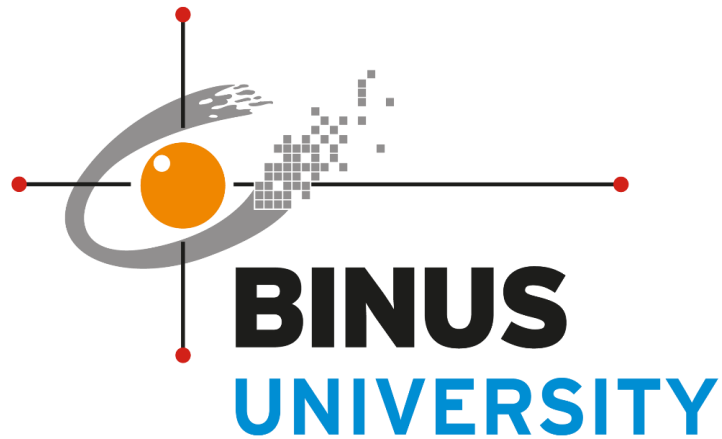


**PROJECT DESCRIPTION**  
**ASSIGNMENT PROJECT**  
**OBJECT ORIENTED PROGRAMMING**



**Kelas LE01**

**DISUSUN OLEH:**

2602190085	Hardi Kusuma
2602076323	Ivan Adrian
2602187815	Wairanatha Halim

**COMPUTER SCIENCE**  
**BINA NUSANTARA UNIVERSITY**  
**JAKARTA**  
**2023**

## **KATA PENGANTAR**

Dengan segala puji dan syukur kami panjatkan kepada Tuhan yang Maha Esa karena atas rahmat dan anugerah-Nya, kami dapat menyelesaikan laporan deskripsi proyek ini dengan tepat waktu dan memenuhi persyaratan yang diminta. Tujuan utama penulisan laporan deskripsi proyek ini ditujukan dalam rangka memenuhi Tugas Proyek mata kuliah Object Oriented Programming.

Laporan deskripsi proyek ini disusun untuk memberikan pemahaman tentang source code dan program yang kami buat yaitu Book Library Application. Di dalam laporan ini dijelaskan juga mengenai pendahuluan dan deskripsi proyek, fitur-fitur yang ada di dalam aplikasi, penerapan multi threading, dan juga relation antara class-class yang ada.

Semoga laporan deskripsi proyek ini dapat memberikan wawasan yang lebih luas dan menjadi sumbangan pemikiran kepada pembaca baik mahasiswa Universitas Bina Nusantara ataupun masyarakat.. Terlepas dari semua itu, kami menyadari sepenuhnya bahwa masih ada kekurangan baik dari segi perincian deskripsi maupun tata bahasa yang digunakan. Oleh karena itu, kami menerima segala saran dan kritik dari dosen pembimbing maupun pembaca lain agar kami dapat memperbaiki isi dan sistematika dari laporan deskripsi proyek ini.

Jakarta, 23 Desember 2023

Penyusun

## DAFTAR ISI

<b>KATA PENGANTAR .....</b>	<b>1</b>
<b>DAFTAR ISI.....</b>	<b>2</b>
<b>CHAPTER I .....</b>	<b>3</b>
<b>INTRODUCTION &amp; PROJECT DESCRIPTION .....</b>	<b>3</b>
1.1 Latar Belakang.....	3
1.2 Rumusan Masalah .....	4
1.3 Tujuan Pembuatan Aplikasi.....	4
<b>CHAPTER II .....</b>	<b>6</b>
<b>APPLICATION FEATURE &amp; CONSOLE APPLICATION .....</b>	<b>6</b>
2.1 Login/Logout: .....	6
2.2 Book Borrowing:.....	7
2.3 View Book List:.....	9
2.4 View Borrowed List:.....	9
2.5 View Customer List: .....	9
2.6 Add Book:.....	10
2.7 Add Customer: .....	11
2.8 Return Book: .....	13
2.9 Delete Book:.....	15
2.9.1 Delete Customer: .....	16
2.9.2 Exit: .....	18
<b>CHAPTER III .....</b>	<b>19</b>
<b>MULTI THREADING PART .....</b>	<b>19</b>
3.1 Add Book.....	19
3.2 Add Customer .....	20
<b>CHAPTER IV .....</b>	<b>23</b>
<b>RELATION BETWEEN CLASS.....</b>	<b>23</b>
4.1 User Class:.....	23
4.2 Admin Class: .....	23
4.3 Customer Class:.....	23
4.4 Book Class: .....	23
4.5 MainClass (Library System):.....	24
Class Relation .....	24
1. Pewarisan (Inheritance):.....	24
2. Komposisi (Composition):.....	24
3. Penggunaan objek-objek kelas lain di MainClass (Association):.....	25
<b>CHAPTER V .....</b>	<b>27</b>
<b>CONCLUSION.....</b>	<b>27</b>

# CHAPTER I

## INTRODUCTION & PROJECT DESCRIPTION

### 1.1 Latar Belakang

Book Library Application merupakan inovasi digital yang dirancang khusus untuk mempermudah tugas administrator perpustakaan atau librarian di lingkungan Universitas Bina Nusantara. Aplikasi ini memberikan solusi terpadu dengan fitur-fitur seperti login, logout, peminjaman buku, melihat daftar buku, melihat daftar pelanggan, menambahkan buku, menambahkan pelanggan, mengembalikan buku, menghapus buku, dan menghapus pelanggan. Pengembangan aplikasi ini mempertimbangkan kebutuhan unik di lingkungan perpustakaan universitas, di mana mahasiswa perlu berkomunikasi langsung dengan librarian untuk melakukan peminjaman buku.

