

LAPORAN TUGAS BESAR

PEMROGRAMAN BERORIENTASI OBJEK

Laporan ini disusun untuk memenuhi tugas akhir mata kuliah Pemrograman Berorientasi Objek
Yang dibimbing oleh Bapak Ir. Galih Wasis Wicaksono, S.Kom., M.Cs.



Disusun Oleh:

Jingga Maulidhina	202410370110045
Tegar Tutu Empar Pranata	202410370110008
Muhammad Sulaiman Rafiq	202410370110028

PRODI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH MALANG
2025

Sistem Arsitektur

1. Model

Dalam LibrarySystem, package model berfungsi sebagai representasi utama dari data perpustakaan. Sebagai entitas utama sistem, di dalamnya terdapat kelas Anggota, Buku, dan Pinjaman. Setiap kelas memiliki data yang dapat disimpan dan metode untuk mengakses atau mengubahnya, seperti getter, setter, dan konstruktor.

Controller akan membuat atau mengubah objek dari kelas-kelas ini untuk memproses logika sistem saat pengguna melakukan tindakan melalui antarmuka, seperti login, menambahkan buku, atau meminjam buku. Misalnya, sistem akan membuat objek "Peminjaman" berdasarkan objek "Anggota" dan "Buku" yang ada. Metode dalam kelas "Buku" akan memungkinkan status buku diperbarui langsung.

Dengan struktur ini, model berfungsi sebagai pengelola data dan logika dasar yang digunakan oleh pengontrol dan pemandangan. Ini memastikan bahwa semua data dalam sistem konsisten, terstruktur, dan siap untuk dikembangkan lebih lanjut, termasuk untuk diintegrasikan dengan database melalui pola DAO.

2. View

Dalam LibrarySystem, package view berfungsi sebagai antarmuka pengguna yang dibangun menggunakan JavaFX. Terdiri dari kelas-kelas seperti LoginView, DashboardView, AnggotaView, BukuView, PeminjamanView, dan LaporanView, masing-masing bertanggung jawab untuk menampilkan dan menangani input pengguna untuk komponen sistem tertentu.

Komponen JavaFX seperti TextField, Button, Label, dan TableView digunakan dalam setiap kelas tampilan untuk menerima input dan menampilkan data. Misalnya, LoginView menampilkan form login, sedangkan AnggotaView menampilkan tabel daftar anggota dan form untuk menambah atau mengubah data.

View berfungsi sebagai jembatan visual antara sistem dan pengguna. Namun, itu bersifat pasif, artinya tidak menjalankan logika bisnis secara langsung. Pemanggilan metode tertentu akan digunakan untuk mengirimkan semua tindakan pengguna, seperti mengklik tombol atau mengisi formulir ke controller. Secara keseluruhan, view memastikan interaksi pengguna yang mudah dan lancar sambil tetap membagi tanggung jawab antara tampilan, logika, dan data.

3. Controller

Dalam sistem LibrarySystem, package controller berfungsi sebagai pengatur logika aplikasi dan juga sebagai penghubung utama antara pemandangan dan model. Semua tindakan yang dilakukan pengguna melalui antarmuka, seperti menekan tombol atau mengisi formulir, dikirim ke controller untuk diproses.

Setiap controller melakukan tugas tertentu berdasarkan modulnya. AuthController bertanggung jawab atas proses autentikasi seperti login dan logout dengan mengambil data dari LoginView dan memverifikasinya dengan data yang tersedia. AnggotaController, BukuController, dan PeminjamanController bertanggung jawab atas alur logika yang digunakan untuk mengelola data anggota, buku, dan transaksi peminjaman. Mereka bertanggung jawab untuk memproses data, termasuk penambahan, perubahan, dan penghapusan, serta pengambilan data yang akan ditampilkan kembali di tampilan.

Selain mengatur alur data, controller juga bertanggung jawab atas proses validasi input sebelum diteruskan ke model. Proses ini memastikan bahwa hanya data yang valid dan sesuai dengan aturan yang disimpan atau diproses. Dengan cara ini, controller menjaga kestabilan alur aplikasi, memisahkan antara tampilan dan logika bisnis, dan memungkinkan sistem diuji dan dikembangkan secara modular.

4. Main

Package Main berisi kelas LibrarySystem, yang berfungsi sebagai titik awal (entry point) dari aplikasi secara keseluruhan. Kelas ini menjalankan metode main() dan metode start(), yang masing-masing akan menginisialisasi dan menampilkan tampilan awal aplikasi, LoginView.

