

# **MAKALAH PROJEK PERPUSTAKAAN PEMROGRAMAN BERORIENTASI OBYEK**

Makalah ini dibuat untuk menyelesaikan Projek Perpustakaan mata kuliah PBO  
yang dibina oleh : Mr. Galih Wasis Wicaksono

Disusun Oleh :

Iyaka Samanda Caysar 202410370110394

Mukhammad Farhan Prayoga 202410370110220

Zinedine Rivan Abdus Syukur 202410370110364

Kelas: 1A



**UNIVERSITAS MUHAMMADIYAH MALANG**

**FAKULTAS TEKNIK**

**PRODI INFORMATIKA 2024**

## ABSTRAK

Makalah ini menyajikan pengembangan aplikasi sistem informasi perpustakaan berbasis Java dengan pendekatan arsitektur MVC (Model-View-Controller). Aplikasi ini dikembangkan untuk mendukung proses digitalisasi perpustakaan, terutama dalam hal manajemen data buku, anggota, dan transaksi peminjaman. Dengan pemanfaatan konsep pemrograman berorientasi objek (OOP), aplikasi memiliki struktur modular dan dapat dikembangkan lebih lanjut. Penggunaan Java GUI menjadikan antarmuka aplikasi ramah pengguna dan efisien.

---

## DAFTAR ISI

<b>DAFTAR ISI .....</b>	<b>2</b>
<b>BAB I - PENDAHULUAN.....</b>	<b>3</b>
1.1 Latar Belakang .....	3
1.2 Rumusan Masalah .....	4
<b>BAB II - TUJUAN PENELITIAN.....</b>	<b>5</b>
2.1 Tujuan Umum.....	5
2.2 Tujuan Khusus .....	5
<b>BAB III - TINJAUAN PUSTAKA.....</b>	<b>6</b>
3.1 Sistem Informasi Perpustakaan .....	6
3.2 Pemrograman Berorientasi Objek (OOP) .....	7
3.4 Konsep Pemrograman Berorientasi Objek (OOP) .....	7
3.5 Basis Data dan Penyimpanan Data Sederhana .....	8
<b>BAB IV - METODOLOGI PENGEMBANGAN .....</b>	<b>8</b>
4.1 Metodologi Waterfall.....	8
4.2 Alat dan Bahan .....	9
<b>BAB V - PERANCANGAN SISTEM .....</b>	<b>9</b>
5.1 Arsitektur MVC.....	9
5.2 Struktur Folder.....	10
5.3 Diagram Konteks .....	10
<b>BAB VI - IMPLEMENTASI SISTEM .....</b>	<b>11</b>

6.1 Struktur Proyek.....	11
6.2 Model .....	11
6.3 Controller.....	11
6.4 View (GUI).....	11
6.5 File CSV .....	12
<b>BAB VII - ALUR KERJA APLIKASI .....</b>	<b>12</b>
7.1 Diagram Alur Umum Sistem.....	12
7.2 Alur Detail Fungsional .....	12
<b>BAB VIII - TEKNOLOGI YANG DIGUNAKAN .....</b>	<b>14</b>
8.1 Bahasa Pemrograman .....	14
8.2 GUI Toolkit.....	14
8.3 Manajemen Data .....	14
8.4 IDE dan Tools Pendukung .....	15
8.5 Arsitektur MVC.....	15
<b>BAB IX - KESIMPULAN .....</b>	<b>16</b>
<b>DAFTAR PUSTAKA.....</b>	<b>16</b>

---

## **BAB I - PENDAHULUAN**

### **1.1 Latar Belakang**

Perpustakaan merupakan salah satu sarana penting dalam menunjang proses pembelajaran, penelitian, dan pengembangan ilmu pengetahuan di lingkungan akademik maupun non-akademik. Dalam era digital seperti sekarang, keberadaan perpustakaan tidak hanya diukur dari jumlah koleksi bukunya, tetapi juga dari bagaimana sistem pengelolaan dan layanan informasinya dijalankan. Namun, kenyataan di lapangan menunjukkan bahwa banyak institusi pendidikan, terutama sekolah dan perguruan tinggi, masih mengelola data perpustakaan secara manual. Hal ini dapat menimbulkan berbagai permasalahan, seperti kesalahan pencatatan, hilangnya data penting, hingga proses pencarian informasi yang memakan waktu lama.

