

# **SISTEM INFORMASI PERPUSTAKAAN DIGITAL**

Aplikasi Desktop Berbasis Java  
dengan MariaDB Database

Nama: Dico MAHENDAR

Kelas: C1

Nim: 24131310030

Dokumentasi Teknis  
Proyek Pemrograman Berorientasi Objek

**2026**

# DAFTAR ISI

BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Ruang Lingkup Sistem .....	2
1.3 Tujuan dan Manfaat .....	2
BAB II ANALISIS SISTEM .....	3
2.1 Permasalahan yang Diselesaikan .....	3
2.2 Analisis Kebutuhan Sistem .....	4
2.3 Aktor dan Peran .....	4
BAB III PERANCANGAN SISTEM .....	5
3.1 Use Case Diagram .....	5
3.2 Class Diagram .....	6
3.3 Activity Diagram .....	7
3.4 Sequence Diagram .....	8
3.5 Entity Relationship Diagram .....	9
BAB IV IMPLEMENTASI .....	10
4.1 Teknologi yang Digunakan .....	10
4.2 Struktur Kode Program .....	11
4.3 Implementasi Class .....	12
4.4 Implementasi Database .....	14
BAB V USER INTERFACE .....	15

5.1 Tampilan Login .....	15
5.2 Tampilan Dashboard .....	16
5.3 Manajemen Buku .....	17
5.4 Manajemen Anggota .....	18
5.5 Transaksi Peminjaman .....	19
BAB VI TESTING DAN EVALUASI .....	20
6.1 Testing Fungsional .....	20
6.2 Hasil Testing .....	21
PENUTUP .....	22

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perpustakaan merupakan salah satu sumber informasi dan pengetahuan yang penting dalam dunia pendidikan. Namun, banyak perpustakaan yang masih menggunakan sistem manual dalam pengelolaan data buku, anggota, dan transaksi peminjaman. Sistem manual ini sering kali menimbulkan berbagai permasalahan seperti kesulitan dalam pencarian data, pengelolaan stok buku yang tidak akurat, dan proses administrasi yang memakan waktu.

Untuk mengatasi permasalahan tersebut, dikembangkan Sistem Informasi Perpustakaan Digital berbasis desktop menggunakan bahasa pemrograman Java dan database MariaDB. Sistem ini dirancang untuk mempermudah pengelolaan perpustakaan secara terkomputerisasi, mulai dari manajemen data buku, anggota, hingga transaksi peminjaman dan pengembalian dengan sistem denda otomatis.

### **1.2 Ruang Lingkup Sistem**

Sistem ini mencakup beberapa modul utama:

- Manajemen Data Buku - meliputi penambahan, pengubahan, penghapusan, dan pencarian data buku
- Manajemen Data Anggota - registrasi anggota baru, update data, dan manajemen status keanggotaan
- Transaksi Peminjaman - proses peminjaman buku dengan validasi ketersediaan
- Transaksi Pengembalian - proses pengembalian dengan perhitungan denda otomatis
- Manajemen Kategori - pengelompokan buku berdasarkan kategori

- Laporan dan Statistik - pembuatan laporan data buku, peminjaman, dan statistik perpustakaan
- Manajemen Petugas - multi-level user dengan hak akses berbeda (Admin dan Petugas)

## **1.3 Tujuan dan Manfaat**

### **Tujuan:**

- Mengembangkan sistem informasi perpustakaan yang efisien dan user-friendly
- Mengotomatisasi proses administrasi perpustakaan
- Menyediakan sistem tracking peminjaman secara real-time
- Mengimplementasikan konsep Pemrograman Berorientasi Objek dalam aplikasi nyata

### **Manfaat:**

- Efisiensi waktu dalam pengelolaan data perpustakaan
- Akurasi data yang lebih baik
- Kemudahan dalam pencarian dan tracking buku
- Laporan yang dapat dihasilkan secara otomatis
- Mengurangi kesalahan human error dalam perhitungan denda

## **BAB II**

### **ANALISIS SISTEM**

#### **2.1 Permasalahan yang Diselesaikan**

Berdasarkan observasi terhadap sistem perpustakaan konvensional, ditemukan beberapa permasalahan utama yang memerlukan solusi:

**1. Pencatatan Data Manual** - Data buku dan anggota dicatat secara manual dalam buku besar, sehingga rentan terhadap kesalahan pencatatan dan kehilangan data.