Sebelum adanya Book Library Application, perpustakaan di Universitas Bina Nusantara menghadapi tantangan signifikan dalam administrasi peminjaman buku. Librarian harus mencatat setiap peminjaman secara manual, menciptakan potensi kesalahan yang dapat merugikan integritas data. Proses manual ini tidak hanya memakan waktu yang cukup besar, tetapi juga meningkatkan risiko terjadinya kesalahan penginputan atau pencatatan yang dapat berdampak pada akurasi informasi peminjaman.

Rentang terjadinya kesalahan dalam pencatatan peminjaman buku menjadi salah satu permasalahan utama. Kesalahan ini bisa mencakup kesalahan penulisan nomor identifikasi buku, nama peminjam, atau tanggal peminjaman. Hal ini tidak hanya menghambat efisiensi operasional, tetapi juga dapat menyulitkan dalam pelacakan dan pemahaman mengenai status peminjaman buku. Dengan keterbatasan sistem manual ini, perpustakaan berisiko kehilangan data yang penting, dan proses administrasi menjadi rentan terhadap ketidakakuratan.

Book Library Application dibuat dengan tujuan utama meningkatkan efisiensi dan akurasi dalam proses peminjaman buku. Dengan fitur-fitur canggih seperti multi-threading dan penggunaan kelas relasi, aplikasi ini membantu menyederhanakan dan mempercepat proses administrasi, termasuk penambahan dan penghapusan buku serta pelanggan. Keamanan data juga diperkuat dengan memberikan hak akses terbatas hanya kepada administrator perpustakaan.

Dalam konteks ini, kehadiran Book Library Application menjadi langkah progresif dalam meningkatkan efisiensi dan keterjangkauan administrasi perpustakaan di Universitas Bina Nusantara. Dengan merampingkan proses peminjaman buku dan memanfaatkan teknologi untuk pengelolaan data, aplikasi ini tidak hanya mempercepat respons terhadap kebutuhan mahasiswa tetapi juga memberikan administrator perpustakaan alat yang kuat untuk memantau dan mengelola koleksi buku dengan lebih efektif. Dengan mengintegrasikan teknologi modern, proyek ini bertujuan menciptakan pengalaman peminjaman buku yang lebih efisien dan memastikan bahwa perpustakaan di universitas tetap menjadi sumber pengetahuan yang dapat diakses dengan mudah dan cepat.

## **1.2 Rumusan Masalah**

1. Bagaimana proses pencatatan manual dalam administrasi peminjaman buku dilakukan di perpustakaan Universitas Bina Nusantara?
2. Mengapa proses manual tersebut rentan terhadap kesalahan penginputan data seperti nomor identifikasi buku, nama peminjam, dan tanggal peminjaman?
3. Bagaimana waktu yang dibutuhkan dalam pencatatan manual dapat mempengaruhi efisiensi operasional perpustakaan dan memperlambat respon terhadap permintaan peminjaman?
4. Mengapa sistem administrasi perpustakaan saat ini rentan kehilangan data terkait peminjaman buku dan sulit dilacak status peminjaman secara akurat?
5. Bagaimana kesalahan dalam pencatatan manual dapat mempengaruhi pemahaman pengguna terhadap status peminjaman, dan mengapa hal ini menjadi isu penting bagi pengguna perpustakaan?

## **1.3 Tujuan Pembuatan Aplikasi**

1. Meningkatkan Efisiensi: Mengotomatisasi proses pencatatan peminjaman buku untuk mempercepat respons terhadap permintaan peminjaman dan pengembalian.

2. Mengurangi Kesalahan: Mengimplementasikan sistem digital untuk mengurangi risiko kesalahan penginputan data seperti nomor identifikasi buku, nama peminjam, dan tanggal peminjaman.
3. Optimasi Waktu: Mengurangi waktu yang diperlukan dalam administrasi peminjaman buku, memastikan efisiensi operasional yang lebih baik.
4. Meningkatkan Akurasi Data: Memastikan keakuratan informasi terkait peminjaman buku untuk mendukung manajemen koleksi yang lebih efektif.
5. Mempercepat Tanggapan terhadap Perubahan Koleksi: Memungkinkan pencatatan perubahan koleksi buku dengan cepat dan efisien.
6. Memberikan Keterjangkauan Informasi: Memberikan kemudahan dalam pelacakan status peminjaman buku dan akses cepat terhadap informasi terkait.
7. Meningkatkan Kepuasan Pengguna: Memberikan pengalaman peminjaman buku yang lebih cepat, akurat, dan efisien bagi pengguna perpustakaan.

## CHAPTER II

### APPLICATION FEATURE & CONSOLE APPLICATION

#### 2.1 Login/Logout:

```
Book Library Admin
=====
1. Login
10. Exit
```

Tampilan ketika salah memilih menu:

```
Book Library Admin
=====
1. Login
11. Exit
12
Invalid option. Please try again.
Book Library Admin
=====
1. Login
11. Exit
a
Something went wrong.
Invalid option. Please try again.
Book Library Admin
=====
1. Login
11. Exit
```

1. Pengguna dapat Login dengan username dan password valid yang diinsert langsung oleh developer aplikasi melalui source code.

```
Book Library Admin
=====
1. Login
11. Exit
1
Enter username [default username: admin]: admin
Enter password [default password: password]: password
```

2. Setelah Login berhasil, sistem menampilkan menu untuk memberikan akses ke fitur eksklusif admin.

```
Book Library Admin
=====
1. Logout
2. Book Borrowing
3. View Book List
4. View Customer List
5. Add Book
6. Add Customer
7. Return Book
8. Delete Book
9. Delete Customer
10. View Borrowed Books
11. Exit
```

Jika salah menginput username dan password:

```

Book Library Admin
=====
1. Login
11. Exit
1
Enter username [default username: admin]: admin
Enter password [default password: password]: password
Login failed.