Salah satu solusi efektif untuk mengatasi permasalahan tersebut adalah dengan mengembangkan sistem informasi perpustakaan yang berbasis komputer. Sistem ini mampu membantu petugas perpustakaan dalam mengelola data buku, transaksi peminjaman dan pengembalian, serta manajemen data pengguna dengan

lebih terstruktur, cepat, dan akurat. Melalui sistem ini, berbagai kegiatan perpustakaan dapat berjalan lebih efisien dan minim kesalahan.

Dalam pengembangan sistem informasi ini, pemilihan bahasa pemrograman menjadi aspek penting. Java dipilih karena memiliki keunggulan sebagai bahasa pemrograman yang bersifat **multiplatform**, artinya aplikasi yang dibangun dengan Java dapat dijalankan di berbagai sistem operasi seperti Windows, macOS, dan Linux tanpa banyak perubahan kode. Selain itu, Java dikenal sebagai bahasa yang **robust** dan **secure**, mendukung pemrograman berbasis objek (Object-Oriented Programming/OOP), serta memiliki pustaka antarmuka pengguna grafis (GUI) yang cukup lengkap seperti Swing dan JavaFX.

Untuk membangun aplikasi yang terstruktur dan mudah dikembangkan, digunakanlah pendekatan arsitektur **Model-View-Controller (MVC)**. Pendekatan ini memisahkan logika program (Model), tampilan antarmuka pengguna (View), dan kendali aplikasi (Controller), sehingga pengembangan dan pemeliharaan sistem menjadi lebih mudah dan efisien.

Sistem informasi perpustakaan yang dikembangkan dalam makalah ini tidak hanya mendukung fitur dasar seperti manajemen buku dan transaksi, tetapi juga mencerminkan praktik pengembangan perangkat lunak yang baik, mulai dari perancangan, implementasi, hingga pengujian aplikasi. Dengan sistem ini, diharapkan pengelolaan perpustakaan di lingkungan pendidikan, khususnya di Universitas Muhammadiyah Malang, dapat dilakukan dengan lebih efektif dan profesional.

## 1.2 Rumusan Masalah

- Bagaimana merancang sistem informasi perpustakaan yang efisien dan terstruktur?
  - Bagaimana mengimplementasikan konsep MVC dalam aplikasi perpustakaan berbasis Java?
  - Apa saja fitur utama yang perlu disediakan dalam aplikasi manajemen perpustakaan?
-

## **BAB II - TUJUAN PENELITIAN**

### **2.1 Tujuan Umum**

Penelitian dan pengembangan sistem informasi perpustakaan ini bertujuan untuk menghasilkan sebuah aplikasi desktop berbasis Java yang dapat digunakan untuk mempermudah proses pengelolaan perpustakaan secara digital, mulai dari manajemen data buku, transaksi peminjaman dan pengembalian, hingga pelaporan dan histori penggunaan oleh pengguna. Sistem ini diharapkan dapat menjadi solusi digitalisasi layanan perpustakaan yang terstruktur, efisien, dan mudah digunakan.

### **2.2 Tujuan Khusus**

Secara khusus, tujuan dari proyek pengembangan sistem informasi perpustakaan ini meliputi:

#### **a. Merancang dan mengembangkan aplikasi manajemen perpustakaan yang lengkap**

Aplikasi ini dirancang untuk mencakup seluruh proses kerja dasar yang ada dalam sebuah sistem perpustakaan konvensional, seperti:

- Penambahan dan penghapusan data buku
- Pencarian buku berdasarkan kata kunci
- Transaksi peminjaman dan pengembalian
- Pengelolaan data pengguna atau anggota perpustakaan
- Pembuatan laporan transaksi dan statistik penggunaan

#### **b. Mengimplementasikan prinsip Object-Oriented Programming (OOP)**

Penggunaan paradigma pemrograman berbasis objek (OOP) seperti inheritance, encapsulation, polymorphism, dan abstraction bertujuan agar sistem yang dibangun memiliki struktur kode yang modular, reusable, dan mudah dikembangkan di masa mendatang.

