

LAPORAN PROYEK AKHIR
PEMROGRAMAN BERORIENTASI OBJEK
APPLICATION LIBRARY MANAGEMENT SYSTEM



Penyusun:

Wahyu Nur Hidayat	(2320506027)
Tania Adna Shafira	(2320506034)
Muhammad Surya Saputro	(2320506035)
Khoirunnisa Fitri Handayani	(2320506054)

PROGRAM STUDI S1 TEKNOLOGI INFORMASI
UNIVERSITAS TIDAR
GANJIL 2024/2025

BAB I

PENDAHULUAN

A. Latar Belakang

Perpustakaan adalah salah satu sumber informasi dan pengetahuan yang berperan penting dalam mendukung kegiatan belajar mengajar, penelitian, dan pengembangan ilmu pengetahuan. Namun, pengelolaan perpustakaan secara manual seringkali menimbulkan berbagai kendala, seperti kesulitan dalam pencatatan buku, pengelolaan data peminjam, dan pelacakan pengembalian buku. Kesalahan dalam pencatatan manual juga dapat menyebabkan data menjadi tidak akurat dan memperlambat proses pelayanan kepada pengguna. Oleh karena itu, diperlukan sebuah sistem yang dapat mengelola proses-proses tersebut secara otomatis dan efisien.

Aplikasi **Library Management System** berbasis Java hadir sebagai solusi untuk mengatasi permasalahan tersebut. Dengan menggunakan teknologi ini, proses pengelolaan data buku, peminjaman, dan pengembalian dapat dilakukan secara lebih terstruktur dan mudah diakses oleh admin maupun pengguna. Sistem ini juga dirancang dengan fitur-fitur yang memungkinkan admin untuk menambah, mencari, dan menghapus data buku secara langsung melalui antarmuka aplikasi. Selain itu, pengguna dapat memanfaatkan fitur pencarian dan peminjaman buku yang lebih cepat dan akurat. Implementasi aplikasi ini diharapkan dapat meningkatkan efisiensi operasional perpustakaan dan memberikan kemudahan bagi para penggunanya.

B. Rumusan Masalah

- Bagaimana merancang sistem manajemen perpustakaan yang efisien dan mudah digunakan oleh berbagai jenis pengguna?
- Bagaimana mengimplementasikan sistem login dan verifikasi peran yang aman, agar hanya pengguna yang berwenang dapat mengakses fitur-fitur tertentu seperti pengelolaan buku dan data pengguna?
- Bagaimana cara mengelola data buku, peminjaman, dan pengembalian secara digital sehingga dapat diakses secara real-time oleh pengguna dan admin?
- Bagaimana memastikan bahwa aplikasi dapat berjalan dengan lancar pada perangkat dengan spesifikasi yang berbeda, tanpa membebani performa sistem?

C. Tujuan

- Memenuhi tugas proyek akhir mata kuliah Pemrograman Berorientasi Objek.
- Mengembangkan aplikasi yang mampu mempermudah pengelolaan data buku dan pengguna perpustakaan.
- Memfasilitasi pengguna untuk mengakses informasi buku yang tersedia dan buku yang dipinjam secara cepat dan akurat.
- Mengurangi risiko kesalahan data dalam pencatatan peminjaman dan pengembalian buku.

BAB II

HASIL & PEMBAHASAN

A. Pengertian Aplikasi

Aplikasi *Library Management System* adalah sebuah perangkat lunak yang dirancang untuk mengelola proses peminjaman, pengembalian, dan manajemen data buku secara digital. Aplikasi ini membantu pengguna, baik admin maupun anggota perpustakaan, untuk melakukan berbagai operasi seperti melihat ketersediaan buku, menambah data buku, serta mengelola informasi pengguna.

Aplikasi ini memiliki dua jenis pengguna utama:

1. **Admin:** Bertanggung jawab dalam mengelola data buku, menambahkan atau menghapus buku, serta memantau aktivitas peminjaman.
2. **User (Anggota):** Dapat melihat daftar buku yang tersedia, meminjam buku, dan melihat buku yang sedang dipinjam.