**2. Kesulitan Pencarian** - Untuk menemukan informasi buku tertentu, petugas harus mencari secara manual melalui catatan yang ada, yang memakan waktu cukup lama.

**3. Tracking Peminjaman** - Sulit untuk mengetahui status peminjaman buku secara real-time, buku mana yang sedang dipinjam, siapa peminjamnya, dan kapan jatuh tempo pengembalian.

**4. Perhitungan Denda** - Perhitungan denda keterlambat dilakukan secara manual, sering terjadi kesalahan perhitungan yang merugikan perpustakaan atau anggota.

**5. Pembuatan Laporan** - Proses pembuatan laporan bulanan atau tahunan memerlukan waktu lama karena harus merekap data secara manual.

**6. Kontrol Stok** - Sulit untuk mengetahui jumlah stok buku yang tersedia secara akurat dan real-time.

#### **2.2 Analisis Kebutuhan Sistem**

**Kebutuhan Fungsional:**

- Sistem harus dapat menyimpan dan mengelola data buku dengan lengkap
- Sistem harus dapat mengelola data anggota perpustakaan
- Sistem harus dapat memproses transaksi peminjaman dan pengembalian
- Sistem harus dapat menghitung denda secara otomatis
- Sistem harus dapat menghasilkan laporan
- Sistem harus memiliki fitur pencarian data
- Sistem harus memiliki autentikasi login untuk keamanan

#### **Kebutuhan Non-Fungsional:**

- Sistem harus responsif dan mudah digunakan
- Sistem harus dapat diandalkan (reliable)
- Data harus aman dan terenkripsi
- Sistem harus memiliki performa yang baik
- Interface harus user-friendly dan intuitif

## **2.3 Aktor dan Peran**

Sistem memiliki dua jenis aktor dengan peran yang berbeda:

### **1. Administrator**

Administrator memiliki hak akses penuh terhadap sistem, meliputi:

- Mengelola data buku (tambah, ubah, hapus)
- Mengelola data anggota
- Mengelola data petugas
- Memproses transaksi peminjaman dan pengembalian
- Mengelola kategori buku
- Membuat dan melihat laporan
- Mengatur konfigurasi sistem

## **2. Petugas**

Petugas memiliki hak akses terbatas, meliputi:

- Mengelola data buku
- Mengelola data anggota
- Memproses transaksi peminjaman dan pengembalian
- Melihat laporan (tanpa bisa menghapus)
- Mengubah password sendiri



## **BAB III**

### **PERANCANGAN SISTEM**

#### **3.1 Use Case Diagram**

Use Case Diagram menggambarkan interaksi antara aktor dengan sistem. Diagram ini menunjukkan fungsionalitas yang dapat dilakukan oleh setiap aktor.

Pada diagram di atas, terdapat dua aktor utama yaitu Administrator dan Petugas. Administrator memiliki akses penuh ke semua fitur sistem, sedangkan Petugas memiliki akses terbatas sesuai dengan kebutuhan operasional harian perpustakaan.

#### **3.2 Class Diagram**

Class Diagram menggambarkan struktur kelas dalam sistem beserta relasi antar kelas. Sistem ini mengimplementasikan konsep Object-Oriented Programming dengan beberapa class utama:

Penjelasan Class:

- Kategori - Menyimpan informasi kategori buku
- Buku - Menyimpan informasi detail buku termasuk stok
- Anggota - Menyimpan informasi anggota perpustakaan
- Petugas - Menyimpan informasi petugas yang mengelola sistem

- Peminjaman - Menyimpan transaksi peminjaman dan pengembalian
- BukuDAO - Data Access Object untuk operasi database buku
- AnggotaDAO - Data Access Object untuk operasi database anggota
- PeminjamanDAO - Data Access Object untuk operasi database peminjaman

### **3.3 Activity Diagram**

Activity Diagram menggambarkan alur aktivitas dalam sistem. Berikut adalah activity diagram untuk proses peminjaman buku:

Alur proses:

1. Petugas login ke sistem
2. Petugas memilih menu peminjaman
3. Sistem menampilkan form peminjaman
4. Petugas memilih anggota dari database
5. Petugas memilih buku yang akan dipinjam
6. Sistem mengecek ketersediaan buku
7. Jika buku tersedia, sistem memproses peminjaman
8. Sistem mengurangi stok buku tersedia
9. Sistem menyimpan data peminjaman
10. Sistem menampilkan konfirmasi sukses

### 3.4 Sequence Diagram

Sequence Diagram menggambarkan interaksi antar objek dalam urutan waktu tertentu. Berikut adalah sequence diagram untuk proses login:

Diagram ini menunjukkan bagaimana objek LoginView, LoginController, PetugasDAO, dan Database berinteraksi dalam proses autentikasi pengguna.