#### **c. Menerapkan arsitektur Model-View-Controller (MVC)**

MVC dipilih sebagai arsitektur utama dalam pengembangan sistem. Dengan pembagian komponen Model, View, dan Controller secara terpisah,

sistem akan lebih mudah diuji, dipelihara, dan dikembangkan oleh tim pengembang yang berbeda.

#### **d. Mengintegrasikan GUI (Graphical User Interface) menggunakan Java Swing**

Aplikasi ini dibangun menggunakan Java Swing untuk menghadirkan antarmuka pengguna yang interaktif, responsif, dan mudah digunakan, bahkan oleh pengguna yang tidak memiliki latar belakang teknologi.

#### **e. Menyediakan dokumentasi dan struktur proyek yang rapi**

Setiap komponen dalam proyek, baik kode program, dokumentasi fungsional, maupun diagram sistem, akan disusun secara terstruktur agar dapat dimanfaatkan kembali oleh mahasiswa, dosen, atau pengembang lainnya yang ingin mengembangkan sistem ini lebih lanjut.

#### **f. Melakukan pengujian dan evaluasi sistem**

Sebelum aplikasi dinyatakan siap digunakan, dilakukan pengujian terhadap seluruh fungsi utama dan fitur untuk memastikan bahwa tidak terdapat bug, dan bahwa semua proses dapat berjalan sesuai yang diharapkan.

---

## **BAB III - TINJAUAN PUSTAKA**

Tinjauan pustaka berfungsi sebagai dasar teori dan referensi ilmiah yang mendukung pengembangan sistem informasi perpustakaan. Dalam pengembangan perangkat lunak ini, berbagai konsep dan teknologi menjadi landasan, baik dari sisi teknis pemrograman maupun pengelolaan sistem informasi.

### **3.1 Sistem Informasi Perpustakaan**

Menurut Jogiyanto (2005), sistem informasi merupakan kombinasi dari teknologi informasi dan aktivitas manusia yang menggunakan teknologi tersebut untuk mendukung operasi dan manajemen. Dalam konteks perpustakaan, sistem informasi digunakan untuk mengelola data koleksi buku, anggota, transaksi peminjaman dan pengembalian, serta laporan.

Perpustakaan modern harus mampu menyajikan informasi secara cepat dan akurat. Sistem informasi perpustakaan dapat membantu pustakawan dalam

mengurangi beban administrasi, meminimalisir kesalahan pencatatan, dan meningkatkan pelayanan kepada pengguna.

### 3.2 Pemrograman Berorientasi Objek (OOP)

Java adalah bahasa pemrograman yang bersifat *object-oriented*, kuat, aman, dan portabel. Java dikembangkan oleh Sun Microsystems dan saat ini berada di bawah pengelolaan Oracle. Java memiliki keunggulan dalam membangun aplikasi desktop melalui pustaka GUI seperti **Swing** dan **JavaFX**, serta dikenal dengan prinsip “Write Once, Run Anywhere”.

Menurut Deitel & Deitel (2014), Java merupakan pilihan yang tepat untuk membangun aplikasi berskala kecil hingga besar karena kestabilannya dan dukungan komunitas yang luas.