```

### 3. Logout akan mengembalikan pengguna ke menu utama

```

Book Library Admin
=====
1. Logout
2. Book Borrowing
3. View Book List
4. View Customer List
5. Add Book
6. Add Customer
7. Return Book
8. Delete Book
9. Delete Customer
10. View Borrowed Books
11. Exit
1
Logout successful.

```

## 2.2 Book Borrowing:

1. Admin dapat memulai proses peminjaman buku untuk seorang pelanggan.
2. Memerlukan ID pelanggan dan ID buku.

```

Book Library Admin
=====
1. Logout
2. Book Borrowing
3. View Book List
4. View Customer List
5. Add Book
6. Add Customer
7. Return Book
8. Delete Book
9. Delete Customer
10. View Borrowed Books
11. Exit
2
Enter customer ID: C001
Enter book ID: B001
Book successfully borrowed.
Press Enter key to continue...

```

3. Memeriksa apakah pelanggan ada dan apakah buku yang diminta tersedia.

Jika customer ID ditemukan dan Buku dengan book ID yang dimasukan tersedia:

```

Enter customer ID: C001
Enter book ID: B001
Book successfully borrowed.
Press Enter key to continue...

```

Jika customer ID tidak ditemukan maka:



Enter customer ID: C011  
 Enter book ID: B001  
 Customer not found.  
 Press Enter key to continue...

ID	Author	Title	Qty
B001	Jane Austen	Pride and Prejudice	0
B002	F. Scott Fitzgerald	The Great Gatsby	7
B003	Gabriel Garcia Marquez	One Hundred Years of Solitude	3
B004	J.K. Rowling	Harry Potter and the Sorcerer's Stone	8
B005	George Orwell	1984	2
B006	Mark Twain	The Adventures of Tom Sawyer	10
B007	Agatha Christie	Murder on the Orient Express	4
B008	Harper Lee	To Kill a Mockingbird	6
B009	Leo Tolstoy	War and Peace	9
B010	J.R.R. Tolkien	The Hobbit	1

Jika customer ID ditemukan, tetapi Buku dengan book ID yang dimasukan tidak tersedia:

Enter customer ID: C001  
 Enter book ID: B001  
 Sorry, the book is not available for borrowing.  
 Press Enter key to continue...

#### 4. Memperbarui ketersediaan buku dan mengaitkan buku dengan pelanggan.

Sebelum diperbarui:

Book List:

ID	Author	Title	Qty
B001	Jane Austen	Pride and Prejudice	5
B002	F. Scott Fitzgerald	The Great Gatsby	7
B003	Gabriel Garcia Marquez	One Hundred Years of Solitude	3
B004	J.K. Rowling	Harry Potter and the Sorcerer's Stone	8
B005	George Orwell	1984	2
B006	Mark Twain	The Adventures of Tom Sawyer	10
B007	Agatha Christie	Murder on the Orient Express	4
B008	Harper Lee	To Kill a Mockingbird	6
B009	Leo Tolstoy	War and Peace	9
B010	J.R.R. Tolkien	The Hobbit	1

Setelah diperbarui:

Book List:

ID	Author	Title	Qty
B001	Jane Austen	Pride and Prejudice	4
B002	F. Scott Fitzgerald	The Great Gatsby	7
B003	Gabriel Garcia Marquez	One Hundred Years of Solitude	3
B004	J.K. Rowling	Harry Potter and the Sorcerer's Stone	8
B005	George Orwell	1984	2
B006	Mark Twain	The Adventures of Tom Sawyer	10
B007	Agatha Christie	Murder on the Orient Express	4
B008	Harper Lee	To Kill a Mockingbird	6
B009	Leo Tolstoy	War and Peace	9
B010	J.R.R. Tolkien	The Hobbit	1

## 2.3 View Book List:

1. Menampilkan daftar buku yang tersedia dengan detail (ID, Pengarang, Judul, Kuantitas).

```
Book Library Admin
=====
1. Logout
2. Book Borrowing
3. View Book List
4. View Customer List
5. Add Book
6. Add Customer
7. Return Book
8. Delete Book
9. Delete Customer
10. View Borrowed Books
11. Exit
3
Book List:
| ID      | Author              | Title                                | Qty |
| B001    | Jane Austen         | Pride and Prejudice                 | 5   |
| B002    | F. Scott Fitzgerald | The Great Gatsby                    | 7   |
| B003    | Gabriel Garcia Marquez | One Hundred Years of Solitude     | 3   |
| B004    | J.K. Rowling        | Harry Potter and the Sorcerer's Stone | 8   |
| B005    | George Orwell       | 1984                                | 2   |
| B006    | Mark Twain          | The Adventures of Tom Sawyer        | 10  |
| B007    | Agatha Christie     | Murder on the Orient Express        | 4   |
| B008    | Harper Lee           | To Kill a Mockingbird               | 6   |
| B009    | Leo Tolstoy          | War and Peace                       | 9   |
| B010    | J.R.R. Tolkien       | The Hobbit                          | 1   |
Press Enter key to continue...
```

2. Memungkinkan admin untuk dengan cepat memeriksa koleksi buku yang ada.

## 2.4 View Borrowed List:

1. Melihat list peminjaman buku

```
List of Borrowed Books:
| ID      | Author              | Title                                | Borrower      |
| B001    | Jane Austen         | Pride and Prejudice                 | AliceJohnson |
Press Enter key to continue...

List of Borrowed Books:
| ID      | Author              | Title                                | Borrower      |
| B001    | Jane Austen         | Pride and Prejudice                 | AliceJohnson |
| B003    | Gabriel Garcia Marquez | One Hundred Years of Solitude     | AliceJohnson |
| B007    | Agatha Christie     | Murder on the Orient Express        | EmilyDavis    |
| B009    | Leo Tolstoy          | War and Peace                       | IsabellaBrown |
Press Enter key to continue...
```

