**Tugas Praktikum   
Pemrograman Berorientasi Objek**

“Tugas 11 Studi Kasus Perpustakaan*”*

**Kelas A**

****

Oleh :

Muhamad Ihsan Setiawan

4523210069

Dosen :

**Adi Wahyu Pribadi**

**S1 Teknik Informatika**

**Universitas Pancasila**

**2024/2025**

**Program Code**

**Book.java**

package com.library.model;

import java.time.LocalDate;

public class Book {

    private String title;

    private String author;

    private String isbn;

    private boolean isBorrowed;

    private LocalDate borrowDate;

    private LocalDate dueDate;

    public Book(String title, String author, String isbn) {

        this.title = title;

        this.author = author;

        this.isbn = isbn;

        this.isBorrowed = false;

    }

    // Getter dan Setter

    public String getTitle() {

        return title;

    }

    public String getAuthor() {

        return author;

    }

    public String getIsbn() {

        return isbn;

    }

    public boolean isBorrowed() {

        return isBorrowed;

    }

    public void setBorrowed(boolean borrowed) {

        isBorrowed = borrowed;

    }

    public LocalDate getBorrowDate() {

        return borrowDate;

    }

    public void setBorrowDate(LocalDate borrowDate) {

        this.borrowDate = borrowDate;

        this.dueDate = borrowDate.plusDays(14); // Misal batas peminjaman 14 hari

    }

    public LocalDate getDueDate() {

        return dueDate;

    }

}

**Member.java**

package com.library.model;

import java.time.LocalDate;

import java.util.ArrayList;

import java.util.List;

public class Member extends Person {

    private List<Book> borrowedBooks;

    private double fine;

    public Member(String name, String id) {

        super(name, id);

        borrowedBooks = new ArrayList<>();

        fine = 0.0;

    }

    public void borrowBook(Book book) {

        borrowedBooks.add(book);

        book.setBorrowed(true);

        book.setBorrowDate(LocalDate.now());

    }

    public void returnBook(Book book) {

        borrowedBooks.remove(book);

        book.setBorrowed(false);

        // Cek keterlambatan

        if (LocalDate.now().isAfter(book.getDueDate())) {

            long daysLate = java.time.temporal.ChronoUnit.DAYS.between(book.getDueDate(), LocalDate.now());

            double lateFine = daysLate \* 5000; // Misal denda Rp5.000 per hari

            addFine(lateFine);

            System.out.println("Anda terlambat " + daysLate + " hari. Denda: Rp" + lateFine);

        }

    }

    public List<Book> getBorrowedBooks() {

        return borrowedBooks;

    }

    public double getFine() {

        return fine;

    }

    public void addFine(double amount) {

        fine += amount;

    }

    public void payFine(double amount) {

        if (amount <= fine) {

            fine -= amount;

            System.out.println("Denda sebesar Rp" + amount + " telah dibayar.");

        } else {

            System.out.println("Jumlah pembayaran melebihi denda yang ada.");

        }

    }

}

**Person.java**

package com.library.model;

public abstract class Person {

    private String name;

    private String id;

    public Person(String name, String id) {

        this.name = name;

        this.id = id;

    }

    // Getter

    public String getName() {

        return name;

    }

    public String getId() {

        return id;

    }

}

**LibraryService.java**

package com.library.service;

import com.library.model.Book;

import com.library.model.Member;

import java.util.ArrayList;

import java.util.List;

public class LibraryService {

    private List<Book> books;

    private List<Member> members;

    public LibraryService() {

        books = new ArrayList<>();

        members = new ArrayList<>();

    }

    // Metode untuk menambahkan buku

    public void addBook(Book book) {

        books.add(book);

        System.out.println("Buku '" + book.getTitle() + "' telah ditambahkan.");

    }

    // Metode untuk menghapus buku

    public void removeBook(String isbn) {

        Book book = findBook(isbn);

        if (book != null) {

            books.remove(book);