Mengatur siklus hidup aplikasi JavaFX adalah fungsi utama LibrarySystem. Saat aplikasi dijalankan, Java akan memanggil method main(), yang kemudian mengeksekusi launch(args) untuk memulai Thread Aplikasi JavaFX. Dalam metode start (Stage primaryStage), LibrarySystem akan membuat instance LoginView dan menampilkannya sebagai antarmuka pertama yang dilihat pengguna.

Selain itu, LibrarySystem juga bertanggung jawab untuk menyambungkan dependensi awal antara komponen view dan controller. Misalnya, instance AuthController dapat memberikan objek LoginView untuk menyelesaikan proses login.

Dengan demikian, controller akan mengontrol alur navigasi berdasarkan tindakan pengguna, seperti berpindah ke DashboardView atau modul lainnya.

Dengan kata lain, package main berfungsi sebagai pintu gerbang ke seluruh sistem, memulai alur kerja aplikasi, menyusun tampilan pertama, dan menyiapkan jalur komunikasi antar komponen. Meskipun hanya terdiri dari satu kelas, peranannya sangat penting untuk memastikan struktur sistem berjalan lancar dan teratur sejak aplikasi dimulai.

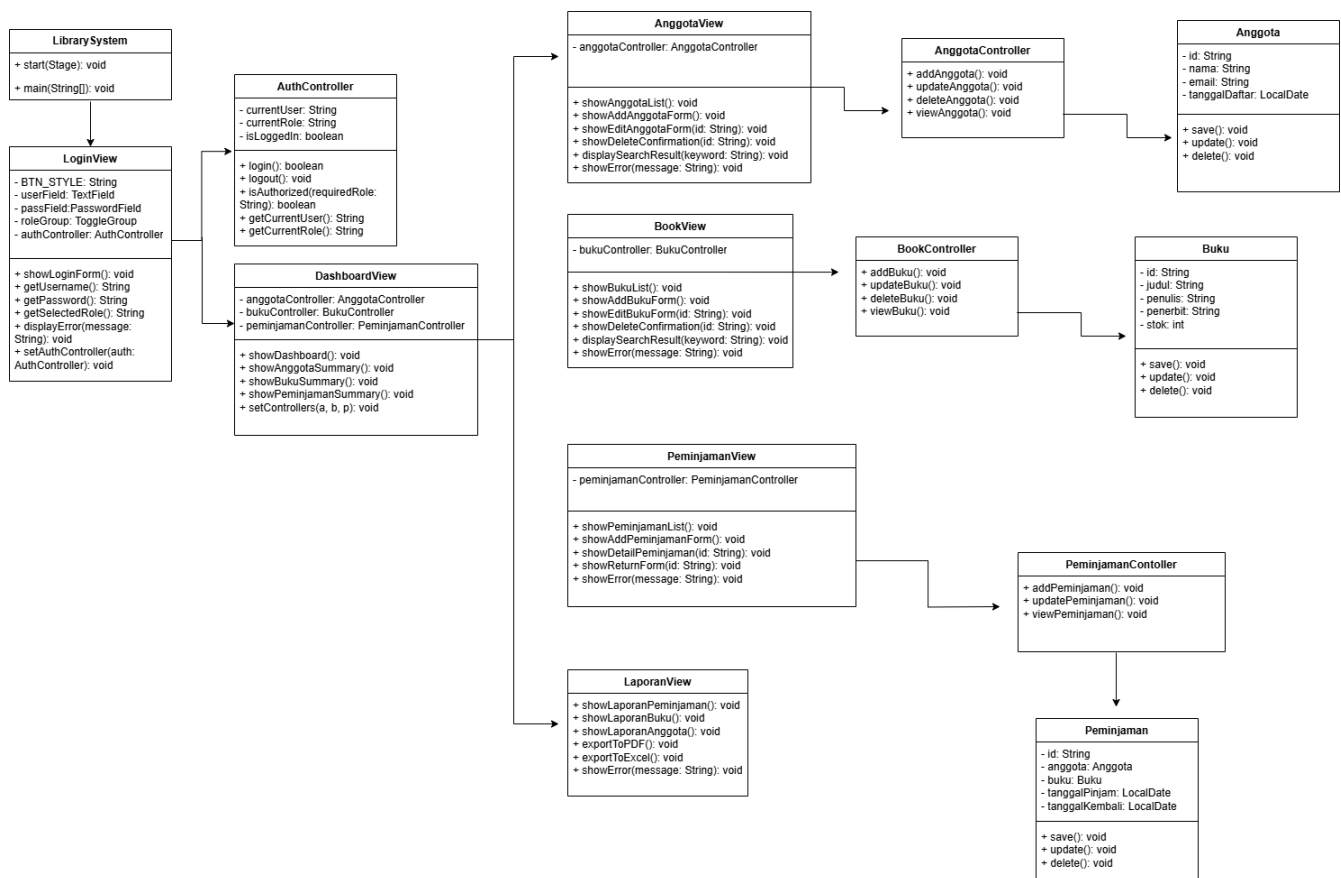
5. Database

Database berfungsi sebagai media penyimpanan permanen berbasis file untuk menyimpan data aplikasi seperti data anggota, data buku, dan data peminjaman. Sistem ini belum menggunakan database seperti MySQL, sehingga sebagai alternatifnya digunakan file .dat (binary/serialized file) untuk menyimpan data dalam format objek Java yang telah diserialisasi.

