

# **PERANCANGAN PERANGKAT LUNAK ( SISTEM MANAJEMEN PERPUSTAKAAN )**

Disusun Untuk Memenuhi Ujian Akhir Semester ( **UAS** )

Pada Mata Kuliah Perancangan Perangkat Lunak

**Dosen Pengampu : Ahmad Roihan, S.Kom., M.Ti**



**DISUSUN OLEH :**

**TIARA FITRIANI : 23130210096**

**TEKNOLOGI INFORMASI 5A  
FAKULTAS TEKNIK  
UNIVERSITAS TANGERANG RAYA  
2026**

## **1. Identifikasi Permasalahan Utama dan Tujuan Pengembangan Sistem**

### **1.1 Permasalahan Utama**

Perpustakaan merupakan sarana penting dalam mendukung kegiatan akademik, penelitian, dan literasi. Namun, masih banyak perpustakaan di lingkungan kampus maupun sekolah yang menggunakan sistem pencatatan manual dalam pengelolaan data buku, anggota, serta transaksi peminjaman. Sistem manual tersebut menimbulkan beberapa permasalahan, yaitu:

1. Risiko kehilangan dan kerusakan data  
Data transaksi peminjaman masih disimpan dalam bentuk fisik seperti buku catatan atau kartu peminjaman sehingga rentan hilang atau rusak.
2. Proses pencarian buku tidak efisien  
Anggota kesulitan mencari buku karena pencarian dilakukan secara manual sehingga membutuhkan waktu lama.
3. Pencatatan transaksi peminjaman dan pengembalian lambat  
Petugas harus mencatat transaksi satu per satu, sehingga rawan kesalahan pencatatan.
4. Kesulitan membuat laporan perpustakaan  
Pembuatan laporan peminjaman dan pengembalian tidak dapat dilakukan secara cepat dan real-time.
5. Monitoring status buku tidak akurat  
Sistem manual sulit mengontrol status buku (tersedia/dipinjam) sehingga sering terjadi kesalahan informasi.

### **1.2 Tujuan Pengembangan Sistem**

Tujuan pengembangan Sistem Manajemen Perpustakaan berbasis web adalah:

- Membuat pengelolaan perpustakaan menjadi lebih efektif dan efisien.
- Mempermudah pustakawan dalam mengelola data buku, anggota, peminjaman, dan pengembalian.
- Memberikan fitur pencarian buku secara cepat kepada anggota.
- Menyediakan laporan perpustakaan secara otomatis dan akurat.
- Mendukung transformasi digital di bidang pendidikan melalui sistem informasi berbasis web.

## **2. Aktor dan Fungsi Utama Sistem serta Deskripsi Use Case**

### **2.1 Aktor Sistem**

Dalam sistem manajemen perpustakaan terdapat dua aktor utama:

1. Pustakawan (Admin)  
Bertugas mengelola seluruh data perpustakaan termasuk buku, anggota, transaksi peminjaman, pengembalian, dan laporan.
2. Anggota Perpustakaan  
Bertugas mencari buku, melihat detail buku, dan melihat riwayat peminjaman.

## 2.2 Fungsi Utama Sistem

Fungsi utama sistem meliputi:

### A. Fungsi Pustakawan (Admin)

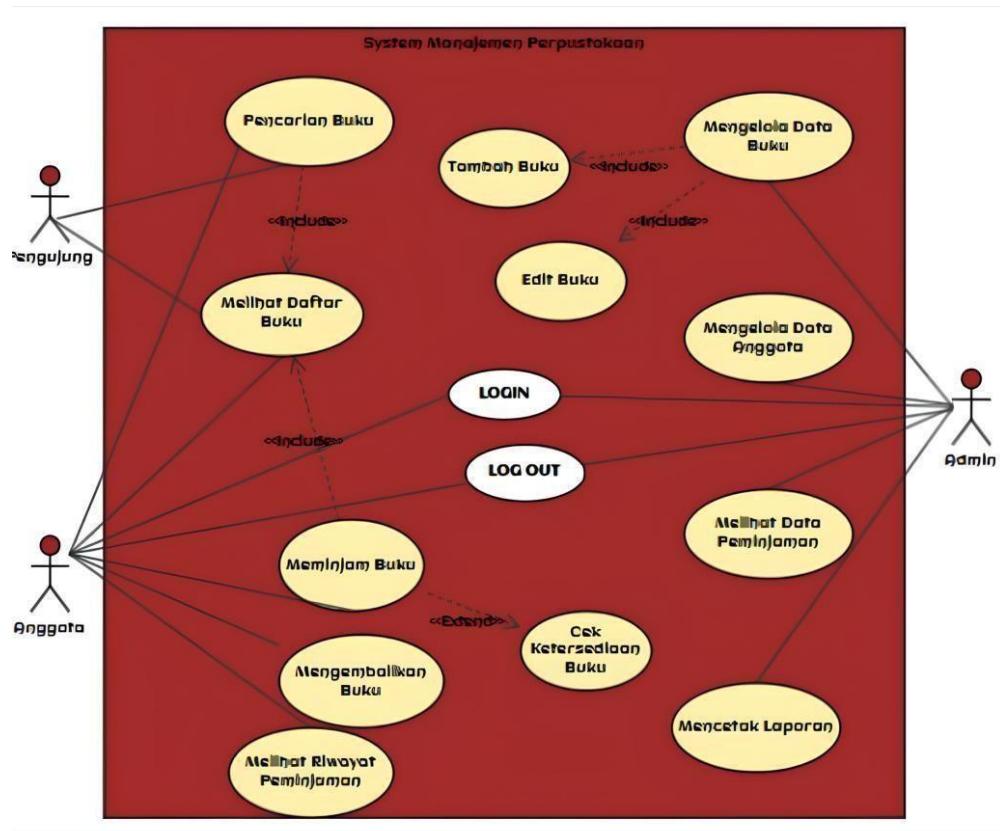
- Login
- Logout
- Kelola Data Buku (Tambah, Edit, Hapus)
- Kelola Data Anggota
- Proses Peminjaman Buku
- Proses Pengembalian Buku
- Melihat Riwayat Peminjaman
- Cetak Laporan

