

Laporan Tugas Besar Pemrograman Berorientasi Objek



Oleh:

Nama : Jeremy Nathanael Sidabutar
NPM : 4522210089

**S1-Teknik Informatika Fakultas Teknik Universitas
Pancasila
2022/2023**

Topik : Riwayat Peminjaman Buku Perpustakaan

Permasalahan yang dihadapi

Latar belakang pembuatan program riwayat peminjaman buku perpustakaan didasari oleh pentingnya peran perpustakaan sebagai lembaga pendidikan dan pengetahuan. Perpustakaan menyediakan akses ke berbagai sumber daya, khususnya koleksi buku, yang memerlukan manajemen efisien untuk melacak dan mencatat riwayat peminjaman. Sistem manual dalam pencatatan dapat menjadi rumit dan rentan terhadap kesalahan manusiawi.

Rumusan masalah dalam konteks ini melibatkan pertanyaan mengenai cara meningkatkan efisiensi dan akurasi pencatatan riwayat peminjaman buku di perpustakaan. Pembuat program juga berfokus pada implementasi solusi komputer yang dapat membantu manajemen perpustakaan dalam mencatat dan melacak riwayat peminjaman buku. Pemilihan konsep pemrograman berorientasi objek (OOP) menjadi pertimbangan untuk merancang program yang modular dan mudah dipahami.

Selain itu, pertanyaan juga diajukan mengenai implementasi JavaFX untuk menciptakan antarmuka pengguna yang interaktif dan user-friendly. Tujuan utama pembuatan program adalah membantu manajemen perpustakaan, memberikan antarmuka pengguna yang mudah digunakan, dan memanfaatkan konsep OOP. Manfaat yang diharapkan meliputi percepatan proses pencatatan, pengurangan kesalahan manusiawi, dan pemahaman yang lebih baik bagi pengguna.

Sasaran pengguna program ini adalah petugas perpustakaan yang melakukan pencatatan peminjaman dan pengembalian buku, serta anggota perpustakaan yang ingin melihat riwayat peminjaman buku mereka. Beberapa pertimbangan tambahan melibatkan keamanan data pribadi, dukungan operasi dasar seperti peminjaman dan pengembalian buku, serta antarmuka pengguna yang ramah dan mudah dipahami oleh berbagai kalangan. Dengan demikian, program ini diharapkan memberikan manfaat signifikan dalam pengelolaan perpustakaan dan meningkatkan pengalaman pengguna.

Siapa saja yang terlibat

Dalam pengembangan program riwayat peminjaman buku perpustakaan, terlibat beberapa pihak yang memiliki peran dan tanggung jawab masing-masing. Berikut adalah beberapa pihak yang umumnya terlibat dalam pengembangan program tersebut:

1. Programmer/ Pengembang Perangkat Lunak:

- Bertanggung jawab untuk merancang, mengembangkan, dan mengimplementasikan program.
- Menyusun struktur program, menggunakan konsep pemrograman berorientasi objek (OOP), dan memastikan kehandalan serta keamanan program.

2. Analisis Sistem:

- Melakukan analisis kebutuhan sistem untuk memahami persyaratan dan tujuan program.
- Mengidentifikasi fungsi-fungsi yang dibutuhkan, mengumpulkan informasi dari pengguna, dan merancang spesifikasi program.

3. Desainer Antarmuka Pengguna (UI/UX Designer):

- Bertanggung jawab merancang antarmuka pengguna yang intuitif dan ramah pengguna.
- Memastikan bahwa tata letak, warna, dan elemen-elemen desain lainnya mendukung pengalaman pengguna yang baik.

4. Manajemen Perpustakaan:

- Memberikan pemahaman tentang kebutuhan dan proses di perpustakaan.

- Memberikan masukan terkait alur kerja peminjaman dan pengembalian buku serta informasi apa yang perlu dicatat dalam riwayat peminjaman.

5. Pengguna Akhir (Petugas Perpustakaan dan Anggota Perpustakaan):

- Pengguna akhir adalah petugas perpustakaan yang akan menggunakan program untuk melakukan operasi peminjaman, pengembalian, dan melihat riwayat.
- Anggota perpustakaan dapat menggunakan program untuk melihat riwayat peminjaman buku mereka sendiri.

6. Administrator Sistem:

- Bertanggung jawab untuk mengelola dan memelihara infrastruktur teknis yang digunakan oleh program.
- Menjamin keamanan data dan melakukan pemeliharaan rutin jika diperlukan.

7. Pengujian (Quality Assurance)

- Melakukan pengujian perangkat lunak untuk memastikan bahwa program berfungsi dengan baik dan sesuai dengan spesifikasi.
- Mengidentifikasi dan melaporkan bug atau masalah kinerja.