## 2.5 View Customer List:

1. Menampilkan daftar pelanggan dengan detail (ID, Nama Pengguna, Email).

```

Book Library Admin
=====
1. Logout
2. Book Borrowing
3. View Book List
4. View Customer List
5. Add Book
6. Add Customer
7. Return Book
8. Delete Book
9. Delete Customer
10. View Borrowed Books
11. Exit
4
Customer List:
| ID      | Username                      | Email                      |
| C001    | AliceJohnson                 | alice.j@binus.ac.id      |
| C002    | BobSmith                     | bob.smith@binus.ac.id    |
| C003    | CharlieBrown                 | charlie.brown@binus.ac.id|
| C004    | DavidMiller                  | david.miller@binus.ac.id |
| C005    | EmilyDavis                   | emily.davis@binus.ac.id  |
| C006    | FrankWhite                   | frank.white@binus.ac.id  |
| C007    | GraceWilson                  | grace.wilson@binus.ac.id |
| C008    | HenryTaylor                  | henry.taylor@binus.ac.id |
| C009    | IsabellaBrown                | isabella.b@binus.ac.id   |
| C010    | JackDavis                    | jack.davis@binus.ac.id   |
Press Enter key to continue...

```

2. Memberikan gambaran umum tentang basis pelanggan untuk admin.

## 2.6 Add Book:

1. Admin dapat menambahkan buku baru ke koleksi perpustakaan.
2. Memerlukan detail buku seperti ID, pengarang, judul, dan kuantitas yang tersedia.

```

Book Library Admin
=====
1. Logout
2. Book Borrowing
3. View Book List
4. View Customer List
5. Add Book
6. Add Customer
7. Return Book
8. Delete Book
9. Delete Customer
10. View Borrowed Books
11. Exit
5
Enter the number of books to add: 1
Enter book ID: B011
Enter title: The Moon and The Sun
Enter author: Ivan Adrian
Enter quantity [Quantity must > 0]: 5

```

Jika kuantitas tidak lebih dari 0:

```

Enter the number of books to add: 1
Enter book ID: B011
Enter title: The Moon and The Sun
Enter author: Ivan Adrian
Enter quantity [Quantity must > 0]: 0
Quantity must be greater than 0.
Enter quantity [Quantity must > 0]: |

```

3. Memeriksa apakah ID buku unik sebelum ditambahkan.

```

Book Library Admin
=====
1. Logout
2. Book Borrowing
3. View Book List
4. View Customer List
5. Add Book
6. Add Customer
7. Return Book
8. Delete Book
9. Delete Customer
10. View Borrowed Books
11. Exit
5
Enter the number of books to add: 1
Enter book ID: B001
Book ID must be unique!
Enter book ID: |

```

4. Menampilkan pesan keberhasilan setelah penambahan berhasil.

```

Enter the number of books to add: 1
Enter book ID: B011
Enter title: The Moon and The Sun
Enter author: Ivan Adrian
Enter quantity [Quantity must > 0]: 5
All books successfully added.
Press Enter key to continue...

```

## 2.7 Add Customer:

1. Admin dapat menambahkan satu atau lebih pelanggan ke sistem.

```

Book Library Admin
=====
1. Logout
2. Book Borrowing
3. View Book List
4. View Customer List
5. Add Book
6. Add Customer
7. Return Book
8. Delete Book
9. Delete Customer
10. View Borrowed Books
11. Exit
6
Enter the number of customers to add: 1

```

2. Memerlukan detail pelanggan seperti ID, nama pengguna, email, dan kata sandi.

```

Enter the number of customers to add: 1
Enter customer ID [Must be Unique, Ex: C011]: C012
Enter username [Not contain space]: IvanAdrian
Enter email [Must end with @binus.ac.id]: ivan.adrian@binus.ac.id
Enter password [Must 6 Characters Length]: pass12

```

3. Memeriksa apakah ID pelanggan unik sebelum ditambahkan.

Jika terdapat customer ID yang sudah ada di customer list maka:

```

Enter the number of customers to add: 1
Enter customer ID [Must be Unique, Ex: C011]: C001
Customer ID must be unique!
Enter customer ID [Must be Unique, Ex: C011]:

```

Jika terjadi username mengandung space:

```

Enter the number of customers to add: 1
Enter customer ID [Must be Unique, Ex: C011]: C012
Enter username [Not contain space]: Ivan Adrian
Enter username [Not contain space]: Ivan Adrian
Enter username [Not contain space]: IvanAdrian

```

Jika email tidak diakhiri @binus.ac.id:

```

Enter the number of customers to add: 1
Enter customer ID [Must be Unique, Ex: C011]: C012
Enter username [Not contain space]: Ivan Adrian
Enter username [Not contain space]: Ivan Adrian
Enter username [Not contain space]: IvanAdrian
Enter email [Must end with @binus.ac.id]: ivan.adrian@gmail.com
Enter email [Must end with @binus.ac.id]: ivan.adrian@binus.ac.id

```

Jika password kurang dari 6 karakter:

```
Enter the number of customers to add: 1
Enter customer ID [Must be Unique, Ex: C011]: C012
Enter username [Not contain space]: Ivan Adrian
Enter username [Not contain space]: Ivan Adrian
Enter username [Not contain space]: IvanAdrian
Enter email [Must end with @binus.ac.id]: ivan.adrian@gmail.com
Enter email [Must end with @binus.ac.id]: ivan.adrian@binus.ac.id
Enter password [Must be 6 Characters Length]: pass1
Enter password [Must be 6 Characters Length]: pass12
Press Enter key to continue...
```

4. Menampilkan pesan keberhasilan setelah penambahan berhasil.

```
Enter the number of customers to add: 1
Enter customer ID [Must be Unique, Ex: C011]: C012
Enter username [Not contain space]: Ivan Adrian
Enter username [Not contain space]: Ivan Adrian
Enter username [Not contain space]: IvanAdrian
Enter email [Must end with @binus.ac.id]: ivan.adrian@gmail.com
Enter email [Must end with @binus.ac.id]: ivan.adrian@binus.ac.id
Enter password [Must be 6 Characters Length]: pass1
Enter password [Must be 6 Characters Length]: pass12
Press Enter key to continue...

All customers successfully added.
Press Enter key to continue...
```

