

LAPORAN FINAL PROJECT PEMROGRAMAN BERORIENTASI OBJEK SISTEM PERPUSTAKAAN

Laporan ini dibuat untuk menyelesaikan tugas akhir mata kuliah Pemrograman Berorientasi
Objek

Yang dibina oleh : Ir. Galih Wasis Wicaksono, S.Kom., M.Cs.

Disusun oleh Team Dash:

1. Naufal Arkaan | Lead Developer | 202410370110020 | 2B
2. Muhamad Robi Ardita | Frontend Specialist | 202410370110002 | 2B
3. Akhmad Arjuan Syuhada | Backend Specialist | 202410370110043 | 2B



UNIVERSITAS MUHAMMADIYAH MALANG

FAKULTAS TEKNIK

PROGRAM STUDI INFORMATIKA

2025

BAB I

Arsitektur Sistem

1.1 Konsep MVC

MVC adalah singkatan dari Model-View-Controller, yaitu sebuah pola arsitektur perangkat lunak yang memisahkan aplikasi menjadi tiga komponen utama:

1. Model

Model bertugas menangani logika bisnis dan data. Ini termasuk representasi data, pengolahan data, dan interaksi dengan database.

Struktur project kami:

- src/SQL_DATA/UserDAO.java
- src/Action/Transaction.java
- src/Action/AdminAction.java, MemberAction.java
- src/model/

Model ini:

- Menyimpan data entitas seperti user, transaksi, buku, dll.
- Berinteraksi langsung dengan database (MySQL via Railway).
- Berisi DAO (Data Access Object) yang memproses CRUD ke database.

2. View

View adalah tampilan antarmuka pengguna (UI). Dalam JavaFX, ini biasanya berupa file .fxml dan file controller yang terhubung.

Contoh file di project kami:

- File .fxml seperti: Register.fxml, Dashboard.fxml, dll.
- Digunakan bersama Scene Builder untuk mendesain GUI.

Contoh file FXML:

```
<Button fx:id="btnSubmit" onAction="#handleSubmit"/>
```

3. Controller

Controller bertugas sebagai jembatan antara View dan Model. Ia menangani input dari pengguna di View, memprosesnya dengan bantuan Model, dan mengupdate View.

Contoh beberapa package dan class:

- src/controller/admin/*.java
 - AddBookController.java
 - DashboardController.java
 - ReportsAndChartsController.java
- src/controller/mahasiswa/*.java
 - BorrowBookController.java
 - RegisterController.java

Contoh fungsi Controller:

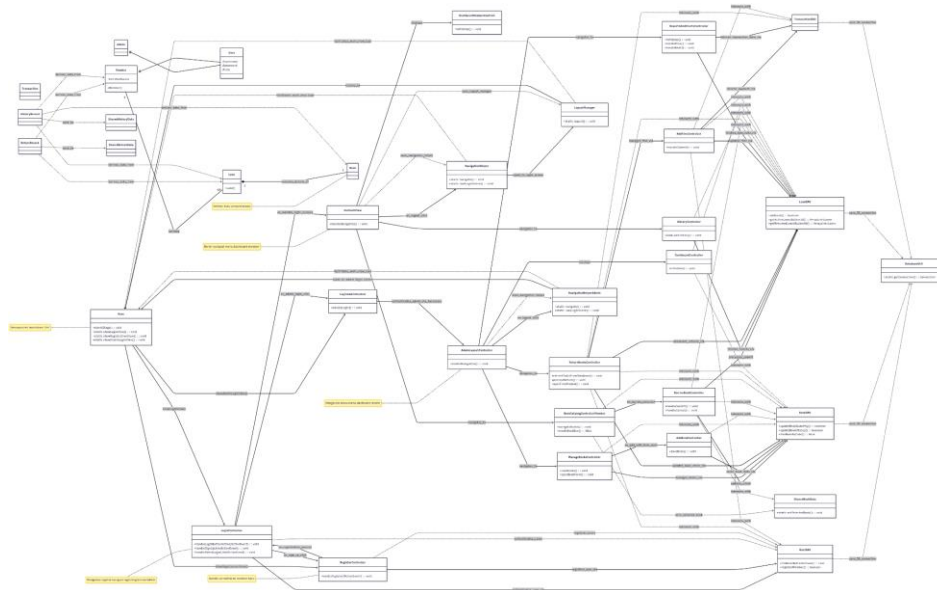
```
btnSubmit.setOnAction(e -> {
```

```
String username = usernameField.getText();
```

```
// Validasi dan simpan ke database via UserDAO (Model)
```

```
});
```