### 3.3 Model-View-Controller (MVC)

Model-View-Controller (MVC) adalah arsitektur perangkat lunak yang memisahkan logika bisnis, tampilan antarmuka, dan pengendali aplikasi ke dalam tiga komponen berbeda:

- **Model:** Komponen yang berhubungan dengan data, logika bisnis, dan aturan pemrosesan.
- **View:** Bagian yang menampilkan antarmuka pengguna.
- **Controller:** Bertugas menerima input dari pengguna dan menentukan tindakan yang akan dilakukan terhadap model dan view

### 3.4 Konsep Pemrograman Berorientasi Objek (OOP)

OOP merupakan paradigma pemrograman yang menggunakan “objek” sebagai unit dasar dalam pemrograman. Konsep ini mencakup:

- **Enkapsulasi:** Menyembunyikan detail implementasi.
- **Pewarisan (Inheritance):** Pewarisan sifat dari kelas induk ke kelas anak.
- **Polimorfisme:** Objek dapat memiliki banyak bentuk.
- **Abstraksi:** Menyederhanakan kompleksitas dengan menyembunyikan detail yang tidak perlu.

Dengan menggunakan OOP, pengembangan sistem menjadi lebih modular, fleksibel, dan mudah dalam hal pemeliharaan serta pengembangan lebih lanjut.

### 3.5 Basis Data dan Penyimpanan Data Sederhana

Dalam proyek ini, penyimpanan data dilakukan menggunakan file CSV sebagai alternatif dari basis data relasional. File CSV (Comma Separated Values) mudah diakses, fleksibel, dan cukup untuk kebutuhan sistem perpustakaan skala kecil hingga menengah. Meskipun tidak sekuat RDBMS seperti MySQL atau PostgreSQL, CSV sudah cukup untuk kebutuhan pencatatan data buku, anggota, dan transaksi.

---

## BAB IV - METODOLOGI PENGEMBANGAN

Metodologi pengembangan perangkat lunak merujuk pada pendekatan sistematis yang digunakan untuk merencanakan, merancang, mengimplementasikan, dan menguji sistem yang akan dikembangkan. Pemilihan metodologi yang tepat akan menentukan keberhasilan suatu proyek perangkat lunak.

### 4.1 Metodologi Waterfall

Dalam pengembangan sistem informasi perpustakaan ini, digunakan metode **Waterfall** karena karakteristiknya yang cocok untuk proyek dengan kebutuhan yang sudah jelas di awal. Waterfall adalah metode pengembangan yang bersifat sekuensial dan linier. Setiap tahap dalam siklus hidup perangkat lunak harus diselesaikan sepenuhnya sebelum melanjutkan ke tahap berikutnya.

#### Tahapan dalam metode Waterfall yang diterapkan:

1. **Analisis Kebutuhan**
  - Mengumpulkan informasi terkait kebutuhan pengguna, fungsionalitas sistem, serta data yang akan dikelola seperti data buku, data anggota, dan data transaksi.
2. **Perancangan Sistem (Design)**
  - Menyusun struktur sistem, diagram alur proses, desain GUI, dan arsitektur kode menggunakan konsep Model-View-Controller (MVC).
3. **Implementasi**
  - Mengubah desain ke dalam bentuk kode program menggunakan bahasa Java, dengan menerapkan prinsip OOP dan pemisahan modul MVC.
4. **Pengujian (Testing)**



- Melakukan uji coba terhadap seluruh fitur yang dikembangkan untuk memastikan sistem berjalan sesuai harapan. Pengujian dilakukan baik secara fungsional maupun dari sisi antarmuka pengguna.
5. **Pemeliharaan (Maintenance)**
- Setelah aplikasi digunakan, dilakukan evaluasi dan perbaikan terhadap bug serta penambahan fitur sesuai kebutuhan pengguna di masa mendatang.

## 4.2 Alat dan Bahan

- Java SDK
  - IDE: IntelliJ IDEA
  - File CSV sebagai penyimpanan data sementara
  - JavaFX/Swing untuk antarmuka pengguna
- 

# BAB V - PERANCANGAN SISTEM

Perancangan sistem merupakan tahap krusial yang menjembatani antara analisis kebutuhan dengan implementasi teknis. Pada tahap ini, sistem informasi perpustakaan dirancang menggunakan pendekatan modular dan arsitektur MVC agar dapat dikembangkan secara terstruktur dan efisien.

