

LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : Rekayasa Perangkat Lunak 2
Kelas : 4IA18
Praktikum ke- : 6
Tanggal : 22 November 2025
Materi : IMPLEMENTASI AOP DANN DEPENDENCY
INJECTION PADA PROJECT SPRING DAN HIBERNATE
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. Jelaskan kode dan langkah-langkah program yang telah dibuat!

Jawab:

1. File pom.xml

File pom.xml adalah pusat konfigurasi proyek Maven yang berfungsi menentukan semua dependensi yang diperlukan program. Di dalamnya terdapat identitas proyek seperti groupId, artifactId, dan version, serta deklarasi *parent* Spring Boot agar konfigurasi standar otomatis diikuti. File ini memuat dependensi penting seperti spring-boot-starter-data-jpa untuk mengelola ORM, mysql-connector-java untuk koneksi database MySQL, dan spring-boot-starter-web untuk kemungkinan penggunaan API. Selain itu, terdapat plugin spring-boot-maven-plugin yang menentukan class utama yang akan dijalankan yaitu MahasiswaApp. Dengan kata lain, file ini adalah fondasi yang memastikan aplikasi dapat berjalan lengkap dengan Spring Boot, JPA, dan MySQL.

2. File MahasiswaApp.java

File ini merupakan titik awal aplikasi dan berfungsi menjalankan Spring Boot sekaligus menampilkan tampilan GUI Swing. Ketika program dimulai, aplikasi mematikan mode headless agar Swing bisa muncul. Spring Boot kemudian melakukan proses inisialisasi dan membangun ApplicationContext yang berisi seluruh bean seperti controller dan service. Setelah konteks terbentuk, aplikasi mengambil bean MahasiswaController dari Spring dan menggunakannya untuk membuat instance MahasiswaView. Tampilan GUI selanjutnya ditampilkan kepada pengguna. File ini adalah penghubung antara backend Spring Boot dengan frontend Swing, sekaligus memastikan seluruh komponen aplikasi aktif saat program dijalankan.

3. File MahasiswaController.java

Controller berfungsi sebagai perantara antara GUI dan service. Ketika pengguna melakukan tindakan pada GUI seperti menambah atau menghapus data, request tersebut diberikan kepada controller. Class ini terdaftar sebagai bean Spring melalui anotasi @Controller, dan Spring menyuntikkan MahasiswaService melalui @Autowired. Controller menyediakan method untuk menambah, mengambil, memperbarui, menghapus, dan menampilkan semua data mahasiswa. Setiap

method meneruskan permintaan dari GUI ke service agar data dapat diproses. Dengan kata lain, controller bertanggung jawab mengelola alur permintaan dari tampilan menuju logika aplikasi.

4. File ModelMahasiswa.java

File ini adalah representasi objek mahasiswa yang dipetakan langsung ke tabel mahasiswa pada database MySQL melalui anotasi `@Entity` dan `@Table`. Setiap atribut seperti id, npm, nama, semester, dan ipk dihubungkan ke kolom database melalui anotasi `@Column`. Atribut id memiliki anotasi `@Id` dan `@GeneratedValue` agar nilainya dibuat otomatis oleh database. File ini juga menyediakan getter dan setter sebagai akses data. Keberadaan constructor kosong diperlukan oleh JPA, sedangkan constructor lengkap berguna untuk pemanggilan manual dalam kode. File ini memastikan bahwa data mahasiswa dapat diproses dan disimpan ke database sebagai objek terstruktur.

5. File ModelTableMahasiswa.java

File ini digunakan untuk menampilkan data mahasiswa ke dalam komponen JTable di GUI Swing. Dengan menurunkan `AbstractTableModel`, file ini menentukan struktur tabel seperti jumlah kolom, nama kolom, jumlah baris, dan nilai di setiap sel. Data mahasiswa disimpan dalam list, lalu ditampilkan ke tabel melalui method `getValueAt()`. Model ini juga menyediakan method `setMahasiswaList()` untuk memperbarui isi tabel jika data mengalami perubahan, diikuti pemanggilan `fireTableDataChanged()` agar tampilan diperbarui secara otomatis. File ini berfungsi sebagai jembatan antara daftar objek mahasiswa dan tampilan tabel pada GUI.

6. File MahasiswaRepository.java

Repository ini bertanggung jawab berkomunikasi langsung dengan database. Dengan memperluas `JpaRepository`, repository secara otomatis memiliki kemampuan lengkap untuk melakukan operasi CRUD tanpa penulisan SQL manual. Spring akan membuat implementasi repository secara otomatis dan menjadikannya bean melalui anotasi `@Repository`. Semua operasi seperti menyimpan data, menghapus berdasarkan ID, dan mengambil seluruh mahasiswa

dikelola oleh repository ini. File ini memastikan bahwa database dapat diakses dengan cara yang sederhana, cepat, dan efisien.

7. File MahasiswaService.java

Service menjadi lapisan logika bisnis yang mengatur alur sebelum data dikirim ke repository atau sebelum data dikembalikan ke controller. Class ini ditandai dengan anotasi `@Service` dan berisi method untuk menambah, mengambil, memperbarui, dan menghapus mahasiswa. Service menggunakan `MahasiswaRepository` yang disuntikkan secara otomatis oleh Spring. Dengan demikian, service menjadi lapisan yang memastikan controller tidak langsung berhubungan dengan database, melainkan melalui proses bisnis yang diatur dengan baik.

LANGKAH-LANGKAH ALUR PROGRAM (FLOW PROGRAM)

Ketika aplikasi dijalankan, Maven terlebih dahulu membaca file `pom.xml` untuk memuat seluruh dependensi sehingga Spring Boot, JPA, dan MySQL siap digunakan. Setelah itu, program memasuki `MahasiswaApp`, yang menjalankan Spring Boot dan membuat `ApplicationContext` berisi bean seperti controller, service, dan repository. Setelah inisialisasi selesai, GUI `MahasiswaView` ditampilkan menggunakan controller yang telah diambil dari konteks Spring. Saat pengguna menekan tombol seperti simpan atau hapus di GUI, perintah dikirim ke `MahasiswaController`. Controller lalu memanggil `MahasiswaService` untuk memproses logika bisnis. Service kemudian memanggil repository untuk melakukan operasi CRUD ke database MySQL melalui JPA. Setelah database memproses permintaan, hasilnya dikembalikan ke service, lalu ke controller, dan akhirnya ditampilkan kembali ke GUI melalui `ModelTableMahasiswa`. Proses ini berjalan setiap kali pengguna melakukan operasi, memastikan tampilan dan database selalu sinkron.