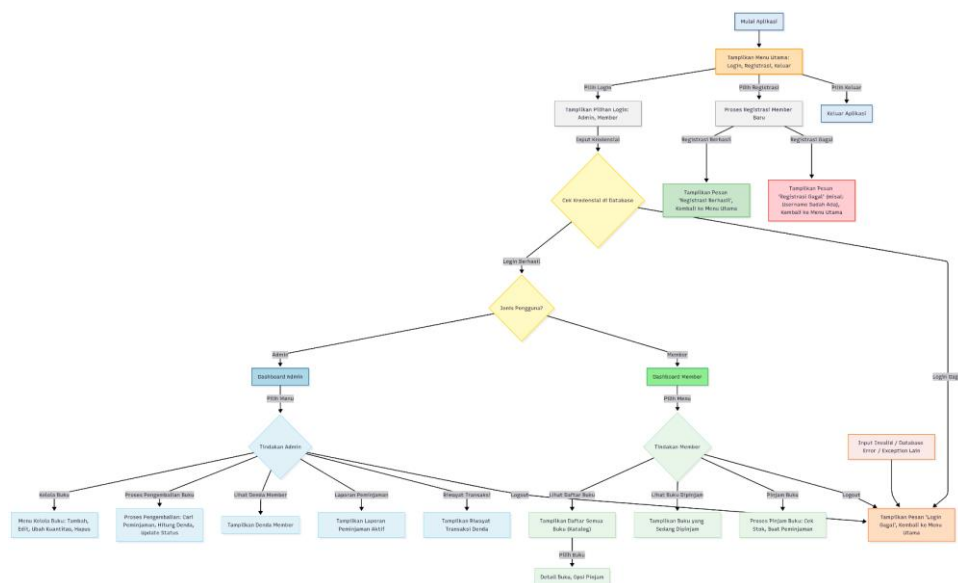
1.2 Diagram Kelas



Link Diagram Kelas

[Class Diagram Library System.png](#)

1.3 Diagram Alir



[Flow Chart](#)

[Library System.png](#)

➤ **Menu utama (Library System)**

Saat pertama kali membuka sistem, pengguna akan disugahi tiga pilihan utama:

1. **Login**
Untuk masuk ke sistem sebagai Admin atau Member/Mahasiswa.
2. **Registrasi**
Untuk mendaftar sebagai pengguna baru (khusus Member/Mahasiswa).
3. **Keluar**
Untuk menutup atau keluar dari sistem.

➤ **Login**

● **Admin**

- **Melihat Dashboard** statistik singkat.
- **Mengelola Buku**: tambah, edit, hapus, dan atur stok.
- **Memproses Pengembalian Buku** dari member.
- **Melihat Denda Member** yang belum dibayar.
- **Melihat Laporan Peminjaman** aktif.
- **Melihat Riwayat Pembayaran Denda**.
- **Logout**.

● **Member/Mahasiswa**

- **Lihat Profil**.
- **Lihat Daftar Buku** di perpustakaan.
- **Lihat Buku yang Dipinjam**.
- **Pinjam Buku** baru jika tersedia.
- **Logout**.

➤ **Registrasi**

Untuk member baru, isi data:

- **Username, password, nama lengkap, jurusan, email, dan Student ID.**

➤ **Keluar**

Fungsi ini mengembalikan pengguna ke luar dari sistem, biasanya digunakan jika pengguna tidak ingin login atau selesai menggunakan sistem.