            System.out.println("Buku '" + book.getTitle() + "' telah dihapus.");

        } else {

            System.out.println("Buku dengan ISBN " + isbn + " tidak ditemukan.");

        }

    }

    // Metode untuk mendaftarkan anggota

    public void registerMember(Member member) {

        members.add(member);

        System.out.println("Anggota '" + member.getName() + "' telah didaftarkan.");

    }

    // Metode untuk meminjam buku

    public void borrowBook(String memberId, String isbn) {

        Member member = findMember(memberId);

        Book book = findBook(isbn);

        if (member != null && book != null && !book.isBorrowed()) {

            member.borrowBook(book);

            System.out.println("Buku '" + book.getTitle() + "' telah dipinjam oleh '" + member.getName() + "'.");

        } else {

            System.out.println("Peminjaman gagal. Buku mungkin sudah dipinjam atau data tidak ditemukan.");

        }

    }

    // Metode untuk mengembalikan buku

    public void returnBook(String memberId, String isbn) {

        Member member = findMember(memberId);

        Book book = findBook(isbn);

        if (member != null && book != null && book.isBorrowed()) {

            member.returnBook(book);

            System.out.println("Buku '" + book.getTitle() + "' telah dikembalikan oleh '" + member.getName() + "'.");

            if (member.getFine() > 0) {

                System.out.println("Total denda Anda: Rp" + member.getFine());

            }

        } else {

            System.out.println("Pengembalian gagal. Buku mungkin belum dipinjam atau data tidak ditemukan.");

        }

    }

    // Metode untuk mencari anggota

    public Member findMember(String memberId) {

        for (Member member : members) {

            if (member.getId().equals(memberId)) {

                return member;

            }

        }

        return null;

    }

    // Metode untuk mencari buku

    public Book findBook(String isbn) {

        for (Book book : books) {

            if (book.getIsbn().equals(isbn)) {

                return book;

            }

        }

        return null;

    }

    // Metode untuk menampilkan buku yang tersedia

    public void listAvailableBooks() {

        System.out.println("Daftar Buku yang Tersedia:");

        for (Book book : books) {

            if (!book.isBorrowed()) {

                System.out.println("- " + book.getTitle() + " oleh " + book.getAuthor() + " (ISBN: " + book.getIsbn() + ")");

            }

        }

    }

    // Metode untuk menampilkan buku yang dipinjam

    public void listBorrowedBooks() {

        System.out.println("Daftar Buku yang Dipinjam:");

        for (Book book : books) {

            if (book.isBorrowed()) {

                System.out.println("- " + book.getTitle() + " oleh " + book.getAuthor() + " (ISBN: " + book.getIsbn() + ")");

            }

        }

    }

    // Metode untuk menampilkan daftar anggota dan buku yang dipinjam

    public void listMembers() {

        if (members.isEmpty()) {

            System.out.println("Belum ada anggota yang terdaftar.");

        } else {

            System.out.println("Daftar Anggota:");

            for (Member member : members) {

                System.out.println("- Nama: " + member.getName() + ", ID: " + member.getId());

                List<Book> borrowedBooks = member.getBorrowedBooks();

                if (borrowedBooks.isEmpty()) {

                    System.out.println("  Tidak ada buku yang dipinjam.");

                } else {

                    System.out.println("  Buku yang dipinjam:");

                    for (Book book : borrowedBooks) {

                        System.out.println("    \* " + book.getTitle() + " (ISBN: " + book.getIsbn() + ")");

                    }

                }

            }

        }

    }

}

**Main.java**

package com.library;

import com.library.model.Book;

import com.library.model.Member;

import com.library.service.LibraryService;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        LibraryService libraryService = new LibraryService();

        Scanner scanner = new Scanner(System.in);

        int mainChoice = 0;