## 5.1 Arsitektur MVC

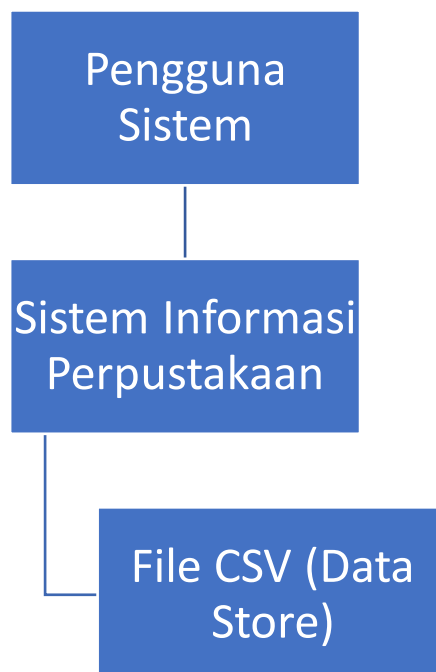
Sistem ini dibangun menggunakan **arsitektur Model-View-Controller (MVC)** yang memisahkan logika aplikasi ke dalam tiga komponen utama:

- **Model:** Mengelola data dan logika bisnis, seperti entitas Buku, User, Peminjaman.
- **View:** Mengatur tampilan antarmuka pengguna menggunakan Java Swing.
- **Controller:** Mengatur alur interaksi antara View dan Model.

## 5.2 Struktur Folder

```
css
CopyEdit
src/
├── dao/
│   └── CSVHelper.java
├── gui/
│   ├── book/BookManagementPage.java
│   ├── common/LoginPage.java, RegisterPage.java
│   ├── transaction/TransactionPage.java
│   └── util/AlertUtil.java
├── main/LibraryApp.java
├── model/Book.java, Member.java, Transaction.java
└── service/LibraryManager.java
```

## 5.3 Diagram Konteks



## BAB VI - IMPLEMENTASI SISTEM

Tahap implementasi merupakan realisasi dari rancangan sistem ke dalam bentuk kode program yang berjalan. Sistem informasi perpustakaan ini dikembangkan menggunakan bahasa pemrograman Java dengan pendekatan berorientasi objek dan pola arsitektur Model-View-Controller (MVC).

Sistem informasi perpustakaan ini diimplementasikan menggunakan bahasa Java dengan pendekatan Model-View-Controller (MVC) dan konsep Object-Oriented Programming (OOP).

### 6.1 Struktur Proyek

Proyek terbagi dalam paket:

- **model:** Kelas data (Buku, User, Transaksi)
- **view:** Antarmuka GUI (LoginView, DashboardView, BukuView)
- **controller:** Logika aplikasi (Login, Buku, Transaksi)
- **database:** Akses file CSV (buku.csv, user.csv, transaksi.csv)

### 6.2 Model

Kelas model merepresentasikan entitas seperti Buku, User, dan Peminjaman lengkap dengan atribut dan metode getter/setter.

### 6.3 Controller

Controller bertugas menangani logika aplikasi, seperti menambah buku, memverifikasi login, dan mengelola transaksi. Data disimpan dan diambil dari file CSV.

### 6.4 View (GUI)

Antarmuka dibangun menggunakan Java Swing. Tampilan utama meliputi:

- Login Admin
- Dashboard
- Form manajemen buku
- Form transaksi peminjaman dan pengembalian