BAB II

Instruksi Pengguna

2.1 Instruksi Pengguna (Versi Developer – IntelliJ IDEA)

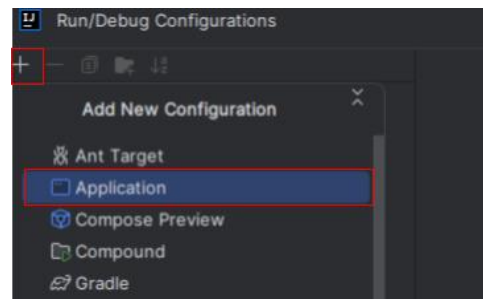
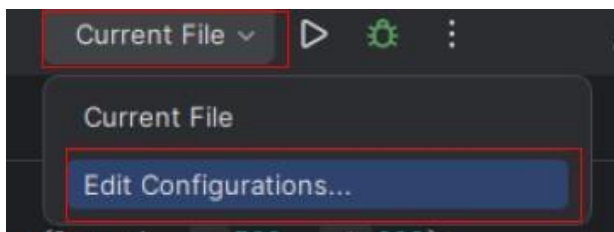
Bagian ini ditujukan untuk pengguna atau pengembang yang ingin menjalankan sistem Library System dari repositori GitHub menggunakan IntelliJ IDEA.

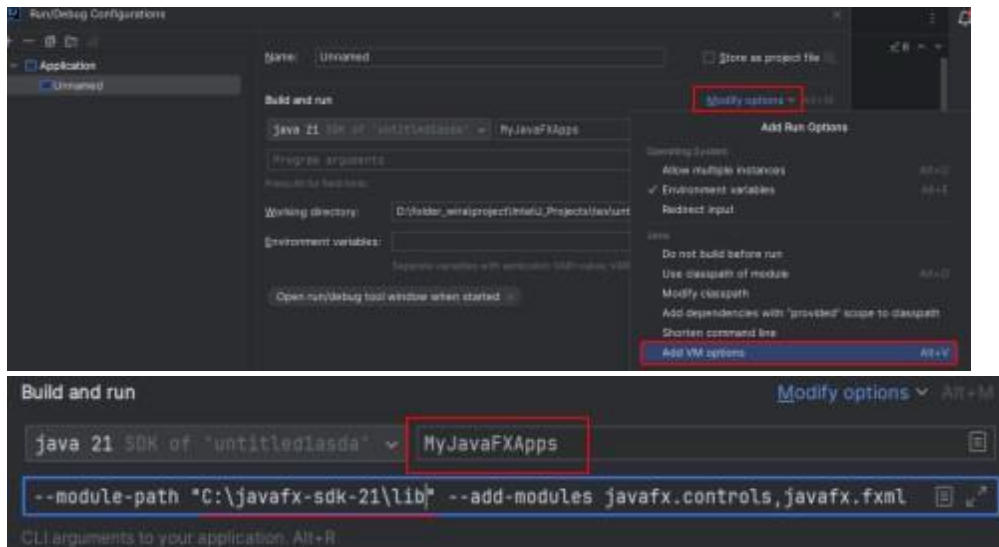
➤ 1. Clone Project dari GitHub

1. Buka terminal atau Git Bash.
2. Jalankan perintah berikut untuk meng-clone repository:
`git clone https://github.com/NaufalArkaan/OOPLibrarySystem_Team-Dash.git`
3. Setelah selesai, buka **IntelliJ IDEA**.
4. Pilih menu **Open** dan arahkan ke folder Final_Project_PBO_Team-Dash hasil clone tadi.

➤ 2. Setting Library dan Struktur Project

- a. Atur SDK (JDK)
 1. Masuk ke menu **File > Project Structure (Ctrl+Alt+Shift+S)**.
 2. Di tab **Project**, pastikan **Project SDK** sudah disetel (misalnya Java 23).
 - Kalau belum ada, klik **Add SDK > JDK** dan arahkan ke folder JDK di komputer kamu.
 3. Pastikan **Project Language Level** sesuai
- b. Tambah Library JavaFX
 1. Download JavaFX SDK dari <https://gluonhq.com/products/javafx/> (Note: Pakai JavaFX versi 17 saja untuk menghindari error versi tidak kompatibel).
 2. Setelah itu silahkan extract file .zip nya.
 3. Arahkan folder openjfx-17.0.15_windows-x64_bin-sdk, buka Project Structure > Libraries, lalu tambahkan folder lib dari JavaFX SDK di lokasi tempat kalian ekstrak .zip nya. Kemudian tekan OK terus sampai selesai.
 4. Setelah itu, Atur VM Options saat menjalankan aplikasi. Ikuti contoh dibawah.