        // Tambahkan beberapa buku dan anggota contoh

        libraryService.addBook(new Book("Pemrograman Java", "Andi", "ISBN001"));

        libraryService.addBook(new Book("Struktur Data", "Budi", "ISBN002"));

        libraryService.addBook(new Book("Basis Data", "Cici", "ISBN003"));

        libraryService.registerMember(new Member("Dewi", "MEM001"));

        libraryService.registerMember(new Member("Eka", "MEM002"));

        do {

            System.out.println("\nSelamat datang di Sistem Manajemen Perpustakaan");

            System.out.println("Pilih jenis pengguna:");

            System.out.println("1. Member");

            System.out.println("2. Petugas Perpustakaan");

            System.out.println("3. Keluar");

            System.out.print("Masukkan pilihan Anda: ");

            mainChoice = scanner.nextInt();

            scanner.nextLine(); // Membersihkan buffer

            if (mainChoice == 1) {

                // Menu untuk Member

                int choice;

                do {

                    System.out.println("\n=== Menu Member ===");

                    System.out.println("1. Daftar Buku yang Tersedia");

                    System.out.println("2. Pinjam Buku");

                    System.out.println("3. Kembalikan Buku");

                    System.out.println("4. Cek Denda");

                    System.out.println("5. Bayar Denda");

                    System.out.println("6. Keluar ke Menu Utama");

                    System.out.print("Pilih opsi: ");

                    choice = scanner.nextInt();

                    scanner.nextLine(); // Membersihkan buffer

                    switch (choice) {

                        case 1:

                            libraryService.listAvailableBooks();

                            break;

                        case 2:

                            System.out.print("Masukkan ID anggota: ");

                            String borrowerId = scanner.nextLine();

                            System.out.print("Masukkan ISBN buku: ");

                            String borrowIsbn = scanner.nextLine();

                            libraryService.borrowBook(borrowerId, borrowIsbn);

                            break;

                        case 3:

                            System.out.print("Masukkan ID anggota: ");

                            String returnerId = scanner.nextLine();

                            System.out.print("Masukkan ISBN buku: ");

                            String returnIsbn = scanner.nextLine();

                            libraryService.returnBook(returnerId, returnIsbn);

                            break;

                        case 4:

                            System.out.print("Masukkan ID anggota: ");

                            String fineMemberId = scanner.nextLine();

                            Member fineMember = libraryService.findMember(fineMemberId);

                            if (fineMember != null) {

                                System.out.println("Total denda: Rp" + fineMember.getFine());

                            } else {

                                System.out.println("Anggota tidak ditemukan.");

                            }

                            break;

                        case 5:

                            System.out.print("Masukkan ID anggota: ");

                            String payMemberId = scanner.nextLine();

                            Member payMember = libraryService.findMember(payMemberId);

                            if (payMember != null) {

                                System.out.print("Masukkan jumlah pembayaran: Rp");

                                double amount = scanner.nextDouble();

                                scanner.nextLine(); // Membersihkan buffer

                                payMember.payFine(amount);

                            } else {

                                System.out.println("Anggota tidak ditemukan.");

                            }

                            break;

                        case 6:

                            System.out.println("Kembali ke menu utama...");

                            break;

                        default:

                            System.out.println("Opsi tidak valid.");

                    }

                } while (choice != 6);

            } else if (mainChoice == 2) {

                // Menu untuk Petugas Perpustakaan

                int choice;

                do {

                    System.out.println("\n=== Menu Petugas Perpustakaan ===");

                    System.out.println("1. Tambah Buku");

                    System.out.println("2. Hapus Buku");

                    System.out.println("3. Daftarkan Anggota");

                    System.out.println("4. Daftar Buku yang Tersedia");

                    System.out.println("5. Daftar Buku yang Dipinjam");

                    System.out.println("6. Daftar Anggota");

                    System.out.println("7. Keluar ke Menu Utama");

                    System.out.print("Pilih opsi: ");

                    choice = scanner.nextInt();

                    scanner.nextLine(); // Membersihkan buffer

                    switch (choice) {

                        case 1:

