dan mendukung berbagai jenis database.
- Mudah digunakan dengan akses real-time.
- Mendukung transaksi dan fleksibel dalam pemrosesan hasil query.

3. Perintah-perintah DML

(Data Manipulation Language)

01.
SELECT

Mengambil data dari database

Contoh: SELECT FROM karyawan;

02.
INSERT

Menambahkan data baru ke tabel.

Contoh: INSERT INTO karyawan (nama, usia) VALUES ('Andika', 20);

03.
UPDATE

Memperbarui data dalam tabel.

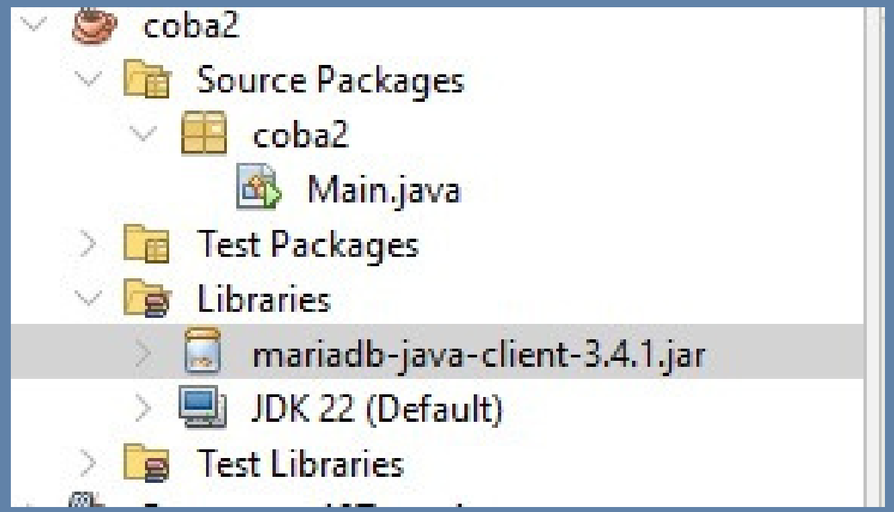
Contoh: UPDATE karyawan SET usia = 21 WHERE nama = 'Andika';