## 2.8 Return Book:

1. Admin dapat memproses pengembalian buku dari seorang pelanggan.
2. Memerlukan ID pelanggan dan ID buku.
3. Memeriksa apakah pelanggan ada dan apakah buku berada dalam daftar buku yang dipinjam pelanggan.

```
Book Library Admin
=====
1. Logout
2. Book Borrowing
3. View Book List
4. View Customer List
5. Add Book
6. Add Customer
7. Return Book
8. Delete Book
9. Delete Customer
10. View Borrowed Books
11. Exit
7
List of Borrowed Books:
| ID   | Author          | Title                | Borrower      |
| B001 | Jane Austen     | Pride and Prejudice  | AliceJohnson |

Press Enter key to continue...
```

Jika customer ID tidak ditemukan:

```
Enter customer ID: C012
Enter book ID to return: B001
Customer not found.
Press Enter key to continue...
```

Jika customer ID ditemukan, tetapi book ID dengan customer ID terkait tidak ditemukan:

```
Enter customer ID: C001
Enter book ID to return: B012
Book not found in the customer's borrowed books.
Press Enter key to continue...
```

Jika customer ID ditemukan dan book ID sesuai dengan customer ID terkait

```
Enter customer ID: C001
Enter book ID to return: B001
Book successfully returned.
Press Enter key to continue...
```

#### 4. Memperbarui ketersediaan buku dan menghapus keterkaitan antara buku dan pelanggan.

*Jumlah buku di B001 dari yang tadinya berjumlah 4, sekarang sudah kembali menjadi 5*

Sebelum dikembalikan:

```
Book List:
| ID      | Author                | Title                                | Qty |
| B001    | Jane Austen           | Pride and Prejudice                 | 4   |
| B002    | F. Scott Fitzgerald   | The Great Gatsby                    | 7   |
| B003    | Gabriel Garcia Marquez | One Hundred Years of Solitude      | 3   |
| B004    | J.K. Rowling          | Harry Potter and the Sorcerer's Stone | 8   |
| B005    | George Orwell         | 1984                                | 2   |
| B006    | Mark Twain            | The Adventures of Tom Sawyer        | 10  |
| B007    | Agatha Christie      | Murder on the Orient Express        | 4   |
| B008    | Harper Lee            | To Kill a Mockingbird               | 6   |
| B009    | Leo Tolstoy           | War and Peace                       | 9   |
| B010    | J.R.R. Tolkien        | The Hobbit                          | 1   |
Press Enter key to continue...
```

Sesudah dikembalikan:

```
Book List:
| ID      | Author                | Title                                | Qty |
| B001    | Jane Austen           | Pride and Prejudice                 | 5   |
| B002    | F. Scott Fitzgerald   | The Great Gatsby                    | 7   |
| B003    | Gabriel Garcia Marquez | One Hundred Years of Solitude      | 3   |
| B004    | J.K. Rowling          | Harry Potter and the Sorcerer's Stone | 8   |
| B005    | George Orwell         | 1984                                | 2   |
| B006    | Mark Twain            | The Adventures of Tom Sawyer        | 10  |
| B007    | Agatha Christie      | Murder on the Orient Express        | 4   |
| B008    | Harper Lee            | To Kill a Mockingbird               | 6   |
| B009    | Leo Tolstoy           | War and Peace                       | 9   |
| B010    | J.R.R. Tolkien        | The Hobbit                          | 1   |
Press Enter key to continue...
```

## 2.9 Delete Book:

1. Admin dapat menghapus buku dari koleksi perpustakaan.

```

Book Library Admin
=====
1. Logout
2. Book Borrowing
3. View Book List
4. View Customer List
5. Add Book
6. Add Customer
7. Return Book
8. Delete Book
9. Delete Customer
10. View Borrowed Books
11. Exit
8
Book List:
| ID      | Author                                | Title                                | Qty |
| B001    | Jane Austen                          | Pride and Prejudice                 | 5   |
| B002    | F. Scott Fitzgerald                  | The Great Gatsby                   | 7   |
| B003    | Gabriel Garcia Marquez               | One Hundred Years of Solitude     | 3   |
| B004    | J.K. Rowling                         | Harry Potter and the Sorcerer's Stone | 8   |
| B005    | George Orwell                       | 1984                               | 2   |
| B006    | Mark Twain                           | The Adventures of Tom Sawyer       | 10  |
| B007    | Agatha Christie                     | Murder on the Orient Express       | 4   |
| B008    | Harper Lee                           | To Kill a Mockingbird              | 6   |
| B009    | Leo Tolstoy                          | War and Peace                      | 9   |
| B010    | J.R.R. Tolkien                       | The Hobbit                         | 1   |
Press Enter key to continue...

Enter book ID to delete:

```

2. Memerlukan ID buku.
3. Menampilkan daftar buku saat ini dan meminta admin untuk memasukkan ID buku yang akan dihapus.

```

Book Library Admin
=====
1. Logout
2. Book Borrowing
3. View Book List
4. View Customer List
5. Add Book
6. Add Customer
7. Return Book
8. Delete Book
9. Delete Customer
10. View Borrowed Books
11. Exit
8
Book List:
| ID      | Author                                | Title                                | Qty |
| B001    | Jane Austen                          | Pride and Prejudice                 | 5   |
| B002    | F. Scott Fitzgerald                  | The Great Gatsby                   | 7   |
| B003    | Gabriel Garcia Marquez               | One Hundred Years of Solitude     | 3   |
| B004    | J.K. Rowling                         | Harry Potter and the Sorcerer's Stone | 8   |
| B005    | George Orwell                       | 1984                               | 2   |
| B006    | Mark Twain                           | The Adventures of Tom Sawyer       | 10  |
| B007    | Agatha Christie                     | Murder on the Orient Express       | 4   |
| B008    | Harper Lee                           | To Kill a Mockingbird              | 6   |
| B009    | Leo Tolstoy                          | War and Peace                      | 9   |
| B010    | J.R.R. Tolkien                       | The Hobbit                         | 1   |
Press Enter key to continue...

Enter book ID to delete: B010
Book successfully deleted.
Press Enter key to continue...