8. Manajemen Proyek:

- Bertanggung jawab untuk merencanakan, mengarahkan, dan mengelola proyek pengembangan perangkat lunak.
- Mengkoordinasikan antara anggota tim dan memastikan proyek berjalan sesuai jadwal.

9. Pemangku Kepentingan (Stakeholders):

- Termasuk pihak yang memiliki kepentingan terhadap keberhasilan program, seperti pimpinan perpustakaan, pengguna akhir, dan mungkin pihak eksternal terkait.

Data apa saja yang disimpan diolah :

Program riwayat peminjaman buku perpustakaan umumnya menyimpan dan mengolah beberapa jenis data untuk memastikan pencatatan peminjaman dan pengembalian buku berjalan dengan baik. Berikut adalah jenis-jenis data yang biasanya disimpan dan diolah dalam program tersebut:

1. Informasi Buku:

- Judul buku
- Penulis buku
- Edisi (khusus untuk buku referensi)
- Genre (khusus untuk buku fiksi)

2. Informasi Peminjam:

- Nama peminjam
- Nomor identifikasi peminjam (ID anggota perpustakaan)
- Detail kontak (alamat, nomor telepon, dll.)

3. Informasi Peminjaman:

- Tanggal peminjaman
- Tanggal batas pengembalian
- Status peminjaman (aktif atau sudah dikembalikan)
- Catatan tambahan (jika ada)

4. Riwayat Peminjaman:

- Catatan peminjaman yang telah selesai, termasuk informasi buku yang dipinjam, peminjam, dan tanggal peminjaman.

5. Log Sistem:

- Log aktivitas sistem seperti catatan login, logout, dan perubahan status peminjaman.

6. Pengaturan dan Konfigurasi:

- Konfigurasi sistem, seperti pengaturan batas waktu peminjaman, pengaturan perpustakaan, dan preferensi lainnya.

7. Data Pengguna dan Hak Akses:

- Informasi pengguna dan hak akses, terutama untuk petugas perpustakaan dan administrator sistem.

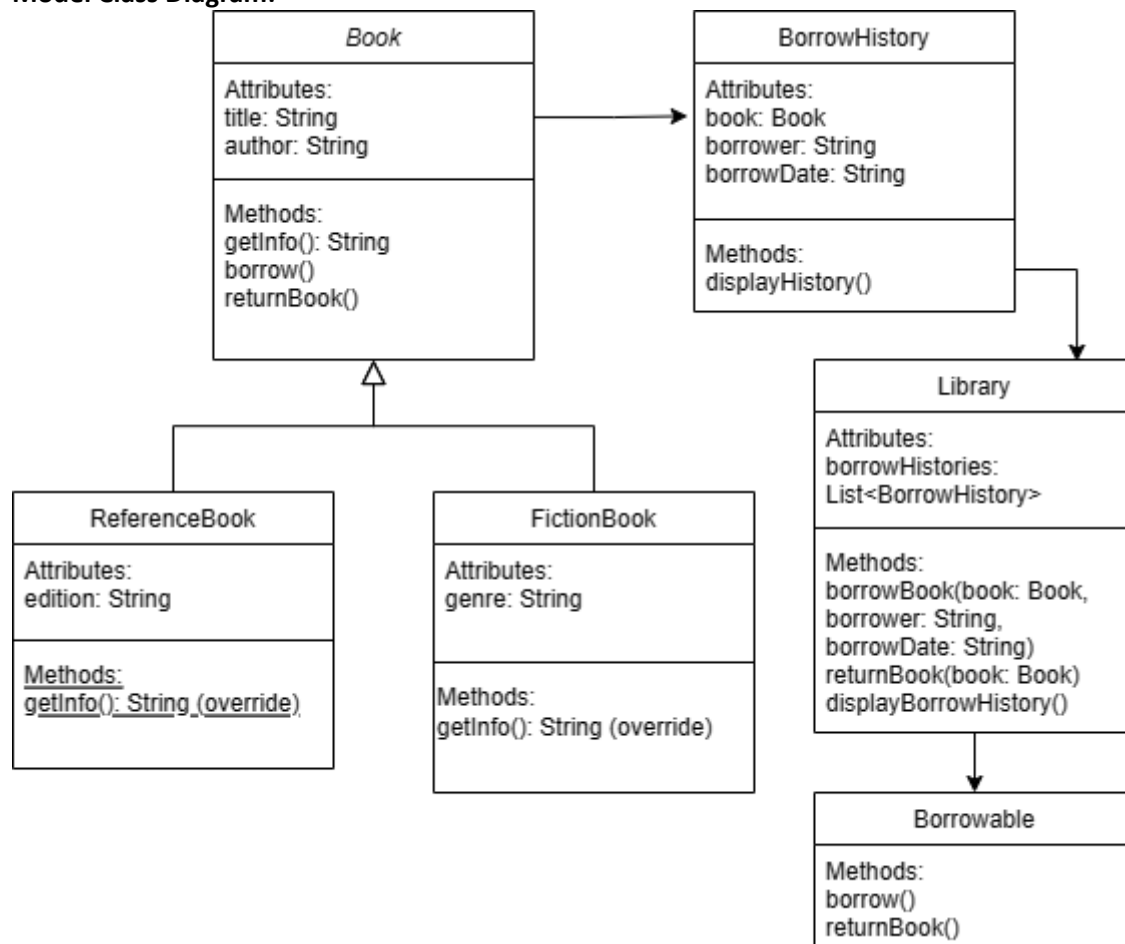
8. Statistik dan Analisis:

- Data statistik terkait dengan frekuensi peminjaman, buku yang paling sering dipinjam, dan informasi analisis lainnya yang mungkin diperlukan untuk manajemen perpustakaan.

9. Data Master:

- Data master terkait dengan buku, peminjam, dan informasi lain yang diperlukan untuk operasi sistem.

Model Class Diagram:



Buatkan relasi/turunan/abstrak/interface:

Dalam program riwayat peminjaman buku perpustakaan yang telah dibuat, terdapat beberapa konsep dasar pemrograman berorientasi objek (OOP), seperti turunan (inheritance), relasi antar kelas, abstraksi, dan penggunaan interface. Berikut adalah penjelasan singkat untuk setiap konsep tersebut:

1. Turunan (Inheritance):

- Konsep turunan digunakan dalam kelas `Book`, `ReferenceBook`, dan `FictionBook`.
- `ReferenceBook` dan `FictionBook` adalah kelas anak dari kelas `Book`, mewarisi sifat-sifat umum dari kelas `Book`, seperti judul dan penulis.

```
abstract class Book implements Borrowable {  
    // ...  
}
```

```
class ReferenceBook extends Book {  
    // ...  
}
```

```
class FictionBook extends Book {  
    // ...  
}
```

2. Relasi/Agregasi:

- Terdapat relasi antara kelas `Library` dan `BorrowHistory`, dimana `Library` memiliki list `borrowHistories`.
- Ini adalah contoh agregasi, dimana `Library` "mengandung" atau "memiliki" banyak `BorrowHistory`.

```
class Library {  
    private List<BorrowHistory> borrowHistories;  
    // ...  
}
```

3. Abstraksi dan Interface:

- Terdapat penggunaan abstraksi melalui interface `Borrowable` yang menyatakan dua metode, `borrow()` dan `returnBook()`.
- Kelas `Book`, `ReferenceBook`, dan `FictionBook` mengimplementasikan interface `Borrowable`.

```
interface Borrowable {  
    void borrow();  
    void returnBook();  
}
```

```
abstract class Book implements Borrowable {
```

```

    // ...
}

class ReferenceBook extends Book {
    // ...
}

class FictionBook extends Book {
    // ...
}

```

4. Asosiasi:

- Asosiasi terjadi antara kelas `Library` dan `Book` melalui metode `borrowBook()`.
- Ini menunjukkan hubungan asosiatif, dimana `Library` menggunakan objek `Book` dalam operasi peminjaman.

```

class Library {
    // ...
    public void borrowBook(Book book, String borrower, String borrowDate) {
        // ...
    }
}

```

Buatkan skenario:

Dengan konsep-konsep tersebut, program dapat dianggap sebagai contoh implementasi konsep OOP yang baik, memanfaatkan abstraksi, turunan, asosiasi, dan interface untuk membuat struktur program yang modular, mudah dimengerti, dan dapat diperluas.

Berikut adalah beberapa skenario penggunaan untuk program riwayat peminjaman buku perpustakaan:

Skenario 1: Peminjaman Buku Baru

1. Aktor: Anggota Perpustakaan

2. Deskripsi:

- Anggota perpustakaan, Alice, ingin meminjam buku baru untuk dibaca.
- Alice masuk ke sistem menggunakan ID anggota dan memilih buku yang ingin dipinjam.
- Setelah memilih buku, Alice mengonfirmasi peminjaman dan sistem mencatat informasi peminjaman tersebut.