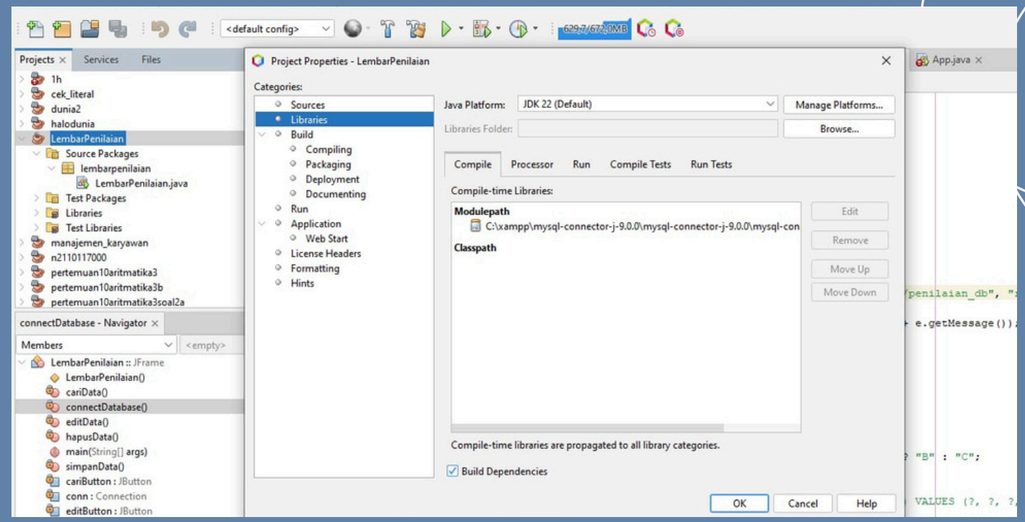
04.
DELETE

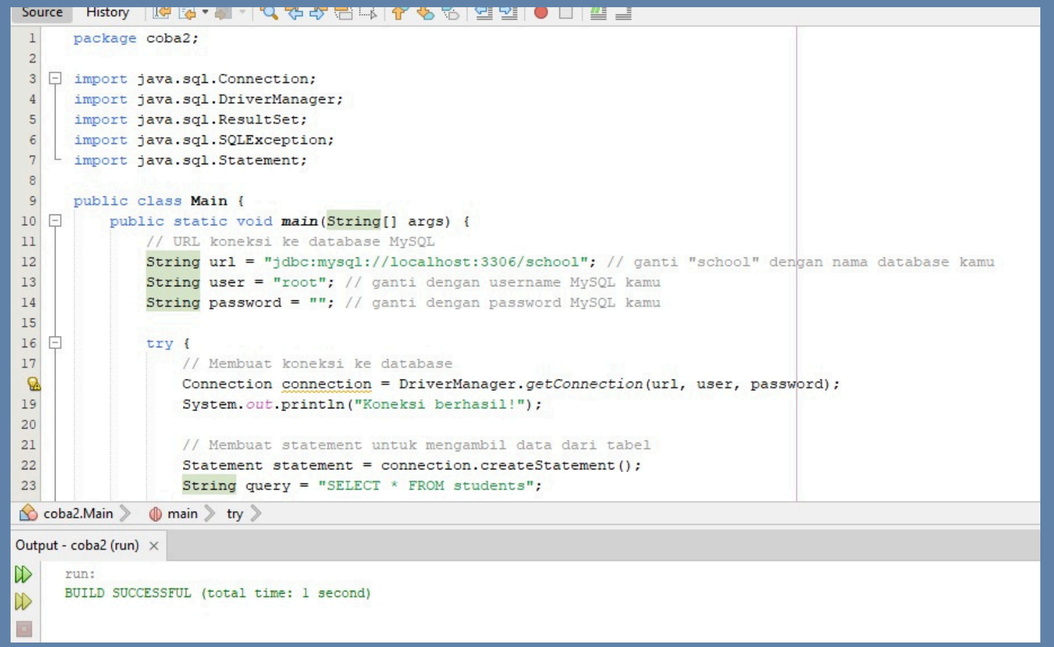
Menghapus data dari tabel.

Contoh: DELETE FROM karyawan WHERE nama = 'Andika';

4. Library My SQL







```java
package coba2;

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

public class Main {
    public static void main(String[] args) {
        // URL koneksi ke database MySQL
        String url = "jdbc:mysql://localhost:3306/school"; // ganti "school" dengan nama database kamu
        String user = "root"; // ganti dengan username MySQL kamu
        String password = ""; // ganti dengan password MySQL kamu

        try {
            // Membuat koneksi ke database
            Connection connection = DriverManager.getConnection(url, user, password);
            System.out.println("Koneksi berhasil!");

            // Membuat statement untuk mengambil data dari tabel
            Statement statement = connection.createStatement();
            String query = "SELECT * FROM students";
```

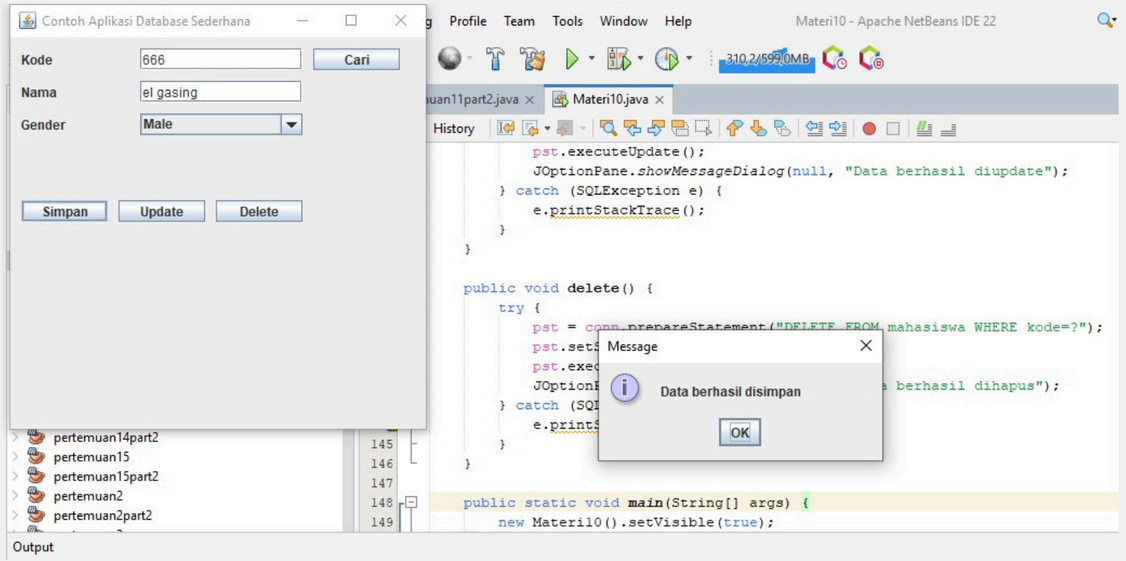
```
Output - coba2 (run)
run:
BUILD SUCCESSFUL (total time: 1 second)
```