### 3.5 Entity Relationship Diagram

Entity Relationship Diagram (ERD) menggambarkan struktur database dan relasi antar tabel. Database sistem ini menggunakan MariaDB dengan 5 tabel utama:

Relasi antar tabel:

- Kategori → Buku (One to Many) - Satu kategori dapat memiliki banyak buku
- Anggota → Peminjaman (One to Many) - Satu anggota dapat melakukan banyak peminjaman
- Buku → Peminjaman (One to Many) - Satu buku dapat dipinjam berkali-kali
- Petugas → Peminjaman (One to Many) - Satu petugas dapat memproses banyak peminjaman

# **BAB IV**

## **IMPLEMENTASI**

### **4.1 Teknologi yang Digunakan**

Sistem Informasi Perpustakaan Digital dikembangkan menggunakan teknologi berikut:

#### **1. Bahasa Pemrograman**

Java SE 11 - Dipilih karena portabilitas, stabilitas, dan dukungan library yang lengkap untuk pengembangan aplikasi desktop.

#### **2. GUI Framework**

Java Swing - Framework GUI bawaan Java yang menyediakan komponen-komponen untuk membuat interface desktop yang responsif dan user-friendly.

#### **3. Database**

MariaDB 10.x - Database management system open source yang kompatibel dengan MySQL, menawarkan performa tinggi dan reliabilitas untuk aplikasi desktop.

#### **4. Build Tool**

Apache Maven - Untuk manajemen dependensi dan build automation.

## 5. Libraries

- MariaDB JDBC Driver 3.1.4 - Koneksi Java ke database MariaDB
- JCalendar 1.4 - Komponen date picker untuk input tanggal
- iText PDF 5.5.13.3 - Pembuatan laporan dalam format PDF
- Apache POI 5.2.3 - Export data ke format Excel

## 4.2 Struktur Kode Program

Kode program diorganisir menggunakan struktur package Maven standar dengan pemisahan berdasarkan layer arsitektur MVC (Model-View-Controller):

```
PerpustakaanDigital/
├── src/main/java/com/perpustakaan/
│   ├── model/                # Entity classes
│   │   ├── Buku.java
│   │   ├── Anggota.java
│   │   ├── Peminjaman.java
│   │   ├── Petugas.java
│   │   └── Kategori.java
│   ├── dao/                  # Data Access Objects
│   │   ├── BukuDAO.java
│   │   ├── AnggotaDAO.java
│   │   ├── PeminjamanDAO.java
│   │   └── PetugasDAO.java
│   ├── controller/          # Business Logic
│   │   ├── BukuController.java
│   │   ├── AnggotaController.java
│   │   └── PeminjamanController.java
│   ├── view/                 # GUI Classes
│   │   ├── LoginFrame.java
│   │   ├── DashboardFrame.java
│   │   ├── BukuPanel.java
│   │   └── PeminjamanPanel.java
│   └── util/                  # Utility Classes
│       ├── DatabaseConnection.java
│       └── ValidationUtil.java
```

```

|   └─ MainApp.java      # Main Class
└─ database/
    └─ schema.sql        # Database Schema
└─ pom.xml                # Maven Configuration

```

Struktur ini mengikuti prinsip Separation of Concerns, dimana setiap layer memiliki tanggung jawab yang jelas dan terpisah.

## 4.3 Implementasi Class

Berikut adalah contoh implementasi beberapa class utama:

### Class Buku (Model)

```

public class Buku {
    private int idBuku;
    private String kodeBuku;
    private String judul;
    private String pengarang;
    private String penerbit;
    private int tahunTerbit;
    private String isbn;
    private int idKategori;
    private int jumlahTotal;
    private int jumlahTersedia;

    // Constructor, Getter, Setter
    // ...
}

```

### Class BukuDAO (Data Access Object)

```

public class BukuDAO {
    public boolean insert(Buku buku) {
        String sql = "INSERT INTO buku (kode_buku, judul, ...)
VALUES (?, ?, ...)";
        try (Connection conn =
DatabaseConnection.getConnection());

```

```

        PreparedStatement pstmt =
conn.prepareStatement(sql)) {
    pstmt.setString(1, buku.getKodeBuku());
    pstmt.setString(2, buku.getJudul());
    // ...
    return pstmt.executeUpdate() > 0;
} catch (SQLException e) {
    e.printStackTrace();
    return false;
}

}

public List<Buku> getAll() { /* ... */ }
public boolean update(Buku buku) { /* ... */ }
public boolean delete(int idBuku) { /* ... */ }
}

```

Class DAO bertanggung jawab untuk semua operasi database (CRUD - Create, Read, Update, Delete) terkait entitas tertentu. Implementasi menggunakan PreparedStatement untuk keamanan dari SQL Injection.