Aplikasi ini juga memiliki kelebihan dan kekurangan:

1. Kelebihan

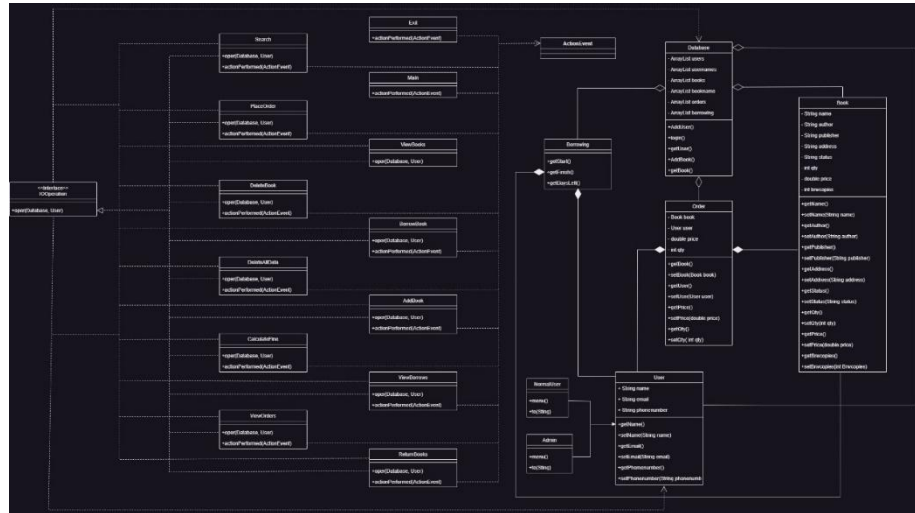
- **Efisiensi Pengelolaan Data:** Aplikasi ini memungkinkan pengelolaan data buku, peminjaman, dan pengembalian secara digital, sehingga mengurangi risiko kesalahan pencatatan manual.
- **Antarmuka User-Friendly:** Dengan antarmuka berbasis Java Swing, pengguna dapat dengan mudah menavigasi fitur-fitur yang ada, seperti menambah, mencari, dan meminjam buku.
- **Fitur Admin Lengkap:** Admin memiliki akses penuh untuk mengelola koleksi buku, termasuk menambah, menghapus, dan melihat daftar buku serta pesanan dan peminjaman.

2. Kekurangan

- **Tidak Terintegrasi dengan Database Online:** Aplikasi ini menggunakan database lokal, sehingga kurang fleksibel jika digunakan dalam jaringan perpustakaan besar yang membutuhkan akses data secara online.
- **Fitur Keamanan Terbatas:** Belum ada implementasi enkripsi data atau fitur autentikasi yang kompleks, sehingga keamanan data masih perlu ditingkatkan.

- **Tidak Mendukung Multi-User secara Real-Time:** Sistem ini tidak dirancang untuk penggunaan multi-user secara bersamaan, sehingga kurang optimal untuk perpustakaan dengan banyak pengguna aktif.

B. Diagram Kelas



1. Kelas Utama dan Relasi

- **IOOperation (Interface):**

Berfungsi sebagai antarmuka untuk berbagai operasi I/O yang akan dilakukan, seperti `open()` dan `actionPerformed()`. Interface ini diimplementasikan oleh beberapa kelas lain seperti `Search`, `PlaceOrder`, `DeleteBook`, dll.

- **Kelas-kelas Operasi:**

Terdapat beberapa kelas operasi diantaranya `Search`, `PlaceOrder`, `DeleteBook`, `AddBook`, `BorrowBook`, `ReturnBooks`, `ViewBooks`, `ViewBorrows`, `CalculateFine`, `ViewOrders`.

Setiap kelas ini memiliki metode `open()` dan `actionPerformed()` yang bertugas mengelola berbagai tindakan pada aplikasi. Mereka kemungkinan mengontrol bagaimana interaksi dengan database berlangsung berdasarkan operasi masing-masing.

2. Kelas untuk Data dan Entitas:

- **Database**

Menyimpan data pengguna (`ArrayList users`, `ArrayList usernames`), buku (`ArrayList bookname`), dan peminjaman serta pesanan (`ArrayList orders` dan `ArrayList borrowing`). Metode penting: `addUser()`, `addBook()`, `getInfo()`.

- **Book**

Berisi atribut buku seperti nama, penulis, alamat, status, harga, dan diskon. Metode getter dan setter digunakan untuk manipulasi data buku.

- **Order**

Menyimpan data pemesanan dengan atribut seperti buku, pengguna, harga, dan ID pesanan. Berhubungan dengan kelas **Book** dan **User**.

- **Borrowing**

Mengelola data peminjaman buku dengan atribut seperti tanggal mulai dan ID.

3. Kelas Pengguna

- **User**

Merupakan kelas induk yang menyimpan data dasar pengguna seperti nama, email, dan nomor telepon.

Admin dan **NormalUser** merupakan turunan dari kelas ini. Kelas **Admin** memiliki akses tambahan ke menu untuk administrasi, sedangkan **NormalUser** hanya memiliki akses ke fungsi dasar.