4. Penjelasan

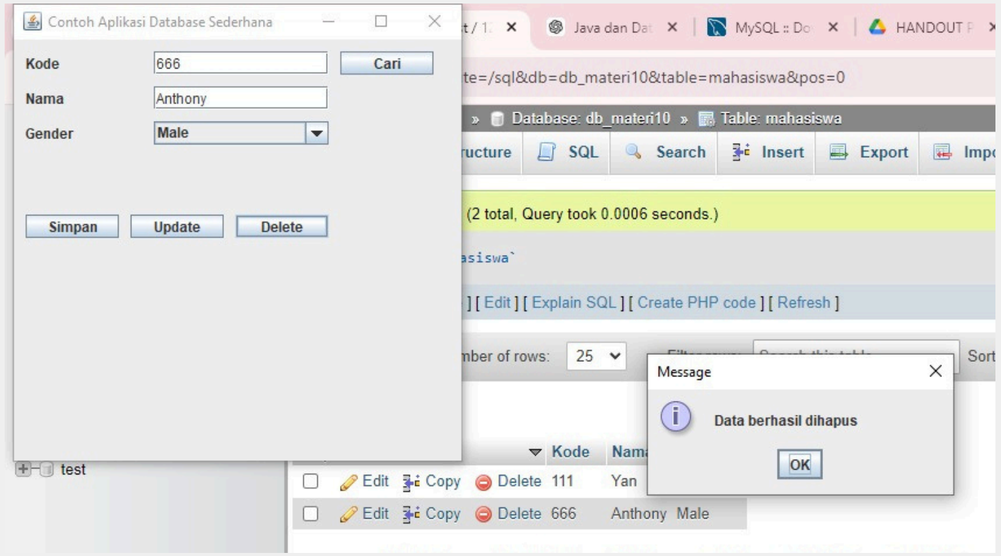
1. **Buka Project:**
   - Buka NetBeans, kemudian buka project `2110117000`.
2. **Tambahkan Library:**
   - Klik kanan pada folder `Libraries` di dalam project `2110117000`.
   - Pilih opsi "Add Library".
3. **Pilih Library MySQL Connector:**
   - Di jendela "Add Library", pilih "MySQL JDBC Driver" jika tersedia di daftar.
   - Jika belum ada, klik "Download Additional Libraries" untuk menambahkannya atau tambahkan secara manual dengan langkah berikutnya.
4. **Tambahkan MySQL Connector Secara Manual (Jika Belum Ada):**
   - Download MySQL Connector dari situs resmi MySQL (dalam format `.jar`).
   - Setelah download, klik kanan pada `Libraries` > "Add JAR/Folder...".
   - Pilih file `mysql-connector-java-X.X.X.jar` yang telah diunduh.
5. **Verifikasi:**
   - Pastikan library MySQL Connector sudah terlihat di bawah folder `Libraries` dalam project `2110117000`.
6. **Gunakan Library:**
   - Anda kini bisa menggunakan library ini di dalam kode Anda, misalnya dengan mengimpor `java.sql.*` dan membuat koneksi ke database MySQL.