```

4. Memeriksa apakah buku dengan book ID yang dimasukan ada di dalam book list  
Jika tidak ditemukan:



```

Book List:
| ID      | Author                               | Title                               | Qty |
| B001    | Jane Austen                         | Pride and Prejudice                | 5   |
| B002    | F. Scott Fitzgerald                 | The Great Gatsby                   | 7   |
| B003    | Gabriel Garcia Marquez              | One Hundred Years of Solitude     | 3   |
| B004    | J.K. Rowling                        | Harry Potter and the Sorcerer's Stone | 8   |
| B005    | George Orwell                       | 1984                               | 2   |
| B006    | Mark Twain                          | The Adventures of Tom Sawyer       | 10  |
| B007    | Agatha Christie                    | Murder on the Orient Express       | 4   |
| B008    | Harper Lee                          | To Kill a Mockingbird              | 6   |
| B009    | Leo Tolstoy                         | War and Peace                      | 9   |
| B010    | J.R.R. Tolkien                      | The Hobbit                         | 1   |
Press Enter key to continue...

Enter book ID to delete: B020
Book not found.
Press Enter key to continue...

```

## 5. Menampilkan pesan keberhasilan setelah penghapusan berhasil.

```

Enter book ID to delete: B010
Book successfully deleted.

```

Buku dengan book ID B010 sudah tidak ada di dalam book list:

```

Book List:
| ID      | Author                               | Title                               | Qty |
| B001    | Jane Austen                         | Pride and Prejudice                | 5   |
| B002    | F. Scott Fitzgerald                 | The Great Gatsby                   | 7   |
| B003    | Gabriel Garcia Marquez              | One Hundred Years of Solitude     | 3   |
| B004    | J.K. Rowling                        | Harry Potter and the Sorcerer's Stone | 8   |
| B005    | George Orwell                       | 1984                               | 2   |
| B006    | Mark Twain                          | The Adventures of Tom Sawyer       | 10  |
| B007    | Agatha Christie                    | Murder on the Orient Express       | 4   |
| B008    | Harper Lee                          | To Kill a Mockingbird              | 6   |
| B009    | Leo Tolstoy                         | War and Peace                      | 9   |
| B010    | J.R.R. Tolkien                      | The Hobbit                         | 1   |
Press Enter key to continue...

Enter book ID to delete: B010
Book successfully deleted.
Press Enter key to continue...

```

### 2.9.1 Delete Customer:

1. Admin dapat menghapus seorang pelanggan dari sistem.
2. Memerlukan ID pelanggan.

```

Book Library Admin
=====
1. Logout
2. Book Borrowing
3. View Book List
4. View Customer List
5. Add Book
6. Add Customer
7. Return Book
8. Delete Book
9. Delete Customer
10. View Borrowed Books
11. Exit
9
Customer List:
| ID      | Username                | Email                      |
| C001    | AliceJohnson           | alice.j@binus.ac.id       |
| C002    | BobSmith                | bob.smith@binus.ac.id     |
| C003    | CharlieBrown            | charlie.brown@binus.ac.id |
| C004    | DavidMiller             | david.miller@binus.ac.id  |
| C005    | EmilyDavis              | emily.davis@binus.ac.id   |
| C006    | FrankWhite              | frank.white@binus.ac.id   |
| C007    | GraceWilson             | grace.wilson@binus.ac.id  |
| C008    | HenryTaylor             | henry.taylor@binus.ac.id  |
| C009    | IsabellaBrown           | isabella.b@binus.ac.id    |
| C010    | JackDavis               | jack.davis@binus.ac.id    |
Press Enter key to continue...

Enter customer ID to delete:

```

- Menampilkan daftar pelanggan saat ini dan meminta admin untuk memasukkan ID pelanggan yang akan dihapus, lalu mengecek apakah customer ID ada di dalam customer list

Jika tidak ditemukan:

```

Customer List:
| ID      | Username                | Email                      |
| C001    | AliceJohnson           | alice.j@binus.ac.id       |
| C002    | BobSmith                | bob.smith@binus.ac.id     |
| C003    | CharlieBrown            | charlie.brown@binus.ac.id |
| C004    | DavidMiller             | david.miller@binus.ac.id  |
| C005    | EmilyDavis              | emily.davis@binus.ac.id   |
| C006    | FrankWhite              | frank.white@binus.ac.id   |
| C007    | GraceWilson             | grace.wilson@binus.ac.id  |
| C008    | HenryTaylor             | henry.taylor@binus.ac.id  |
| C009    | IsabellaBrown           | isabella.b@binus.ac.id    |
| C010    | JackDavis               | jack.davis@binus.ac.id    |
Press Enter key to continue...

Enter customer ID to delete: C030
Customer not found.
Press Enter key to continue...

```

- Menampilkan pesan keberhasilan setelah penghapusan berhasil.

```

Enter customer ID to delete: C001
Customer successfully deleted.
Press Enter key to continue...