4. Relasi dan Asosiasi

- Kelas **Order** dan **Borrowing** memiliki relasi dengan kelas **Book** dan **User**.
- **Database** berfungsi sebagai pusat penyimpanan semua data terkait pengguna, buku, dan peminjaman/pemesanan.
- Semua operasi (seperti mencari buku atau meminjam buku) memanfaatkan data dari kelas **Database**.

C. Penjelasan Kode Program & Kelas

1. Kelas AddBook

Kelas AddBook adalah bagian dari sistem perpustakaan yang memungkinkan admin untuk menambahkan buku baru ke dalam database. Kelas ini menampilkan antarmuka grafis menggunakan JFrame, di mana pengguna dapat memasukkan detail buku seperti nama, penulis, penerbit, alamat koleksi, jumlah, harga, dan jumlah salinan yang dipinjam. Setelah validasi input, data buku akan dikonversi menjadi objek Book dan ditambahkan ke database menggunakan metode AddBook() pada objek database.

2. Kelas Admin

Kelas Admin adalah turunan dari kelas User, yang dirancang untuk memberikan akses ke berbagai operasi administratif seperti menambah, menghapus buku, mencari buku, melihat data peminjaman, dan mengelola pesanan. Kelas ini memiliki metode `adminMenu()` yang menampilkan menu operasi admin menggunakan antarmuka grafis. Dengan operasi yang terstruktur dalam array operations, kelas ini memungkinkan admin untuk mengelola perpustakaan secara komprehensif.

3. Kelas Book

Kelas Book merepresentasikan entitas buku dalam sistem perpustakaan, dengan atribut seperti nama buku, penulis, penerbit, alamat koleksi, jumlah stok (qty), harga, dan jumlah salinan yang dapat dipinjam (brwcopies). Kelas ini menyediakan konstruktor default dan parameter, serta metode getter dan setter untuk mengakses dan memodifikasi atribut buku. Kelas ini juga memiliki dua metode `toString()`, yang pertama untuk menampilkan informasi buku dalam format yang mudah dibaca, dan yang kedua (`toString2()`) untuk format penyimpanan data.

4. Kelas BorrowBook

Kelas BorrowBook mengimplementasikan antarmuka `IOOperation` untuk mengatur proses peminjaman buku dalam sistem perpustakaan. Kelas ini menggunakan antarmuka grafis berbasis `JFrame` dengan input pengguna berupa nama buku. Jika buku tersedia dan belum dipinjam sebelumnya oleh pengguna, buku akan dipinjam, dan jumlah salinan yang dapat dipinjam (brwcopies) berkurang. Notifikasi juga ditampilkan terkait batas waktu pengembalian. Sebaliknya, jika buku tidak tersedia atau sudah dipinjam, pengguna diberi peringatan dengan pesan dialog.

5. Kelas Borrowing

Kelas Borrowing menangani proses peminjaman buku dengan menyimpan informasi terkait tanggal peminjaman, tanggal pengembalian, sisa hari hingga batas pengembalian, serta objek buku dan pengguna yang meminjam. Kelas ini memiliki dua konstruktor, satu untuk inisialisasi otomatis dengan tanggal peminjaman hari ini dan tanggal pengembalian 14 hari setelahnya, serta satu lagi untuk memungkinkan pengaturan tanggal mulai dan tanggal akhir secara manual. Tanggal-tanggal ini diformat menggunakan `DateTimeFormatter`, dan jumlah hari yang tersisa dihitung

dengan `Period.between()`. Kelas ini juga menyediakan berbagai metode untuk mengambil dan mengubah informasi terkait peminjaman, seperti `getStart()`, `getFinish()`, `getDaysLeft()`, serta dua versi metode `toString()` untuk menampilkan informasi peminjaman dalam format yang berbeda. Kelas ini memfasilitasi pengelolaan data peminjaman buku di sistem manajemen perpustakaan.

6. Kelas CalculateFine

Kelas `CalculateFine` adalah implementasi dari interface `IOOperation` yang menyediakan antarmuka untuk menghitung denda keterlambatan pengembalian buku. Kelas ini membuat jendela dengan input untuk nama buku dan tombol untuk menghitung denda atau membatalkan. Ketika tombol "Calculate Fine" ditekan, sistem akan memeriksa apakah buku yang dipinjam terlambat dan menghitung dendanya (50 per hari keterlambatan). Jika buku tidak ditemukan dalam daftar peminjaman, atau jika pengguna tidak terlambat, pesan yang sesuai akan ditampilkan. Jika tombol "Cancel" ditekan, jendela akan ditutup.