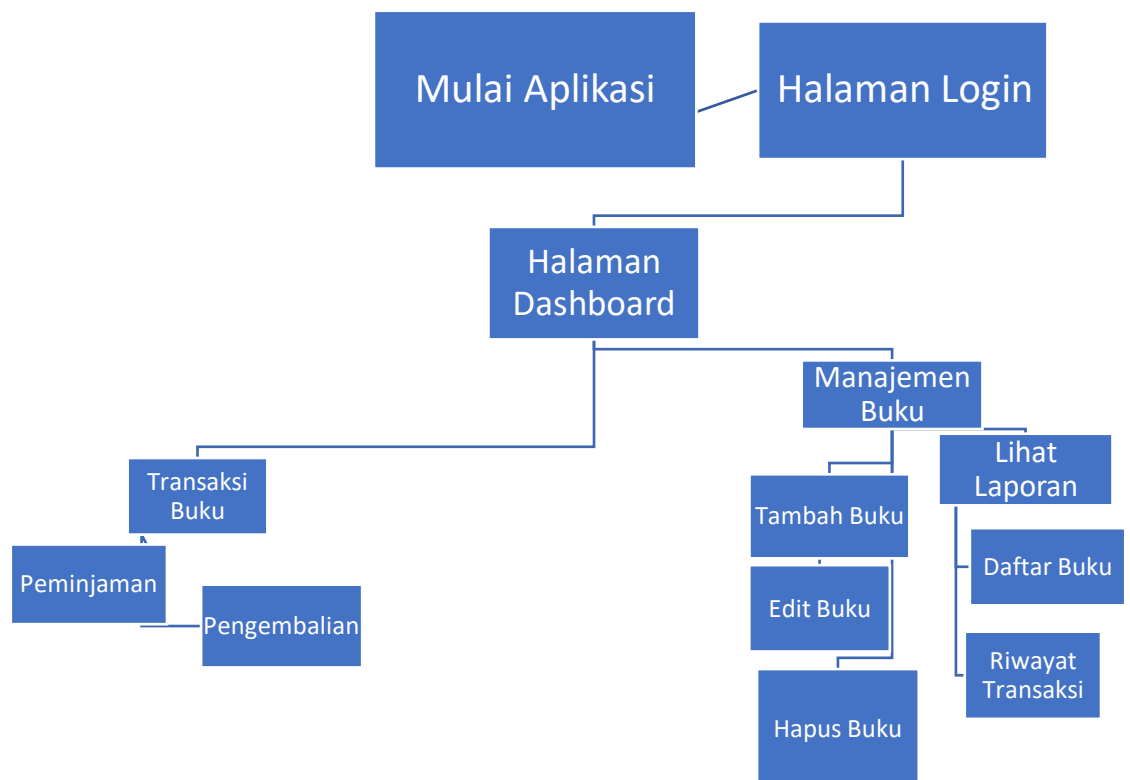
## 6.5 File CSV

Digunakan untuk menyimpan data secara sederhana, tanpa basis data kompleks. Pembacaan dan penulisan dilakukan menggunakan `BufferedReader` dan `BufferedWriter`.

---

# BAB VII - ALUR KERJA APLIKASI

## 7.1 Diagram Alur Umum Sistem



## 7.2 Alur Detail Fungsional

### 1. Login

- Pengguna memasukkan username dan password.
- Sistem memverifikasi dari file `user.csv`.
- Jika berhasil, pengguna masuk ke dashboard sesuai peran.

## **2. Dashboard**

- Menampilkan menu utama:
  - Buku
  - Transaksi
  - Laporan
  - Logout

## **3. Manajemen Buku**

- Admin dapat menambah, mengedit, dan menghapus data buku.
- Data tersimpan otomatis dalam file buku.csv.

## **4. Transaksi Peminjaman**

- Petugas memilih buku dan memasukkan data peminjam.
- Sistem menyimpan data ke peminjaman.csv.
- Jumlah buku otomatis berkurang.

## **5. Transaksi Pengembalian**

- Petugas memasukkan ID peminjaman.
- Sistem memperbarui status dan menambah jumlah buku kembali.

## **6. Laporan**

- Menampilkan seluruh daftar buku dan riwayat peminjaman dalam bentuk tabel.

## **7. Logout**

- Menutup sesi pengguna dan kembali ke halaman login.
-

## **BAB VIII - TEKNOLOGI YANG DIGUNAKAN**

Dalam pengembangan aplikasi **Sistem Informasi Perpustakaan UMM**, digunakan berbagai teknologi dan tools pendukung yang bersifat open-source, ringan, dan sesuai untuk pengembangan aplikasi desktop berbasis Java.