## 4.4 Implementasi Database

Database perpustakaan\_db terdiri dari 5 tabel utama dengan struktur sebagai berikut:

### Tabel Kategori

- id\_kategori (INT, PRIMARY KEY, AUTO\_INCREMENT)
- nama\_kategori (VARCHAR(100), NOT NULL)
- deskripsi (TEXT)
- created\_at (TIMESTAMP)
- updated\_at (TIMESTAMP)

### Tabel Buku

- id\_buku (INT, PRIMARY KEY, AUTO\_INCREMENT)
- kode\_buku (VARCHAR(20), UNIQUE, NOT NULL)
- judul (VARCHAR(200), NOT NULL)
- pengarang (VARCHAR(150), NOT NULL)
- penerbit (VARCHAR(150))
- tahun\_terbit (YEAR)
- isbn (VARCHAR(20))
- id\_kategori (INT, FOREIGN KEY)
- jumlah\_total (INT)
- jumlah\_tersedia (INT)
- lokasi\_rak (VARCHAR(50))

### **Tabel Anggota**

- id\_anggota (INT, PRIMARY KEY, AUTO\_INCREMENT)
- no\_anggota (VARCHAR(20), UNIQUE, NOT NULL)
- nama (VARCHAR(150), NOT NULL)
- email (VARCHAR(100))
- no\_telepon (VARCHAR(20))
- alamat (TEXT)
- tanggal\_daftar (DATE)
- tanggal\_expired (DATE)
- status (ENUM: Aktif, Nonaktif, Suspended)

### **Tabel Peminjaman**

- id\_peminjaman (INT, PRIMARY KEY, AUTO\_INCREMENT)
- kode\_peminjaman (VARCHAR(30), UNIQUE)
- id\_anggota (INT, FOREIGN KEY)
- id\_buku (INT, FOREIGN KEY)
- id\_petugas (INT, FOREIGN KEY)
- tanggal\_pinjam (DATE)



- tanggal\_jatuh\_tempo (DATE)
- tanggal\_kembali (DATE)
- status (ENUM: Dipinjam, Dikembalikan, Terlambat)
- denda (DECIMAL(10,2))
- catatan (TEXT)

## **BAB V**

### **USER INTERFACE**

Berikut adalah tampilan antarmuka (User Interface) dari Sistem Informasi Perpustakaan Digital:

#### **5.1 Tampilan Login**

Halaman login merupakan gerbang masuk ke sistem. Petugas harus memasukkan username dan password yang valid untuk dapat mengakses sistem.

Fitur pada halaman login:

- Input username dan password
- Validasi kredensial user
- Pesan error jika login gagal
- Tombol login dengan shortcut Enter
- Password terenkripsi (MD5)

#### **5.2 Tampilan Dashboard**

Dashboard adalah halaman utama setelah login berhasil. Menampilkan statistik perpustakaan dan menu navigasi untuk mengakses fitur-fitur sistem.

Informasi yang ditampilkan:

- Total jumlah buku dalam perpustakaan
- Total anggota terdaftar
- Jumlah buku yang sedang dipinjam
- Jumlah peminjaman yang terlambat
- Menu navigasi (Buku, Anggota, Peminjaman, Laporan)
- Informasi user yang sedang login

### **5.3 Manajemen Buku**

Panel manajemen buku menyediakan fitur lengkap untuk mengelola data buku.

*Gambar 5.3 Tampilan Manajemen Buku*

Fitur yang tersedia:

- Tabel daftar buku dengan informasi lengkap
- Tombol Tambah untuk menambah buku baru
- Tombol Edit untuk mengubah data buku
- Tombol Hapus untuk menghapus buku
- Pencarian buku berdasarkan kode, judul, atau pengarang
- Filter berdasarkan kategori
- Informasi stok buku tersedia

## 5.4 Manajemen Anggota

Panel manajemen anggota untuk mengelola data anggota perpustakaan.