7. Kelas Database

Kelas `Database` bertugas untuk mengelola data pengguna, buku, pemesanan, dan peminjaman di sistem manajemen perpustakaan. Kelas ini menyediakan metode untuk menambah dan mencari data pengguna serta buku, menyimpan perubahan data ke file, dan mengelola pemesanan serta peminjaman buku. Data pengguna disimpan dalam file `Users`, data buku dalam file `Books`, pemesanan dalam file `Orders`, dan peminjaman dalam file `Borrowings`. Setiap kali ada perubahan pada data, seperti penambahan pengguna atau buku, perubahan tersebut disimpan secara otomatis ke file terkait. Kelas ini juga memastikan bahwa file dan folder yang diperlukan ada saat aplikasi dijalankan.

8. Kelas DeleteAllData

Kelas `DeleteAllData` bertanggung jawab untuk menampilkan konfirmasi kepada pengguna sebelum menghapus seluruh data dari sistem. Ketika metode `oper` dipanggil, kelas ini akan menampilkan jendela dengan pesan konfirmasi yang bertanya apakah pengguna benar-benar ingin menghapus semua data. Terdapat dua tombol di jendela tersebut: tombol "Continue" untuk melanjutkan penghapusan data, dan tombol "Cancel" untuk membatalkan aksi. Jika pengguna memilih "Continue", seluruh data dihapus dengan memanggil metode `deleteAllData` dari objek `database`, kemudian

aplikasi akan keluar. Jika pengguna memilih "Cancel", jendela ditutup dan pengguna dikembalikan ke menu utama.

9. Kelas DeleteBook

Kelas DeleteBook berfungsi untuk menghapus buku dari database berdasarkan nama buku yang diberikan pengguna. Ketika metode `oper` dijalankan, jendela akan muncul dengan form input untuk memasukkan nama buku yang akan dihapus. Jika nama buku tidak diisi, akan muncul pesan peringatan. Setelah pengguna mengisi nama buku dan menekan tombol "Delete Book", sistem akan mencari buku tersebut di database. Jika buku ditemukan, buku akan dihapus dan pesan konfirmasi akan ditampilkan. Jika buku tidak ada, pesan kesalahan akan muncul. Tombol "Cancel" akan menutup jendela tanpa melakukan aksi apa pun.

10. Kelas Exit

Kelas Exit merupakan implementasi dari interface `IOOperation`, yang menangani proses login dan registrasi pada sistem manajemen perpustakaan. Di dalam kelas ini, terdapat beberapa komponen grafis menggunakan `JFrame`, seperti label, teks, dan tombol yang memungkinkan pengguna untuk login sebagai admin atau pengguna biasa, serta melakukan registrasi dengan validasi input tertentu seperti hanya angka untuk nomor telepon dan hanya huruf untuk nama. Fungsi `login()` memverifikasi apakah pengguna terdaftar, sedangkan `registerAdmin()` dan `registerUser()` masing-masing menangani proses pendaftaran admin dan pengguna dengan validasi yang ketat. Setelah login atau registrasi berhasil, pengguna atau admin diarahkan ke menu utama sesuai peran mereka.

11. Kelas IOOperation

Kelas `IOOperation` adalah sebuah interface yang mendefinisikan kontrak untuk operasi input/output dalam sistem perpustakaan. Interface ini memiliki satu metode abstrak, yaitu `oper()`, yang menerima parameter `Database` dan `User`. Metode ini digunakan oleh kelas lain untuk menjalankan operasi tertentu seperti login, registrasi, atau fungsi lainnya yang melibatkan interaksi dengan pengguna dan database. Dengan menggunakan interface, berbagai kelas dapat mengimplementasikan metode ini dengan cara yang sesuai dengan fungsionalitas masing-masing.

12. Kelas Main

Program ini adalah sistem manajemen perpustakaan menggunakan Java Swing untuk antarmuka grafis. Dimulai dengan layar login yang meminta nomor telepon dan kata sandi pengguna. Terdapat tombol untuk login, pendaftaran sebagai admin, dan pendaftaran pengguna biasa. Fitur validasi diterapkan pada input nomor telepon (hanya angka) dan nama (hanya huruf). Jika login berhasil, pengguna diarahkan ke menu utama. Admin dapat membuat akun baru dengan memasukkan kode admin yang valid, sedangkan pengguna biasa dapat membuat akun tanpa kode admin. Database digunakan untuk menyimpan data pengguna dan memeriksa apakah nama pengguna sudah ada.