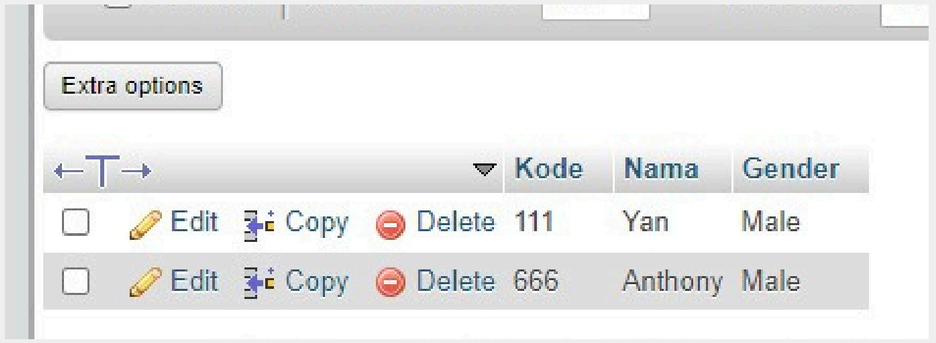
5.Database Sederhana

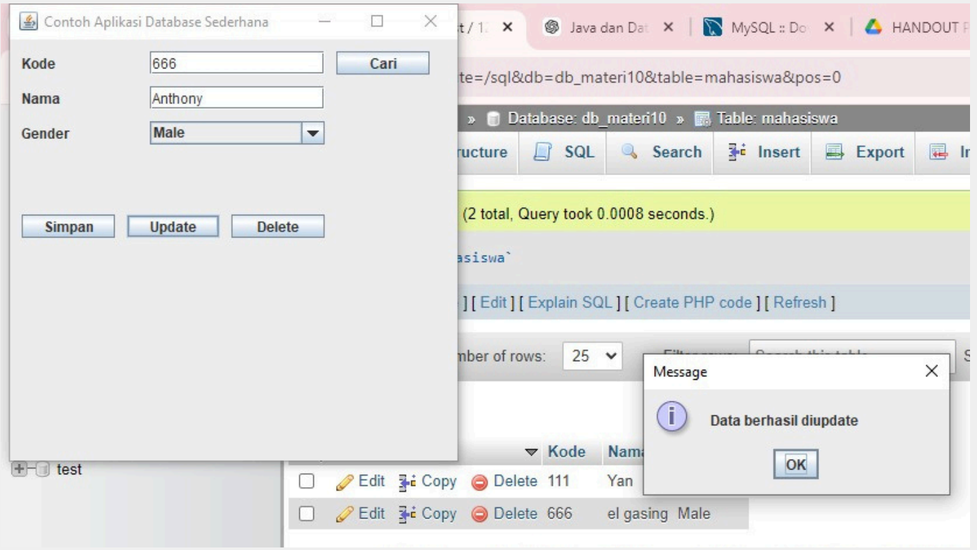
```
Source  History
1  package Materi10;
2
3  import javax.swing.*;
4  import java.awt.event.*;
5  import java.sql.*;
6
7  public class Materi10 extends JFrame implements ActionListener {
8      // GUI components
9      JTextField txtKode, txtNama;
10     JComboBox<String> comboGender;
11     JButton btnCari, btnSimpan, btnUpdate, btnDelete;
12
13     // Database connection
14     Connection conn;
15     PreparedStatement pst;
16
17     public Materi10() {
18         setTitle("Contoh Aplikasi Database Sederhana");
19         setSize(400, 400);
20         setDefaultCloseOperation(EXIT_ON_CLOSE);
21         setLayout(null);
22
23         // Kode
24         JLabel lblKode = new JLabel("Kode");
25         lblKode.setBounds(10, 10, 100, 20);
26         add(lblKode);
27         txtKode = new JTextField();
28         txtKode.setBounds(120, 10, 150, 20);
29         add(txtKode);
30
31         // Nama
32         JLabel lblNama = new JLabel("Nama");
```



5.Database Sederhana







5. Penjelasan

Tampilan Program

- **Text Field** untuk kode, nama, dan gender.
- **Button** untuk simpan, hapus, dan update.
- **Tabel** untuk menampilkan data.