Setiap kali pengguna menambahkan data baru (misalnya anggota atau buku), sistem akan menulis (write) data tersebut ke file terkait menggunakan mekanisme ObjectOutputStream. Begitu pula saat aplikasi dijalankan kembali, sistem akan membaca (read) isi file .dat melalui ObjectInputStream, lalu mengubahnya kembali menjadi daftar objek (List<Anggota>, List<Buku>, dll) yang dapat digunakan oleh program. Dengan kata lain, folder data berfungsi sebagai pengganti database sederhana yang memungkinkan sistem menyimpan dan memuat data secara berkelanjutan meskipun aplikasi ditutup.

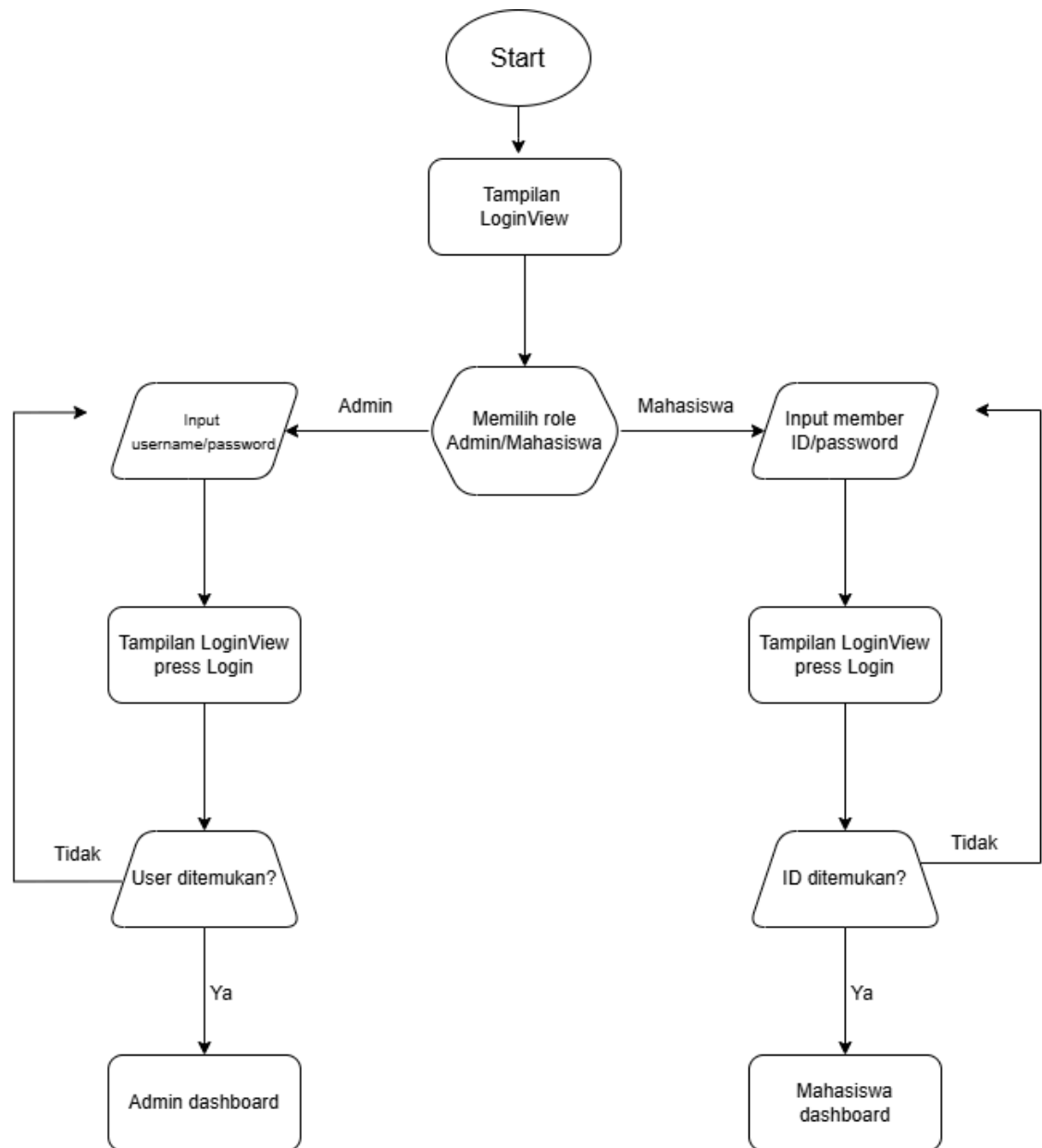
Secara umum, fungsi utama folder data dalam LibrarySystem adalah menyimpan semua informasi yang dimasukkan pengguna selama penggunaan aplikasi, memuat kembali data lama saat aplikasi dijalankan ulang dan menjaga data tetap tersimpan secara lokal tanpa koneksi ke database eksternal.