```

Customer dengan customer ID C001 sudah tidak ada di dalam customer list:

```
Customer List:
| ID      | Username                | Email                      |
| C002    | BobSmith                | bob.smith@binus.ac.id     |
| C003    | CharlieBrown            | charlie.brown@binus.ac.id |
| C004    | DavidMiller             | david.miller@binus.ac.id  |
| C005    | EmilyDavis              | emily.davis@binus.ac.id   |
| C006    | FrankWhite              | frank.white@binus.ac.id   |
| C007    | GraceWilson             | grace.wilson@binus.ac.id  |
| C008    | HenryTaylor             | henry.taylor@binus.ac.id  |
| C009    | IsabellaBrown           | isabella.b@binus.ac.id    |
| C010    | JackDavis               | jack.davis@binus.ac.id    |
Press Enter key to continue...
```

## 2.9.2 Exit:

1. Keluar dari program dengan lancar.

```
Book Library Admin
=====
1. Login
11. Exit
1
Enter username [default username: admin]: admin
Enter password [default password: password]: password
Login successful.
Book Library Admin
=====
1. Logout
2. Book Borrowing
3. View Book List
4. View Customer List
5. Add Book
6. Add Customer
7. Return Book
8. Delete Book
9. Delete Customer
10. View Borrowed Books
11. Exit
11
Exiting...
```

## CHAPTER III

### MULTI THREADING PART

Bagian dari program yang menerapkan multi-threading terdapat pada metode `addBook()` dan `addCustomer()`. Di kedua metode ini, setiap objek buku dan pelanggan yang baru ditambahkan dimasukkan ke dalam vektor `addedBooks` dan `addedCustomers`. Kemudian, untuk setiap objek dalam vektor tersebut, sebuah thread baru dibuat untuk menambahkannya ke dalam vektor utama (`bookList` atau `customerList`). Proses ini dilakukan dengan menggunakan blok `synchronized` untuk memastikan operasi penambahan tersebut aman dari race condition, di mana beberapa thread dapat mengakses dan memodifikasi data bersamaan.

#### 3.1 Add Book

##### Source Code:

```
for (Book book : addedBooks) {
    Thread addBookThread = new Thread() -> {
        synchronized (bookList) {
            try {
                bookList.add(book);
            } catch (Exception e) {
                System.out.println("Error adding book: " + e.getMessage());
            }
        }
    };

    addBookThread.start();
}
```

##### Penjelasan:

##### Pendekatan Multi-threading pada Metode `addBook()`

###### 1. Thread Baru untuk Setiap Buku:

- Program membuat thread baru (`addBookThread`) untuk setiap buku yang akan ditambahkan.
- Ini memungkinkan program untuk menangani penambahan buku secara paralel, meningkatkan responsivitas program.

## 2. Synchronized Block:

- Thread-thread tersebut dijalankan dalam blok synchronized untuk memastikan bahwa operasi penambahan buku dilakukan secara aman.
- Dengan demikian, mencegah terjadinya race condition di mana dua atau lebih thread berusaha mengakses dan memodifikasi bookList secara bersamaan.

## 3. Asynchronous Execution:

- Thread-thread yang dibuat dapat dijalankan secara asinkron, memungkinkan program untuk melanjutkan eksekusi tanpa harus menunggu penambahan buku selesai.
- Ini meningkatkan efisiensi program dan memberikan pengalaman pengguna yang lebih baik.

## 4. Pesan Keberhasilan/KeGagal-an:

- Program memberikan pesan ke pengguna terkait keberhasilan atau kegagalan penambahan buku ke dalam vektor bookList.
- Pemberian pesan ini tidak diblokir oleh operasi penambahan buku, karena thread-thread berjalan secara asinkron.

Dalam metode addBook(), program mulai dengan meminta pengguna memasukkan jumlah buku yang akan ditambahkan. Setiap buku yang akan ditambahkan memicu pembuatan objek Book baru, dan objek tersebut dimasukkan ke dalam vektor addedBooks. Selanjutnya, untuk setiap buku yang ditambahkan, thread baru dibuat dengan menggunakan lambda expression. Dalam blok thread, operasi penambahan buku ke vektor bookList dilakukan di dalam blok synchronized, memastikan bahwa operasi tersebut aman secara thread.

## 3.2 Add Customer

### Source Code:

```
for (Customer customer : addedCustomers) {  
    Thread addCustomerThread = new Thread(() -> {  
        synchronized (customerList) {  
            try {  
                customerList.add(customer);  
            } catch (Exception e) {
```

```

        System.out.println("Error adding customer: " + e.getMessage());
    }
}
});

addCustomerThread.start();
}

```

### **Penjelasan:**

#### **Pendekatan Multi-threading pada Metode addCustomer()**

1. Thread Baru untuk Setiap Customer:
  - Program membuat thread baru (addCustomerThread) untuk setiap customer yang akan ditambahkan.
  - Ini memungkinkan program untuk menangani penambahan customer secara paralel, meningkatkan responsivitas program.
2. Synchronized Block:
  - Thread-thread tersebut dijalankan dalam blok synchronized untuk memastikan bahwa operasi penambahan customer dilakukan secara aman.
  - Dengan demikian, mencegah terjadinya race condition di mana dua atau lebih thread berusaha mengakses dan memodifikasi customerList secara bersamaan.
3. Asynchronous Execution:
  - Thread-thread yang dibuat dapat dijalankan secara asinkron, memungkinkan program untuk melanjutkan eksekusi tanpa harus menunggu penambahan customer selesai.
  - Ini meningkatkan efisiensi program dan memberikan pengalaman pengguna yang lebih baik.
4. Pesan Keberhasilan/KeGagal-an:
  - Program memberikan pesan ke pengguna terkait keberhasilan atau kegagalan penambahan customer ke dalam vektor bookList.
  - Pemberian pesan ini tidak diblokir oleh operasi penambahan customer, karena thread-thread berjalan secara asinkron.