Event Handling

- **Simpan**: Menyimpan data yang diinput ke database.
- **Hapus**: Menghapus data yang dipilih dari tabel berdasarkan kode.
- **Update**: Memperbarui data yang dipilih berdasarkan kode.
- **Klik Tabel**: Menampilkan data dari tabel ke form input untuk diedit.

Koneksi Database

Program terhubung ke database menggunakan **JDBC**, memungkinkan manipulasi data (CRUD).

Proses Simpan:

Mengirim data dari form ke database menggunakan query **INSERT**.

5. Penjelasan

Proses Hapus

Menghapus data dari database berdasarkan kode menggunakan query **DELETE.**

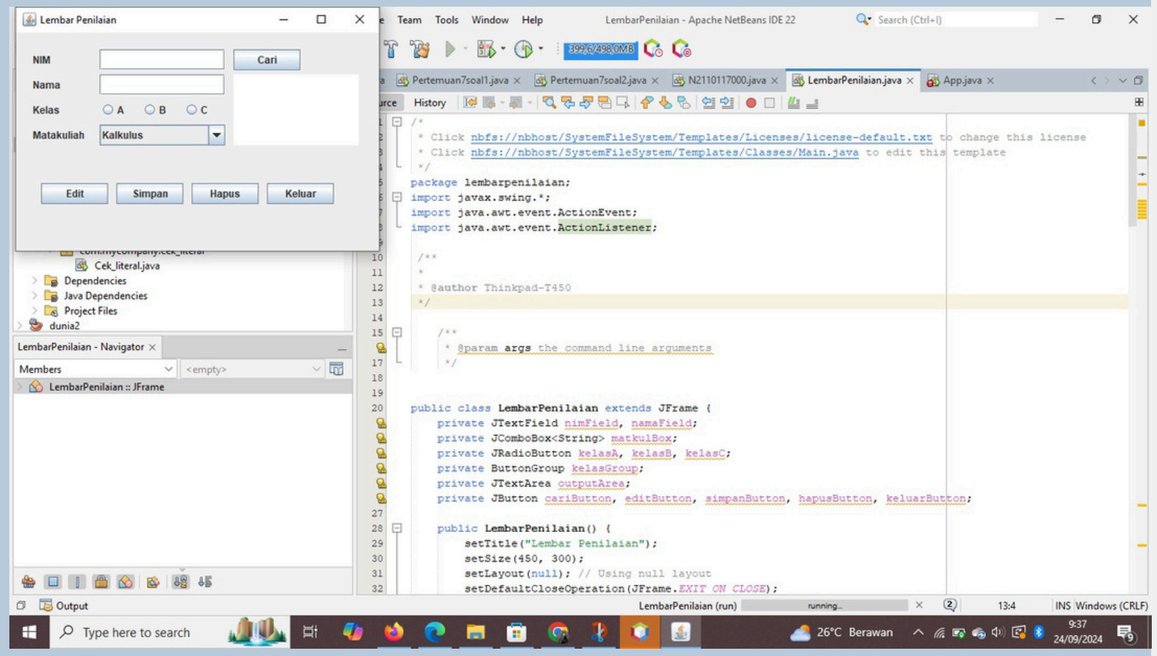
Proses Update

Memperbarui data di database berdasarkan kode menggunakan query **UPDATE.**

Struktur Tabel

- **kode:** Integer (10), primary key.
- **nama:** Varchar (50).
- **gender:** Varchar (1) (M/F).

