

LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : Rekayasa Perangkat Lunak 2

Kelas : 4IA18

Praktikum ke- : 3

Tanggal : 01 November 2025

Materi : MVC

NPM : 50422215

Nama : Andi Purnama

Ketua Asisten : Haikal Abizar

Paraf Asisten : -

Nama Asisten : 1.

2.

3.

Jumlah Lembar : 4 Lembar



LABORATORIUM INFORMATIKA

UNIVERSITAS GUNADARMA

2025

1. Apa yang kalian ketahui tentang MVC?

MVC (Model-View-Controller) adalah pola arsitektur yang digunakan untuk memisahkan logika program menjadi tiga bagian utama agar lebih terstruktur dan mudah dikelola. Model bertanggung jawab untuk mengelola data serta logika bisnis, seperti penyimpanan, pengambilan, dan pengolahan data dari database. View berperan sebagai tampilan yang berinteraksi langsung dengan pengguna, menampilkan data dan menerima input. Sementara itu, Controller menjadi penghubung antara View dan Model; ia menerima input dari pengguna melalui View, memprosesnya menggunakan Model, lalu mengirimkan hasilnya kembali ke View untuk ditampilkan. Dengan pembagian ini, pengembangan aplikasi menjadi lebih teratur karena setiap bagian memiliki tanggung jawab yang jelas dan tidak saling bergantung secara langsung.

2. Jelaskan kode dan langkah-langkah program yang telah dibuat!

1. MahasiswaController.java

File ini berperan sebagai Controller dalam arsitektur MVC (Model-View-Controller). Tugas utamanya adalah menjadi penghubung antara tampilan (View) dan data (Model/DAO). MahasiswaController menerima input dari pengguna melalui MahasiswaView, kemudian memproses logika bisnis dan berinteraksi dengan MahasiswaDAO untuk melakukan operasi database. Di dalamnya terdapat fungsi-fungsi seperti menampilkan seluruh data mahasiswa (`displayAllMahasiswa`), menambah data (`addMahasiswa`), memperbarui data (`updateMahasiswa`), menghapus data (`deleteMahasiswa`), dan mengecek koneksi database (`checkDatabaseConnection`). Controller juga memiliki metode untuk menampilkan pesan dan daftar mahasiswa ke console. Jadi, seluruh alur logika aplikasi dan koordinasi antar komponen terjadi di sini.

2. MahasiswaDAO.java

File ini adalah lapisan akses data (Data Access Object / DAO) yang bertugas berinteraksi langsung dengan database MySQL. Di dalamnya, terdapat kode untuk membuat koneksi ke database (`DriverManager.getConnection`), menambah data (`addMahasiswa`), membaca seluruh data (`getAllMahasiswa`), memperbarui

(updateMahasiswa), menghapus (deleteMahasiswa), serta menutup koneksi (closeConnection). Kelas ini menggunakan JDBC (Java Database Connectivity) untuk menjalankan perintah SQL seperti INSERT, SELECT, UPDATE, dan DELETE. Singkatnya, MahasiswaDAO adalah jembatan antara Java dan database — seluruh aktivitas CRUD dilakukan di sini tanpa melibatkan tampilan atau logika bisnis.

3. ModelMahasiswa.java

File ini merupakan kelas model atau entitas data yang merepresentasikan satu objek mahasiswa. Ia berisi atribut seperti id, npm, nama, semester, dan ipk, beserta getter dan setter untuk mengakses dan memodifikasi data tersebut. ModelMahasiswa hanya berfungsi sebagai wadah atau template data yang digunakan oleh DAO dan Controller. Tidak ada logika di dalamnya, hanya definisi struktur data. Dengan adanya model ini, aplikasi dapat dengan mudah memindahkan data antar lapisan (misalnya dari database ke controller, lalu ke view) dalam bentuk objek.

4. MahasiswaView.java

File ini berfungsi sebagai View atau tampilan aplikasi, meskipun masih berbasis console (CLI), bukan GUI. Di dalamnya terdapat main() sebagai titik masuk program. View menampilkan menu pilihan kepada pengguna seperti menampilkan semua data, menambah, mengedit, menghapus, mengecek koneksi, dan keluar. Setiap pilihan akan memanggil metode yang sesuai di MahasiswaController. Dengan demikian, MahasiswaView hanya fokus pada interaksi pengguna (input/output) tanpa mengurus logika bisnis maupun koneksi database. Ia memanfaatkan Scanner untuk membaca input dari user.

5. pom.xml

File ini adalah file konfigurasi Maven, yaitu sistem manajemen proyek Java yang mengatur dependensi, build, dan metadata proyek. Dalam file ini, terdapat identitas proyek (groupId, artifactId, version), pengaturan kompilasi (maven.compiler.release), serta deklarasi dependensi penting yaitu mysql-connector-java versi 8.0.33, yang memungkinkan aplikasi terhubung ke database MySQL. pom.xml juga menentukan class utama (exec.mainClass) agar program

dapat dijalankan langsung lewat Maven. Singkatnya, file ini memastikan proyek dapat dikompilasi dan dijalankan dengan seluruh library yang dibutuhkan.