Dalam metode `addCustomer()`, program meminta pengguna memasukkan jumlah pelanggan yang akan ditambahkan. Setiap pelanggan yang akan ditambahkan menghasilkan objek `Customer` baru, dan objek tersebut dimasukkan ke dalam vektor `addedCustomers`. Selanjutnya, untuk setiap pelanggan yang ditambahkan, thread baru dibuat dengan menggunakan `lambda expression`. Dalam blok thread, operasi penambahan pelanggan ke vektor `customerList` dilakukan di dalam blok `synchronized`, memastikan keamanan operasi tersebut saat digunakan oleh beberapa thread secara bersamaan.

## **CHAPTER IV**

### **RELATION BETWEEN CLASS**

Betikut ini adalah tinjauan hubungan antara kelas-kelas tersebut:

#### **4.1 User Class:**

Hubungan dengan Admin dan Customer:

- Ini adalah kelas dasar yang memiliki informasi umum tentang pengguna, seperti id, username, email, dan password.
- Tidak ada relasi khusus di sini, hanya kelas dasar untuk Admin dan Customer.

#### **4.2 Admin Class:**

Hubungan dengan User:

- Admin adalah turunan dari kelas User, menunjukkan hubungan pewarisan antara kelas User dan Admin.

#### **4.3 Customer Class:**

Hubungan dengan User:

- Customer juga adalah turunan dari kelas User, menunjukkan hubungan pewarisan antara kelas User dan Customer.

Hubungan dengan Book:

- Customer memiliki vector (vektor) `borrowedBooks` yang berisi objek-objek dari kelas Book.
- Ini menunjukkan hubungan komposisi, di mana Customer memiliki kepemilikan penuh terhadap objek-objek Book yang dipinjam.

#### **4.4 Book Class:**

- Tidak ada hubungan khusus dengan kelas lain di sini.



## 4.5 MainClass (Library System):

Hubungan dengan Admin, Customer, dan Book:

- MainClass menggunakan objek-objek dari kelas Admin, Customer, dan Book untuk menjalankan fungsionalitas perpustakaan.
- Ini bisa dianggap sebagai hubungan asosiasi, karena MainClass menggunakan objek-objek dari kelas lain, tetapi tidak ada kepemilikan atau ketergantungan yang kuat.

Beberapa bagian yang menunjukkan hubungan-hubungan ini:

### Class Relation

#### 1. Pewarisan (Inheritance):

Hubungan pewarisan antara User, Admin, dan Customer terlihat di sini:

```
public class Admin extends User {  
    // ...  
}  
  
public class Customer extends User {  
    // ...  
},
```

#### 2. Komposisi (Composition):

Hubungan komposisi antara Customer dan Book terlihat di sini, di mana Customer memiliki vector borrowedBooks yang berisi objek-objek Book:

```
import java.util.Vector;  
  
class Customer extends User{  
    private Vector<Book> borrowedBooks;  
  
    public Customer(String id, String username, String email, String password) {  
        super(id, username, email, password);  
        this.borrowedBooks = new Vector<>();  
    }  
  
    public Vector<Book> getBorrowedBooks() {  
        return borrowedBooks;  
    }  
  
    public void addBorrowedBook(Book book) {  
        borrowedBooks.add(book);  
    }  
  
    public void returnBook(Book book) {  
        borrowedBooks.remove(book);  
    }  
}
```

```

    public void returnBook(Book book) {
        borrowedBooks.remove(book);
    }

    public void printBooks() {
        System.out.println("Borrowed Books:");
        for (Book book : borrowedBooks) {
            System.out.println(book);
        }
    }

    @Override
    public String toString() {
        return String.format("\n| %-5s | %-30s | %-30s |", id, username, email);
    }
}

```

### 3. Penggunaan objek-objek kelas lain di MainClass (Association):

Penggunaan objek-objek dari kelas Admin, Customer, dan Book di MainClass menunjukkan hubungan asosiasi:

```

public class MainClass {
    static Vector<Book> bookList = new Vector<>();
    static Vector<Customer> customerList = new Vector<>();
    static Vector<Admin> adminList = new Vector<>();
    // ...

    public static void main(String[] args) {
        // ...

        // Penggunaan objek-objek Admin, Customer, dan Book
        adminList.add(new Admin("A001", "admin", "hardi.kusuma@library.com", "password"));
        // ...
        customerList.add(new Customer("C001", "AliceJohnson", "alice.j@gmail.com", "pass1"
            ));
        // ...
        bookList.add(new Book("B001", "Jane Austen", "Pride and Prejudice", 5));
        // ...
    }
    // ...
}

```

## **CHAPTER V**

### **CONCLUSION**

Program Book Library yang telah dibuat menghasilkan solusi yang efektif untuk manajemen perpustakaan, terutama dalam konteks tugas seorang librarian. Dengan fitur-fitur seperti peminjaman, pengembalian buku, penambahan buku, dan penambahan pelanggan, program ini menyediakan alat yang dapat digunakan untuk mengelola kegiatan sehari-hari di perpustakaan. Penggunaan struktur data Vector memudahkan penyimpanan dan manipulasi data, sementara konsep multithreading memberikan peningkatan efisiensi dengan memproses penambahan buku dan pelanggan secara paralel.

Program ini juga menonjolkan aspek keamanan dengan memperhatikan validasi input, seperti pemeriksaan ID unik untuk buku dan pelanggan. Proses login memerlukan kombinasi username dan password yang valid, memberikan lapisan keamanan pada akses admin. Selain itu, penggunaan menu-driven approach memudahkan interaksi dengan pengguna, dengan setiap menu memiliki logika tersendiri dan memberikan respons yang jelas.

Kode program ditulis dengan baik dan terstruktur, memanfaatkan konsep OOP dengan baik melalui penggunaan class untuk merepresentasikan entitas seperti buku, pelanggan, dan admin. Pengembangan lebih lanjut dapat dilakukan, termasuk penambahan fitur atau penyempurnaan desain antarmuka pengguna. Program ini memberikan dasar yang kuat untuk pengelolaan perpustakaan yang efisien dan dapat diandalkan.