Skenario 2: Pengembalian Buku

1. Aktor: Anggota Perpustakaan

2. Deskripsi:

- Setelah membaca buku, Alice ingin mengembalikan buku tersebut.
- Alice masuk ke sistem, memilih opsi pengembalian, dan mengonfirmasi pengembalian buku.
- Sistem mencatat informasi pengembalian, mengupdate status peminjaman, dan jika ada keterlambatan, menghitung denda yang harus dibayarkan.

Skenario 3: Melihat Riwayat Peminjaman Sendiri

1. Aktor: Anggota Perpustakaan

2. Deskripsi:

- Alice ingin melihat riwayat peminjaman buku yang pernah dia lakukan.
- Alice masuk ke sistem dan memilih opsi untuk melihat riwayat peminjaman.
- Sistem menampilkan riwayat peminjaman Alice, termasuk buku yang pernah dipinjam, tanggal peminjaman, dan status pengembalian.

Skenario 4: Pencarian Buku dalam Koleksi

1. Aktor: Petugas Perpustakaan

2. Deskripsi:

- Petugas perpustakaan, Bob, menerima permintaan dari anggota yang ingin mencari buku tertentu.
- Bob masuk ke sistem, menggunakan fitur pencarian, dan mencari buku berdasarkan judul, penulis, atau kategori.
- Sistem menampilkan hasil pencarian yang sesuai dengan kriteria yang dimasukkan oleh Bob.

Skenario 5: Melihat Statistik Peminjaman Bulanan

1. Aktor: Administrator Sistem

2. Deskripsi:

- Administrator ingin melihat statistik peminjaman bulanan untuk mengidentifikasi tren dan kebutuhan koleksi buku.
- Administrator masuk ke sistem, memilih opsi statistik, dan memilih filter untuk melihat statistik peminjaman bulan tertentu.
- Sistem menampilkan statistik peminjaman, termasuk buku yang paling sering dipinjam dan jumlah peminjaman bulan itu.

Skenario-skenario di atas mencakup berbagai interaksi yang mungkin terjadi antara pengguna dan sistem, mencakup operasi peminjaman, pengembalian, pencarian, dan analisis statistik. Program dirancang untuk mendukung kebutuhan sehari-hari perpustakaan dan memastikan pengalaman pengguna yang efisien dan nyaman.

Tuliskan source code:

Program:

```
package com.example.demo2;
```

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.TextArea;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;
```

```
import java.util.ArrayList;
import java.util.List;
```

```
interface Borrowable {
    void borrow();
    void returnBook();
}
```

```
abstract class Book implements Borrowable {
    private String title;
```

```

private String author;

public Book(String title, String author) {
    this.title = title;
    this.author = author;
}

public String getTitle() {
    return title;
}

public String getAuthor() {
    return author;
}

@Override
public void borrow() {
    System.out.println("Borrowing book: " + title);
}

@Override
public void returnBook() {
    System.out.println("Returning book: " + title);
}
}

class BorrowHistory {
    private Book book;
    private String borrower;
    private String borrowDate;

    public BorrowHistory(Book book, String borrower, String borrowDate) {
        this.book = book;
        this.borrower = borrower;
        this.borrowDate = borrowDate;
    }

    public String getHistoryAsString() {
        return "Book: " + book.getTitle() + "\n" +
            "Borrower: " + borrower + "\n" +
            "Borrow Date: " + borrowDate + "\n";
    }
}

class ReferenceBook extends Book {
    private String edition;

    public ReferenceBook(String title, String author, String edition) {
        super(title, author);
        this.edition = edition;
    }
}

```



```

    public String getEdition() {
        return edition;
    }
}

class FictionBook extends Book {
    private String genre;

    public FictionBook(String title, String author, String genre) {
        super(title, author);
        this.genre = genre;
    }

    public String getGenre() {
        return genre;
    }
}

class Library {
    private List<BorrowHistory> borrowHistories;

    public Library() {
        this.borrowHistories = new ArrayList<>();
    }

    public void borrowBook(Book book, String borrower, String borrowDate) {
        book.borrow();
        borrowHistories.add(new BorrowHistory(book, borrower, borrowDate));
    }

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

    public String getBorrowHistoryAsString() {
        StringBuilder historyString = new StringBuilder();
        historyString.append("Borrow History:\n");
        borrowHistories.forEach(history ->
historyString.append(history.getHistoryAsString()).append("\n"));
        return historyString.toString();
    }
}

public class Main extends Application {
    public static void main(String[] args) {
        launch(args);
    }

    @Override
    public void start(Stage primaryStage) {

```

```

    ReferenceBook referenceBook = new ReferenceBook("Java Programming", "John Doe", "3rd
Edition");
    FictionBook fictionBook = new FictionBook("The Adventure", "Jane Doe", "Adventure");