13. Kelas NormalUser

Kelas NormalUser adalah turunan dari kelas User yang mewakili pengguna biasa dalam sistem manajemen perpustakaan. Kelas ini memiliki dua konstruktor: satu untuk inisialisasi nama pengguna dan satu lagi untuk inisialisasi nama, email, dan nomor telepon pengguna. Kedua konstruktor ini menginisialisasi operasi yang dapat dilakukan oleh pengguna, seperti melihat buku, mencari buku, memesan, meminjam, menghitung denda, mengembalikan buku, dan keluar. Metode menu digunakan untuk menampilkan menu pengguna, memanggil metode userMenu dari kelas induk, dan metode toString untuk mengembalikan informasi pengguna dalam format tertentu.

14. Kelas Order

Kelas Order mewakili pemesanan buku dalam sistem perpustakaan. Kelas ini memiliki empat atribut: book (buku yang dipesan), user (pengguna yang melakukan pemesanan), price (harga buku), dan qty (jumlah buku yang dipesan). Terdapat dua konstruktor: satu untuk inisialisasi nilai atribut, dan satu lagi konstruktor default tanpa parameter. Kelas ini juga menyediakan metode getter dan setter untuk setiap atributnya. Metode toString mengembalikan informasi pemesanan dalam format yang mudah dibaca, sedangkan toString2 mengembalikan informasi pemesanan dalam format yang lebih sederhana dan terpisah dengan tanda <N/>.

15. Kelas PlaceOrder

Kelas PlaceOrder mengimplementasikan interface IOOperation untuk menangani operasi pemesanan buku dalam sistem perpustakaan. Pada metode oper, kelas ini

membuat antarmuka grafis untuk memungkinkan pengguna melakukan pemesanan buku, termasuk memasukkan nama buku dan jumlah buku yang ingin dipesan. Jika pengguna tidak mengisi salah satu kolom atau memberikan input yang salah (seperti memasukkan jumlah yang bukan angka), akan muncul pesan kesalahan. Setelah validasi input, kelas ini memeriksa apakah buku yang diminta tersedia dalam database. Jika ada, objek Order dibuat, harga dihitung berdasarkan jumlah, dan jumlah buku yang tersedia diperbarui. Pemesanan kemudian ditambahkan ke database dan pengguna diberi tahu bahwa pemesanan berhasil. Jika terjadi kesalahan, misalnya buku tidak ada atau jumlah yang diminta tidak valid, pesan kesalahan akan ditampilkan. Tombol "Cancel" akan menutup jendela tanpa melakukan pemesanan.

16. Kelas ReturnBook

Kelas ReturnBook mengimplementasikan interface IOOperation dan berfungsi untuk mengelola proses pengembalian buku. Pada metode oper, kelas ini membuat antarmuka grafis yang memungkinkan pengguna memasukkan nama buku yang ingin dikembalikan. Setelah nama buku dimasukkan, sistem memverifikasi apakah buku tersebut benar-benar dipinjam oleh pengguna yang bersangkutan. Jika buku yang dimaksud ada dalam daftar peminjaman pengguna, sistem memeriksa apakah ada denda yang harus dibayar jika buku terlambat dikembalikan. Jika tidak ada masalah, jumlah salinan buku yang tersedia diperbarui, dan buku tersebut dikembalikan ke database. Jika buku tidak dipinjam oleh pengguna atau terdapat kesalahan lainnya, pesan kesalahan akan muncul. Jika pengguna menekan tombol "Cancel", jendela akan ditutup tanpa melakukan tindakan lebih lanjut.

17. Kelas Search

Kelas Search mengimplementasikan interface IOOperation dan digunakan untuk mencari buku berdasarkan nama. Pada metode oper, kelas ini membuat antarmuka grafis yang memungkinkan pengguna memasukkan nama buku yang ingin dicari. Jika pengguna memasukkan nama buku, sistem akan mencari buku tersebut di dalam database. Jika buku ditemukan, informasi buku akan ditampilkan dalam bentuk pesan dialog. Jika buku tidak ditemukan, pesan kesalahan akan muncul. Jika pengguna menekan tombol "Cancel", jendela pencarian akan ditutup tanpa melakukan pencarian lebih lanjut.

18. Kelas User

Kelas User adalah kelas abstrak yang mendefinisikan pengguna dalam sistem manajemen perpustakaan dengan atribut name, email, dan phonenumber, serta array operations yang berisi operasi yang dapat dilakukan oleh pengguna. Kelas ini memiliki konstruktor untuk menginisialisasi data pengguna dan metode userMenu() untuk menampilkan menu utama berisi pilihan seperti "View Books", "Search", "Place Order", dll. Metode frame() digunakan untuk membuat antarmuka grafis dengan tombol-tombol menu yang, ketika diklik, akan menjalankan operasi terkait. Kelas ini berfungsi sebagai dasar untuk kelas pengguna lain dengan operasi yang spesifik.