Praktikum 1



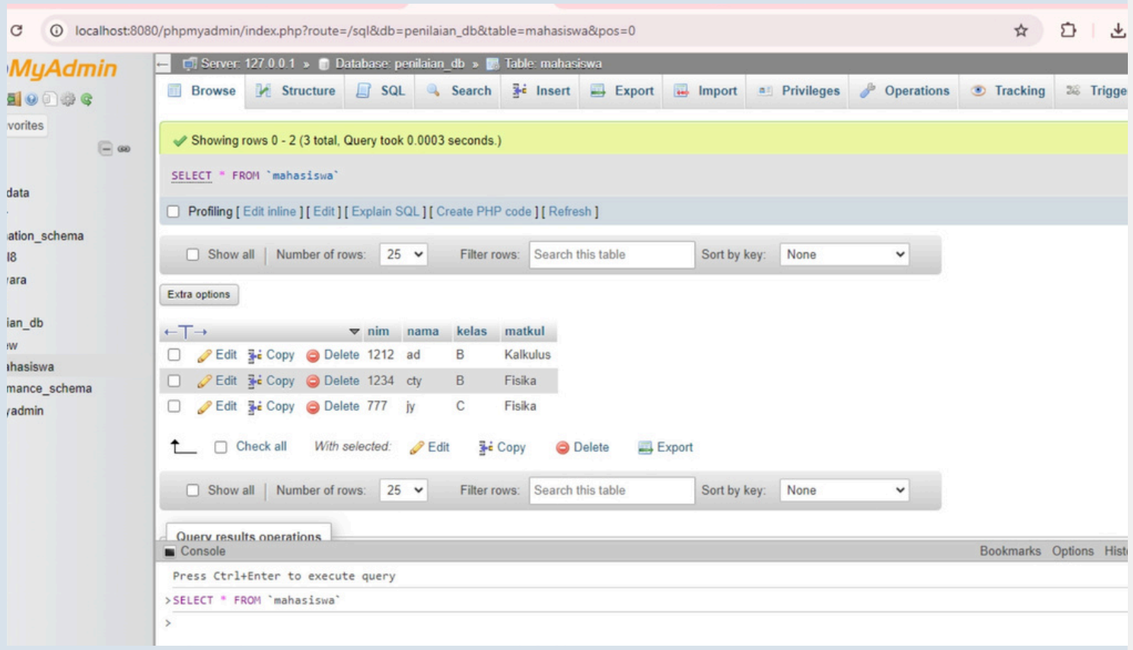

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */
package lembarpenilaian;
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

/**
 *
 * @author Thinkpad-T450
 */

    /**
     * @param args the command line arguments
     */


public class LembarPenilaian extends JFrame {
    private JTextField nimField, namaField;
    private JComboBox<String> matkulBox;
    private JRadioButton kelasA, kelasB, kelasC;
    private ButtonGroup kelasGroup;
    private JTextArea outputArea;
    private JButton cariButton, editButton, simpanButton, hapusButton, keluarButton;

    public LembarPenilaian() {
        setTitle("Lembar Penilaian");
        setSize(450, 300);
        setLayout(null); // Using null layout
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```



Penjelasan Program

Komponen Form:
- Input untuk **NIM**, **Nama**, **Kelas**, **Mata Kuliah**.
- Tombol: **Cari**, **Simpan**, **Hapus**, **Edit**, dan **Keluar**.

Koneksi Database:
Menggunakan **JDBC** untuk mengelola data dengan query **INSERT**, **DELETE**, dan **UPDATE**.

Event Handling:
- **Cari**: Mengambil data dari database berdasarkan **NIM**.
- **Simpan**: Menyimpan data input ke database.
- **Hapus**: Menghapus data berdasarkan **NIM**.
- **Edit**: Memperbarui data yang dipilih di database.
-K**eluar**: Menutup aplikasi.

Inti Program:
Form ini mempermudah pengelolaan data mahasiswa di database melalui pencarian, penyimpanan, penghapusan, dan pengeditan data.

Kesimpulan

Pada praktikum ini, kita telah belajar bagaimana mengintegrasikan database MySQL ke dalam aplikasi Java menggunakan JDBC. Selain itu, kita telah memahami dasar-dasar pengoperasian database, seperti penyimpanan, pengeditan, pencarian, dan penghapusan data. Kesimpulan yang dapat diambil adalah pentingnya penguasaan SQL serta keterampilan pemrograman event handling untuk memanipulasi data secara dinamis dalam aplikasi berbasis GUI.

Terima Kasih

Apakah ada pertanyaan?