Diagram Class



Gambar 1. Diagram Class

Diagram Alir



Gambar 2. Diagram Alir

User Instruction Installisasi GUI & JavaFX di IntelliJ

1. Persiapan Awal

Sebelum memulai, pastikan hal-hal berikut telah tersedia dan terpasang di perangkat Anda:

- JDK (Java Development Kit) minimal versi 17.
- JavaFX SDK versi 17 (bisa diunduh dari: <https://gluonhq.com/products/javafx>).
- IntelliJ IDEA sebagai IDE untuk pengembangan.
- Koneksi internet aktif untuk mengunduh dependensi tambahan jika diperlukan.

2. Langkah Installasi

Unduh dan Ekstrak JavaFX SDK

- Kunjungi situs resmi Gluon.
- Pilih dan unduh JavaFX SDK sesuai sistem operasi Anda.
- Ekstrak hasil unduhan ke direktori yang mudah diakses, contoh: `C:\javafx-sdk-17`.

Buat Proyek Java Baru di IntelliJ

- Buka IntelliJ IDEA.
- Pilih **File > New > Project**.
- Pilih template **Java**, lalu pastikan JDK 17 atau versi lebih baru dipilih.
- Klik **Next**, beri nama proyek sesuai keinginan, lalu klik **Finish**.

Tambahkan JavaFX ke Library Proyek

- Buka **File > Project Structure** atau tekan `Ctrl + Alt + Shift + S`.
- Masuk ke tab **Libraries**, lalu klik ikon **+** dan pilih **Java**. Arahkan ke folder lib di dalam direktori JavaFX SDK, misalnya `C:\javafx-sdk-17\lib`.
- Pilih semua file `.jar`, lalu klik **OK**, lalu **Apply** dan **OK**.

Atur VM Options untuk Mendukung Modul JavaFX

- Buka menu **Run > Edit Configurations**. Di bagian **VM Options**, tambahkan baris berikut:


```
--module-path "C:\javafx-sdk-17\lib"
--add-modules javafx.controls,javafx.fxml
```
- Pastikan path disesuaikan dengan lokasi sebenarnya di komputer Anda.
- Klik **Apply**, lalu **OK**

User Instruction Instalasi MySQL Community Edition

1. Persiapan Awal

Pastikan perangkat telah memenuhi kebutuhan berikut:

- Sistem operasi Windows / macOS / Linux.
- Koneksi internet stabil.
- Ruang penyimpanan kosong minimal 500 MB.
- Hak akses administrator (untuk instalasi).

2. Langkah Instalasi MySQL (Windows/macOS/Linux)

Unduh Installer MySQL

- Buka situs resmi: <https://dev.mysql.com/downloads/mysql/>
- Pilih versi **MySQL Community Server**.
- Pilih sistem operasi yang sesuai (Windows, macOS, atau Linux).
- Klik **Download** dan simpan installer ke perangkat.

Jalankan Installer

- Buka file installer yang telah diunduh.
- Pilih tipe instalasi: **Developer Default** (disarankan, mencakup MySQL Server, Workbench, Shell, dll).
- Klik **Next** untuk melanjutkan.

Konfigurasi MySQL Server

- Pilih port (default: 3306).
- Pilih mode autentikasi: **Use Strong Password Encryption**.
- Buat dan simpan **root password** yang mudah diingat dan aman.

Selesaikan Instalasi

- Klik **Execute** untuk memulai proses instalasi dan konfigurasi.
- Tunggu hingga semua status bertanda "**Complete**".
- Klik **Finish**.

Verifikasi Instalasi

- Buka **MySQL Workbench** (jika ikut diinstal).
- Lakukan koneksi ke **localhost** menggunakan user **root** dan password yang telah dibuat.
- Jika berhasil login, instalasi selesai.