    Library library = new Library();

    library.borrowBook(referenceBook, "Alice", "2024-01-01");
    library.borrowBook(fictionBook, "Bob", "2024-01-05");

    library.returnBook(referenceBook);
    library.returnBook(fictionBook);

    // JavaFX UI
    VBox root = new VBox();
    Button displayHistoryButton = new Button("Display Borrow History");

    // TextArea untuk menampilkan output
    TextArea outputTextArea = new TextArea();
    outputTextArea.setEditable(false); // Agar tidak bisa diedit oleh pengguna
    outputTextArea.setWrapText(true); // Agar teks berjalan ke baris baru jika tidak muat

    // Menambahkan aksi tombol
    displayHistoryButton.setOnAction(event -> {
        String borrowHistory = library.getBorrowHistoryAsString();
        outputTextArea.setText(borrowHistory);
    });

    root.getChildren().addAll(displayHistoryButton, outputTextArea);

    primaryStage.setTitle("Library Management");
    primaryStage.setScene(new Scene(root, 400, 300));
    primaryStage.show();
}
}

```

Controller:

```

package com.example.demo2;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.control.Button;
import javafx.scene.control.TextArea;
import javafx.scene.layout.VBox;

public class LibraryController {

    @FXML
    private VBox root;

    @FXML
    private Button displayHistoryButton;

```

```

@FXML
private TextArea outputTextArea;

private Library library;

@FXML
public void initialize() {
    ReferenceBook referenceBook = new ReferenceBook("Java Programming", "John Doe", "3rd
Edition");
    FictionBook fictionBook = new FictionBook("The Adventure", "Jane Doe", "Adventure");

    library = new Library();

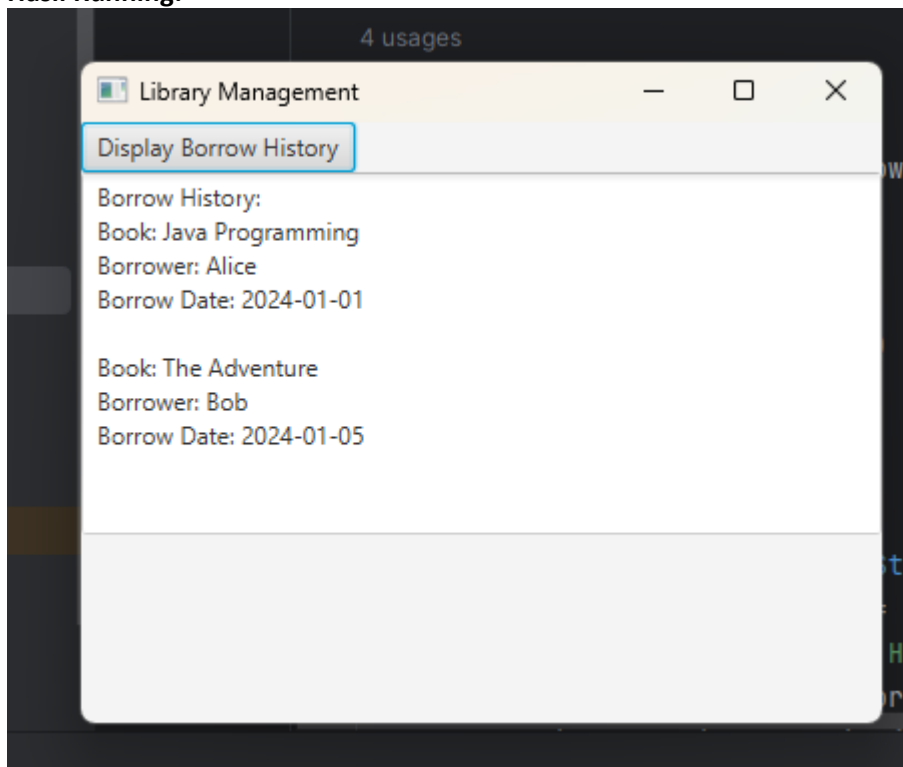
    library.borrowBook(referenceBook, "Alice", "2024-01-01");
    library.borrowBook(fictionBook, "Bob", "2024-01-05");

    displayHistoryButton.setOnAction(this::displayHistory);
}

@FXML
private void displayHistory(ActionEvent event) {
    String borrowHistory = library.getBorrowHistoryAsString();
    outputTextArea.setText(borrowHistory);
}
}

```

Hasil Running:



Link Github:

[Northenwest/Tugas_Besar_PBO \(github.com\)](https://github.com/Northenwest/Tugas_Besar_PBO)