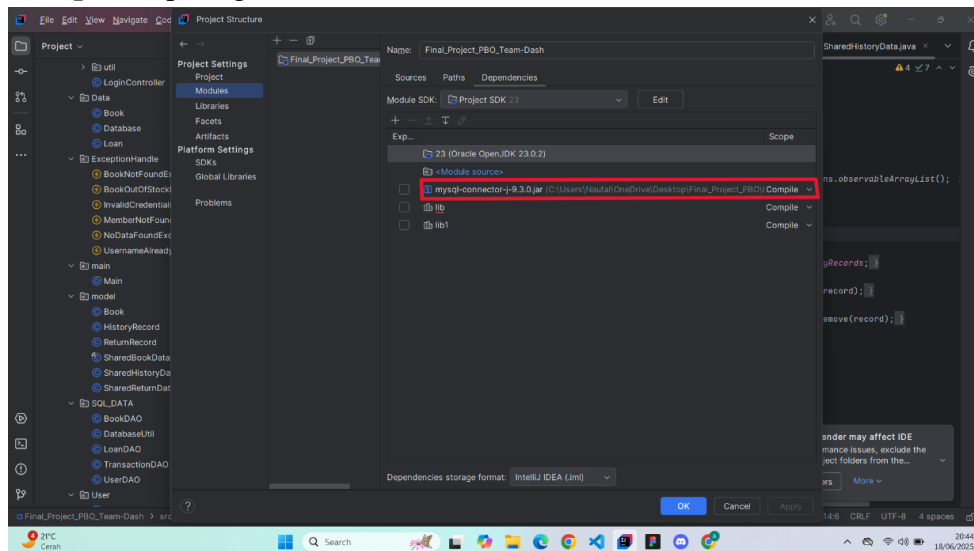




5. Di bagian “C:\javafx-sdk-21\lib” ganti dengan lokasi lib dari javafx-17 kalian. Untuk bagian Main Class nya isi dengan class yang ada fungsi “Main” nya (termasuk dengan packagenya). Jika sudah klik ok terus sampai selesai.

➤ 3. Menambahkan MySQL.jar ke intelliJ

1. Masuk ke halaman <https://dev.mysql.com/downloads/connector/j/> , kemudian Select Operating System: **Platform Independent**, lanjut klik download **Platform Independent (Architecture Independent)**, ZIP Archive.
2. Setelah itu silahkan extract file .zip nya.
3. Masukkan file mysql-connector-j-9.3.0 ke dalam **project structure > Modules >** klik icon “+”, dan masukan file mysql-connector-j-9.3.0 dengan status **Compile**, seperti gambar di bawah ini:



4. Setelah itu klik Apply dan OK

➤ 4. Jalankan Aplikasi

1. Klik kanan pada file Main.java, pilih Run 'Main' atau gunakan tombol Shift + F10.
2. Aplikasi akan berjalan, dan kamu bisa mulai menggunakan sistem sesuai alur login:
 - Admin login untuk mengelola data

- Member login untuk pinjam buku

2.2 Intruksi Pengguna Menggunakan Sistem Perpustakaan

1. Registrasi

- Pilih menu Registrasi di halaman utama.
- Isi data: username, password, nama lengkap, jurusan, email, dan student ID.
- Klik **Submit** untuk membuat akun.

2. Login

- Klik Login, masukkan username dan password.
- Setelah login, pengguna akan diarahkan sesuai peran:
 - **Admin** → ke halaman admin.
 - **Member** → ke halaman pengguna biasa.

3. Fitur Admin

- Kelola data buku: tambah, edit, hapus, atur stok.
- Proses pengembalian buku.
- Lihat denda, laporan peminjaman, dan riwayat pembayaran.
- Logout jika sudah selesai.

4. Fitur Member

- Lihat profil.
- Lihat daftar semua buku.
- Lihat buku yang sedang dipinjam.
- Pinjam buku jika tersedia.
- Logout setelah selesai.

5. Keluar dari Sistem

- Gunakan menu Keluar di halaman utama untuk menutup sesi.