19. Kelas ViewBooks

Kelas ViewBooks mengimplementasikan antarmuka IOOperation dan bertanggung jawab untuk menampilkan daftar buku yang ada di dalam database. Pada metode oper(), kelas ini membuat sebuah jendela JFrame dengan tabel yang menampilkan informasi buku seperti nama, pengarang, penerbit, kategori, jumlah, harga, dan jumlah salinan yang dipinjam. Setiap buku ditampilkan dalam baris dengan label yang dibungkus dalam JLabel, dan ditata menggunakan GridLayout. Kelas ini memungkinkan pengguna melihat detail buku yang tersedia dalam sistem perpustakaan.

20. Kelas ViewBorrows

Kelas ViewBorrows mengimplementasikan antarmuka IOOperation dan digunakan untuk menampilkan data peminjaman buku dari database. Metode oper() membuat tampilan dengan kolom input untuk nama buku dan dua tombol, yaitu untuk melihat peminjaman buku dan membatalkan. Jika buku ada, kelas ini akan memanggil metode viewBorrows() untuk menampilkan daftar peminjaman yang terkait dengan buku tersebut, meliputi nama buku, nama pengguna, tanggal peminjaman, dan tanggal pengembalian. Data peminjaman ditampilkan dalam tabel dengan layout GridLayout. Jika buku tidak ditemukan, sebuah pesan kesalahan akan ditampilkan.

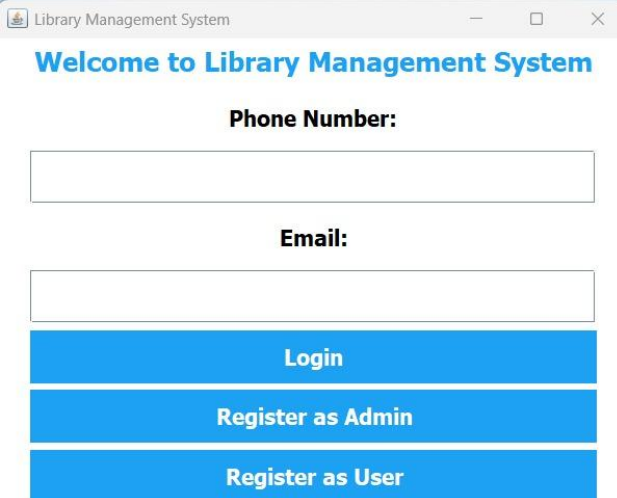
21. Kelas ViewOrders

Kelas ViewOrders mengimplementasikan antarmuka IOOperation dan digunakan untuk menampilkan data pemesanan buku dalam aplikasi. Metode oper() membuat

tampilan dengan kolom input untuk nama buku dan dua tombol: untuk melihat pemesanan buku dan membatalkan. Jika nama buku valid dan ada dalam database, metode `viewOrders()` akan dipanggil untuk menampilkan daftar pemesanan buku yang terkait, termasuk nama buku, nama pengguna, jumlah pemesanan, dan harga. Data ditampilkan dalam bentuk tabel menggunakan layout `GridLayout`. Jika buku tidak ditemukan, pesan kesalahan akan ditampilkan.

D. Hasil Aplikasi

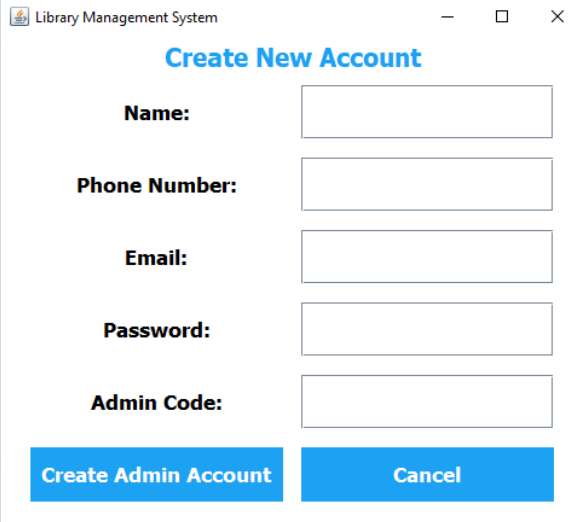
1. Halaman Utama



The screenshot shows a web browser window titled "Library Management System". The page has a blue header with the text "Welcome to Library Management System". Below the header, there are two input fields: "Phone Number:" and "Email:". Below these fields are three blue buttons: "Login", "Register as Admin", and "Register as User".