### B. Fungsi Anggota

- Login
- Logout
- Cari Buku
- Melihat Detail Buku
- Melihat Riwayat Peminjaman

## 2.3 Deskripsi Use Case Sistem

Use Case Diagram adalah diagram UML yang menggambarkan interaksi antara aktor (pengguna sistem) dan use case (fungsi-fungsionalitas sistem). Menurut (Sommerville, 2011), use case diagram berfungsi untuk memodelkan perspektif pengguna terhadap sistem dan menunjukkan bagaimana pengguna berinteraksi dengan fitur-fitur yang tersedia. Berdasarkan konsep tersebut, use case diagram Sistem Manajemen Perpustakaan ini dirancang untuk memvisualisasikan fungsi-fungsi utama sistem yang meliputi pencarian buku, manajemen koleksi, dan pengelolaan peminjaman. Diagram ini memperlihatkan bagaimana setiap aktor dapat mengakses fitur-fitur sistem sesuai dengan peran dan hak akses yang dimilikinya.

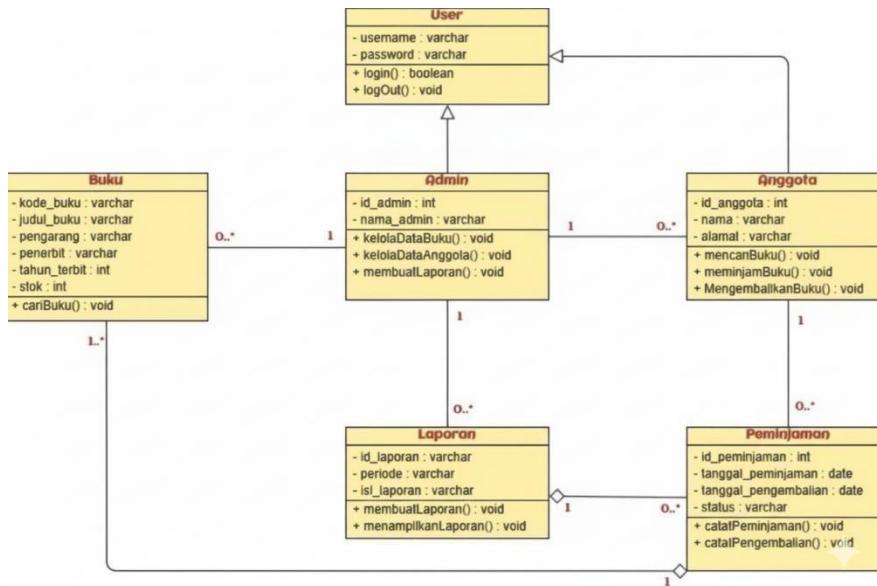


Gambar 1 : Use Case Diagram Sistem Manajemen Perpustakaan

Dalam sistem manajemen perpustakaan ini terdapat dua aktor utama, yaitu Pustakawan dan Anggota Perpustakaan. Pustakawan merupakan aktor utama yang memiliki hak akses penuh terhadap pengelolaan sistem perpustakaan. Melalui sistem, pustakawan dapat melakukan proses LOGIN untuk masuk ke dashboard yang berfungsi sebagai pusat pengelolaan koleksi buku dan peminjaman. Selain itu, pustakawan bertanggung jawab dalam melakukan Pencarian Buku yang mencakup melihat daftar buku dan data buku, Penambahan Buku baru dengan melibatkan data anggota dan data peminjaman, serta Edit Buku yang mencakup pengelolaan data buku dan data anggota. Sistem ini juga mendukung proses Peminjaman Buku dengan fitur extend untuk mengecek ketersediaan buku, serta Pengembalian Buku sebagai ekstensi dari proses peminjaman. Pustakawan dapat melihat riwayat peminjaman dan melakukan LOG OUT setelah menyelesaikan tugas. Sedangkan anggota perpustakaan memiliki akses terbatas untuk melakukan pencarian buku dan melihat riwayat peminjaman pribadi mereka.