Fitur yang tersedia:

- Tabel daftar anggota
- Form registrasi anggota baru
- Update data anggota
- Manajemen status keanggotaan
- Pencarian anggota
- Informasi masa berlaku keanggotaan
- Riwayat peminjaman per anggota

## 5.5 Transaksi Peminjaman

Panel transaksi untuk memproses peminjaman dan pengembalian buku.

*Gambar 5.5 Tampilan Transaksi Peminjaman*

Fitur yang tersedia:

- Form peminjaman buku baru
- Pemilihan anggota dari database

- Pemilihan buku yang akan dipinjam
- Penentuan tanggal jatuh tempo
- Validasi ketersediaan buku
- Proses pengembalian buku
- Perhitungan denda otomatis
- Riwayat transaksi peminjaman
- Cetak struk peminjaman

## BAB VI

### TESTING DAN EVALUASI

#### 6.1 Testing Fungsional

Testing fungsional dilakukan untuk memastikan semua fitur sistem berjalan sesuai dengan yang diharapkan. Berikut adalah hasil testing untuk setiap modul:

##### 1. Testing Modul Login

Test Case	Input	Expected	Status
Login valid	username: admin password: admin123	Login berhasil, masuk dashboard	PASS
Password salah	username: admin password: wrong	Error: password salah	PASS
User tidak ada	username: xxx password: xxx	Error: user tidak ditemukan	PASS
Field kosong	username: (kosong) password: (kosong)	Error: field tidak boleh kosong	PASS

##### 2. Testing Modul Buku

Test Case	Expected	Status
Tambah buku baru	Data buku tersimpan di database	PASS
Edit data buku	Data buku ter-update	PASS
Hapus buku	Buku terhapus dari database	PASS
Pencarian buku	Menampilkan hasil yang sesuai	PASS
Validasi ISBN	Menolak format ISBN invalid	PASS

### 3. Testing Modul Peminjaman

Test Case	Expected	Status
Proses peminjaman	Stok buku berkurang	PASS
Pengembalian tepat waktu	Denda = 0	PASS
Pengembalian terlambat	Denda terhitung otomatis	PASS
Buku tidak tersedia	Menampilkan pesan error	PASS
Anggota suspended	Tidak bisa meminjam	PASS

## 6.2 Hasil Testing

Dari seluruh test case yang dilakukan, sistem menunjukkan hasil yang memuaskan dengan tingkat keberhasilan 100%. Semua fitur utama berfungsi dengan baik sesuai spesifikasi.

Kesimpulan testing:

- Semua fitur CRUD (Create, Read, Update, Delete) berfungsi dengan baik
- Validasi input bekerja dengan sempurna
- Perhitungan denda otomatis akurat
- Interface responsif dan user-friendly
- Sistem stabil tanpa error fatal
- Performa database baik untuk operasi query

# PENUTUP

## Kesimpulan

Sistem Informasi Perpustakaan Digital telah berhasil dikembangkan dengan menerapkan konsep Pemrograman Berorientasi Objek menggunakan bahasa Java dan database MariaDB. Sistem ini menyediakan solusi komprehensif untuk mengatasi permasalahan pengelolaan perpustakaan secara manual.

Beberapa pencapaian utama dari sistem ini:

- Implementasi arsitektur MVC yang terstruktur dan mudah dimaintain
- Database yang terdesain dengan baik menggunakan normalisasi
- Interface yang intuitif dan mudah digunakan
- Fitur-fitur lengkap untuk operasional perpustakaan
- Sistem keamanan dengan autentikasi dan enkripsi password
- Perhitungan denda otomatis yang akurat
- Laporan yang dapat di-export ke PDF dan Excel

## Saran Pengembangan

Untuk pengembangan lebih lanjut, beberapa fitur yang dapat ditambahkan:

- Implementasi sistem notifikasi email untuk pengingat jatuh tempo
- Integrasi dengan barcode scanner untuk input buku lebih cepat
- Dashboard analytics dengan grafik statistik yang lebih lengkap
- Fitur booking/reservasi buku online
- Aplikasi mobile untuk member perpustakaan
- Sistem rekomendasi buku berdasarkan riwayat peminjaman
- Backup database otomatis
- Multi-branch support untuk perpustakaan dengan cabang



Demikian dokumentasi Sistem Informasi Perpustakaan Digital ini dibuat. Semoga bermanfaat untuk pembelajaran dan pengembangan lebih lanjut.