- Halaman utama memungkinkan pengguna untuk login dengan memasukkan nomor telepon dan email untuk mengakses aplikasi. Halaman utama juga memungkinkan pengguna untuk mendaftar akun sebagai admin atau user.

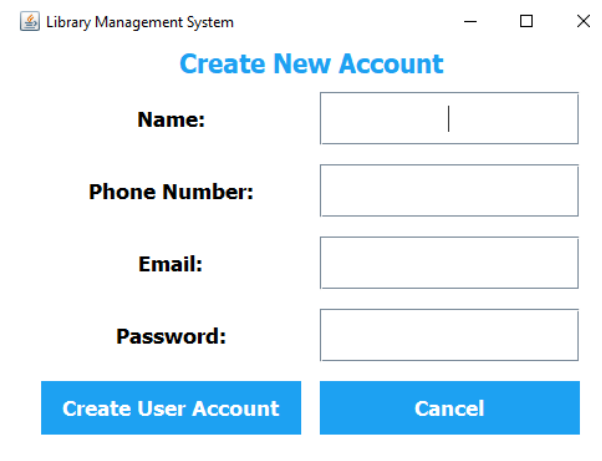
2. Tampilan Pembuatan Akun Admin



The screenshot shows a web browser window titled "Library Management System". The page has a blue header with the text "Create New Account". Below the header, there are five input fields: "Name:", "Phone Number:", "Email:", "Password:", and "Admin Code:". Below these fields are two blue buttons: "Create Admin Account" and "Cancel".

- Admin dapat mendaftar akun dengan memasukkan nama, nomor hp, email, password, dan admin code. Admin code hanya dapat diisi dengan “ADMIN123” yang hanya diketahui admin perpustakaan.

3. Tampilan Pembuatan Akun User



Library Management System

Create New Account

Name:

Phone Number:

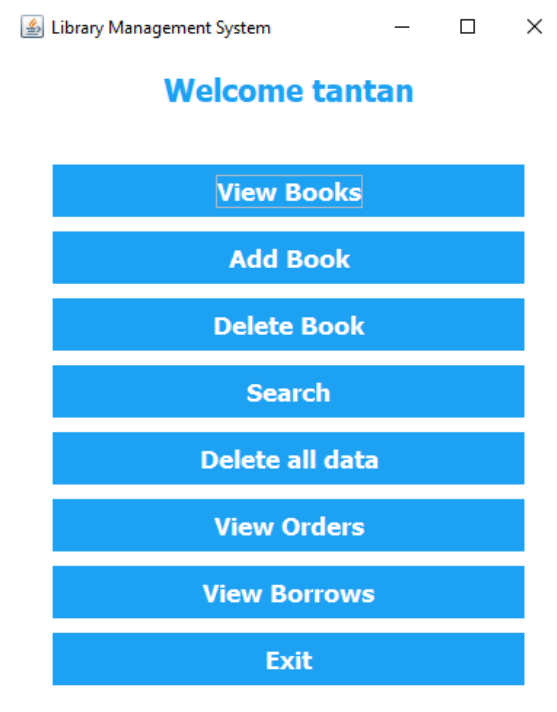
Email:

Password:

Create User Account **Cancel**

- User atau pelanggan perpustakaan dapat membuat atau mendaftar akun dengan memasukkan nama, nomer hp, email dan password.

4. Tampilan Dashboard Admin



Library Management System

Welcome tantan

View Books

Add Book

Delete Book

Search

Delete all data

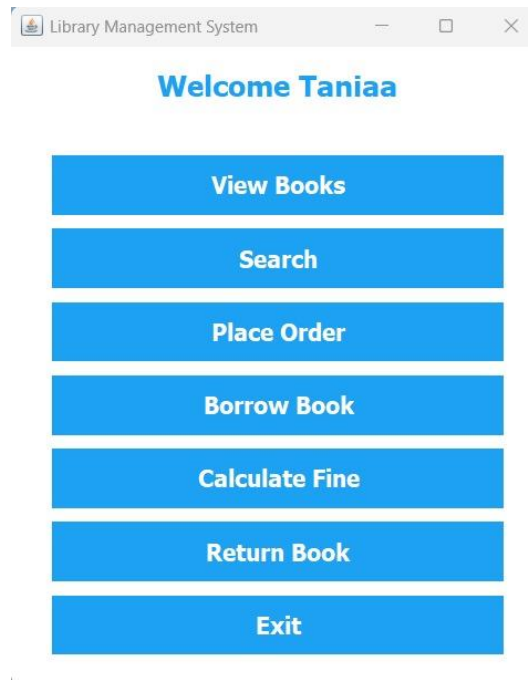
View Orders

View Borrows

Exit

- Admin dapat mengelola data buku melalui menu ini, dengan opsi untuk menambahkan atau menghapus buku. Admin juga bisa melihat data pembelian dan peminjaman buku, juga bisa melihat buku yang tersedia di perpustakaan.