### **8.1 Bahasa Pemrograman**

Java SE (Standard Edition)

Digunakan sebagai bahasa utama karena mendukung pemrograman berorientasi objek (OOP), stabil, serta kompatibel di berbagai sistem operasi (multiplatform).

- Versi: Java JDK 8 atau lebih baru
- Paradigma: OOP dan Event-driven Programming

### **8.2 GUI Toolkit**

Java Swing

Merupakan library bawaan Java untuk membangun tampilan antarmuka (Graphical User Interface). Swing digunakan karena ringan, fleksibel, dan tidak memerlukan instalasi tambahan.

- Komponen yang digunakan: JFrame, JPanel, JTable, JButton, JLabel, JTextField, JPasswordField, dll.
- Keunggulan: Mudah dikustomisasi dan mendukung pembuatan form-form interaktif.

### **8.3 Manajemen Data**

File CSV (Comma-Separated Values)

Digunakan sebagai media penyimpanan data sederhana tanpa memerlukan database server. File CSV digunakan untuk menyimpan:

- Data buku (buku.csv)
- Data user (user.csv)
- Data transaksi (peminjaman.csv)

Pemrosesan dilakukan menggunakan:

- `BufferedReader`, `BufferedWriter` (untuk baca/tulis)
- `Scanner`, `FileWriter`

## 8.4 IDE dan Tools Pendukung

Tool	Fungsi
<b>NetBeans IDE</b>	Menulis, menjalankan, dan debug kode Java
<b>Draw.io</b>	Membuat diagram konteks dan DFD
<b>VS Code (opsional)</b>	Editor ringan alternatif
<b>Command Prompt / Terminal</b>	Menjalankan aplikasi Java secara langsung

## 8.5 Arsitektur MVC

Teknologi ini diimplementasikan dalam struktur kode sebagai:

- **Model:** Struktur data dan logika bisnis
- **View:** Antarmuka pengguna
- **Controller:** Penghubung antara model dan view

Pemisahan ini memudahkan pemeliharaan dan pengembangan lebih lanjut.

Komponen	Teknologi
Bahasa Pemrograman	Java 11+
GUI	JavaFX / Swing
Arsitektur	MVC
Persistensi	CSV File
IDE	IntelliJ IDEA
Paradigma	OOP

## BAB IX - KESIMPULAN

Pengembangan aplikasi perpustakaan berbasis Java ini menunjukkan bahwa arsitektur MVC efektif dalam memisahkan logika, tampilan, dan data. Sistem yang dibangun mampu mencatat data buku, anggota, dan transaksi dengan baik menggunakan antarmuka yang sederhana namun fungsional. Dengan pendekatan modular, aplikasi ini sangat fleksibel untuk dikembangkan, seperti integrasi ke database SQL atau menambahkan fitur multi-role user.

---

## DAFTAR PUSTAKA

- Deitel, H. M., & Deitel, P. J. (2012). *Java: How to Program* (9th ed.). Pearson Education.
- Eckel, B. (2006). *Thinking in Java* (4th ed.). Prentice Hall.
- Oracle. (2023). *Java Platform, Standard Edition Documentation*. Diakses dari: <https://docs.oracle.com/javase/8/docs/>
- Oracle. (2023). *Java Swing Tutorial*. Diakses dari: <https://docs.oracle.com/javase/tutorial/uiswing/>
- Siregar, R. (2021). *Belajar Java Dasar untuk Pemula*. Yogyakarta: Informatika.
- Sunarto, D. (2020). *Pemrograman Berorientasi Objek dengan Java*. Bandung: Penerbit Andi.
- W3Schools. (2023). *Java File Handling*. Diakses dari: [https://www.w3schools.com/java/java\\_files.asp](https://www.w3schools.com/java/java_files.asp)