### 3. Class Diagram

Class diagram adalah representasi visual yang menunjukkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas dalam model desain suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Jadi dapat kita anggap bahwa class diagram ini adalah tampilan dari struktur sistem program (Ramdany, Pengembangan Sistem Informasi Manajemen Perpustakaan Berbasis Web, 2024)



Gambar 2 : Class Diagram Sistem Manajemen Perpustakaan

Class diagram pada sistem perpustakaan di atas menggambarkan struktur tetap dari sistem dengan menampilkan enam kelas utama yaitu User, Admin, Anggota, Buku, Peminjaman, dan Laporan beserta atribut dan metode yang dimiliki masing-masing kelas, serta hubungan antar kelas tersebut. Diagram ini menjadi representasi dari rancangan logika program yang menggambarkan bagaimana data pengguna, admin, anggota, buku, transaksi peminjaman, dan laporan disimpan, diolah, dan saling terhubung antar bagian sistem. Setiap kelas memiliki atribut khas, seperti id, nama, atau data spesifik lainnya, dan metode untuk melakukan berbagai operasi sistem seperti login, pencarian buku, pengelolaan data, serta pencatatan transaksi peminjaman dan laporan. Relasi antar kelas terlihat pada pengelolaan data yang dilakukan admin terhadap buku, anggota, dan laporan, serta pencatatan transaksi peminjaman yang dilakukan oleh anggota dan direkam dalam laporan; semua interaksi ini memastikan integrasi dan alur informasi yang sistematis di dalam aplikasi perpustakaan berbasis objek.

#### 4. Penerapan Design Pattern atau Prinsip OOD (SOLID)

##### 4.1 Prinsip OOD ( SOLID )

SOLID adalah lima prinsip utama dalam desain perangkat lunak berorientasi objek yang membantu agar sistem lebih mudah dipelihara, fleksibel, dan efisien. Prinsip ini terdiri dari:

S – Single Responsibility Principle (SRP)

O – Open/Closed Principle (OCP)

L – Liskov Substitution Principle (LSP)

I – Interface Segregation Principle (ISP)

D – Dependency Inversion Principle (DIP)

Prinsip yang Paling Relevan untuk sistem manajemen perpustakaan : Single Responsibility Principle (SRP).

Dalam sistem perpustakaan, penerapan prinsip SRP bisa dilihat pada pemisahan tanggung jawab setiap kelas utama berikut :

Kelas : Tanggung Jawab Utama (Satu Fokus Kerja)

User : Menangani autentikasi (login, logout) dan identitas pengguna.

Anggota : Mengelola data pribadi anggota dan riwayat peminjaman.

Buku : Mengelola data koleksi buku, status ketersediaan, dan pencarian.

Peminjaman : Mencatat transaksi peminjaman dan pengembalian buku.

Laporan : Membuat dan menampilkan laporan aktivitas perpustakaan.

Contoh penerapannya :

Jika terjadi perubahan di proses login, maka hanya kelas User yang perlu diubah tidak akan berdampak pada Buku, Peminjaman, atau Laporan.

Ini menandakan sistem sudah sesuai prinsip SRP dan terstruktur dengan baik.

#### 4.2 Penerapan Tiga Creational Design Pattern

Untuk mendukung implementasi prinsip SOLID, sistem perpustakaan ini juga dapat menggunakan beberapa pola desain creational agar pembuatan objek lebih efisien dan fleksibel.

##### 1. Singleton Pattern

Tujuan : Menjamin hanya ada satu instance dari kelas tertentu (biasanya untuk koneksi database).

Penerapan : Digunakan pada kelas DatabaseConnection agar semua bagian sistem memakai koneksi yang sama.

Manfaat: Mencegah duplikasi koneksi dan meningkatkan efisiensi sistem.

##### 2. Factory Method Pattern

Tujuan : Membuat objek tanpa menentukan kelasnya secara langsung, sehingga kode lebih fleksibel.

Penerapan : Ketika sistem membuat objek pengguna berdasarkan jenis peran (Admin atau Anggota).

Manfaat : Memudahkan ekspansi sistem jika ada tipe pengguna baru, misalnya “Petugas Lapangan”.

##### 3. Builder Pattern

Tujuan : Membuat objek kompleks langkah demi langkah, tanpa membuat konstruktor yang rumit.

Penerapan : Digunakan dalam pembuatan laporan peminjaman, yang terdiri dari banyak komponen data (anggota, buku, tanggal, status, total, dsb).

Manfaat : Laporan bisa disusun secara modular dan mudah disesuaikan tanpa mengubah struktur utama kelas.

5. Membuat dokumentasi perancangan sistem manajemen perpustakaan dalam bentuk ARTIKEL ILMIAH (Jurnal)

**Jawaban No. 5 Tertera Pada Jurnal**