5. Tampilan Dashboard User



- User dapat melihat dan mencari buku yang tersedia di perpustakaan. User juga dapat membeli, meminjam, dan mengembalikan buku menggunakan sistem ini. User juga bisa melihat data peminjaman buku jika melami keterlambatan pengembalian buku maka akan kena denda, dan bisa di akses di menu calculate fine.

6. Tampilan View Books

Library Management System						
View Books						
Name	Author	Publisher	CLA	Qty	Price	Brw cps
Programming	John Smith	Tech Press	Rak A, Baris 1	37	100000.0	21
Machine Learn	Sarah Davis	AI Publisher	Rak B, Baris 2	18	110000.0	7
Web Creation	Brian Clark	WebTech	Rak C, Baris 1	11	115000.0	5

- User dapat melihat koleksi buku perpustakaan dengan menekan tombol "View Books", usr dapat melihat daftar buku yang tersedia lengkap dengan detail seperti judul, pengarang, penerbit, lokasi di rak (CLA), jumlah buku yang ada (Qty), harga buku (Price), dan jumlah buku yang sedang dipinjam (Brw cps). Tampilan tabel ini

memudahkan user untuk mencari dan mendapatkan informasi mengenai setiap buku dalam koleksi perpustakaan.

7. Tampilan View Borrows

Library Management System			
View Borrows			
Book	User	Borrow Date	Return Date
Web Creation	Surya	2024-12-01	2024-12-15

- User dapat melihat informasi lengkap mengenai riwayat peminjaman buku. Dengan mengklik tombol "View Borrows", user dapat melihat daftar buku yang sedang dipinjam. Tabel yang ditampilkan berisi detail setiap transaksi peminjaman, termasuk judul buku, nama peminjam, tanggal peminjaman, dan tanggal pengembalian yang telah ditentukan. Fitur ini sangat berguna untuk melacak status peminjaman, mengelola pengembalian buku, dan membuat laporan terkait aktivitas peminjaman di perpustakaan.

8. Tampilan View Order

Library Management System			
View Orders			
Book	User	Qty	Price
Web Creati...	Wahyu	2	230000.0

- User dapat melihat daftar lengkap semua pesanan buku yang telah dilakukan oleh user perpustakaan dengan mengklik tombol "View Orders", user dapat melihat detail setiap pesanan, termasuk judul buku yang dipesan, nama pemesan, jumlah buku yang dipesan, dan total harga yang harus dibayarkan. Fitur ini sangat berguna untuk memantau status pesanan, mengelola proses pemesanan, dan menghasilkan laporan penjualan buku di perpustakaan.

BAB III

KESIMPULAN & SARAN

A. Kesimpulan

Pengembangan aplikasi Library Management System berbasis Java telah berhasil memberikan solusi dalam pengelolaan data buku, peminjaman, dan pengembalian secara lebih efektif dan efisien. Aplikasi ini dilengkapi dengan fitur login untuk pengguna dengan peran admin dan user, pencatatan buku, serta proses peminjaman dan pengelolaan data yang terstruktur. Dengan adanya sistem ini, diharapkan perpustakaan dapat meningkatkan pelayanan, mengurangi kesalahan pencatatan manual, serta mempermudah akses pengguna dalam mencari dan meminjam buku. Selain itu, peran admin dapat mengelola koleksi buku dengan lebih optimal melalui fitur-fitur seperti penambahan, penghapusan, dan pencarian buku.

B. Saran

Untuk pengembangan lebih lanjut, aplikasi ini dapat ditingkatkan dengan menambahkan fitur notifikasi pengembalian buku agar pengguna dapat diingatkan sebelum jatuh tempo. Selain itu, Pengembangan antarmuka pengguna yang lebih interaktif dan intuitif juga dapat meningkatkan pengalaman pengguna saat mengoperasikan aplikasi.

DAFTAR PUSTAKA

- Aini, N. (2020). *Manajemen Perpustakaan Digital*. Jakarta: Pustaka Ilmu.
- Rahmawati, D. (2019). "Pengaruh Sistem Informasi Terhadap Efisiensi Pengelolaan Perpustakaan." *Jurnal Teknologi Informasi*, 7(2), 123-130.
- Nugroho, R. (2021). *Pengembangan Aplikasi Berbasis Java*. Yogyakarta: Graha Ilmu.