                            System.out.print("Masukkan judul buku: ");

                            String title = scanner.nextLine();

                            System.out.print("Masukkan penulis: ");

                            String author = scanner.nextLine();

                            System.out.print("Masukkan ISBN: ");

                            String isbn = scanner.nextLine();

                            libraryService.addBook(new Book(title, author, isbn));

                            break;

                        case 2:

                            System.out.print("Masukkan ISBN buku yang akan dihapus: ");

                            String removeIsbn = scanner.nextLine();

                            libraryService.removeBook(removeIsbn);

                            break;

                        case 3:

                            System.out.print("Masukkan nama anggota: ");

                            String memberName = scanner.nextLine();

                            System.out.print("Masukkan ID anggota: ");

                            String memberId = scanner.nextLine();

                            libraryService.registerMember(new Member(memberName, memberId));

                            break;

                        case 4:

                            libraryService.listAvailableBooks();

                            break;

                        case 5:

                            libraryService.listBorrowedBooks();

                            break;

                        case 6:

                            libraryService.listMembers();

                            break;

                        case 7:

                            System.out.println("Kembali ke menu utama...");

                            break;

                        default:

                            System.out.println("Opsi tidak valid.");

                    }

                } while (choice != 7);

            } else if (mainChoice == 3) {

                System.out.println("Terima kasih telah menggunakan sistem ini.");

            } else {

                System.out.println("Pilihan tidak valid. Silakan coba lagi.");

            }

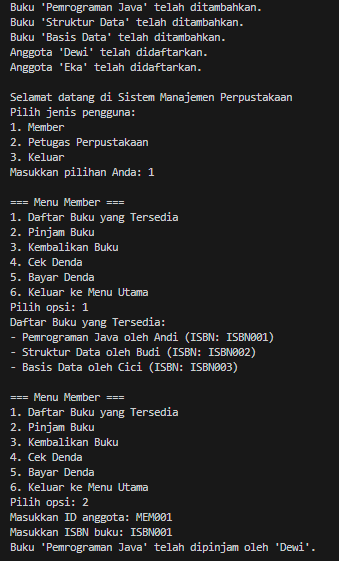
        } while (mainChoice != 3);

    }

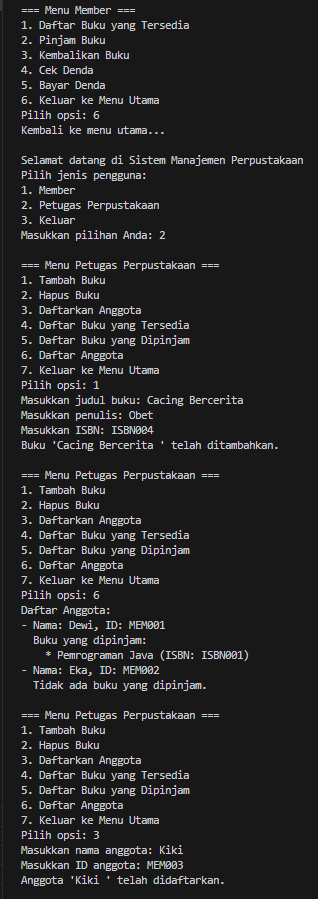
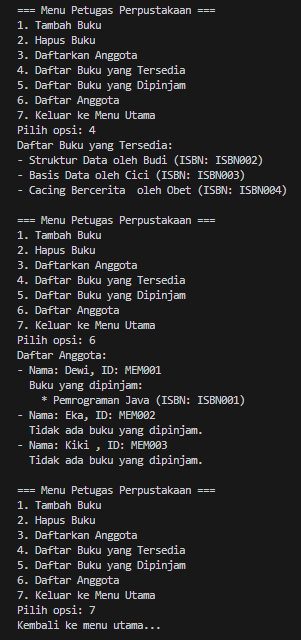
}

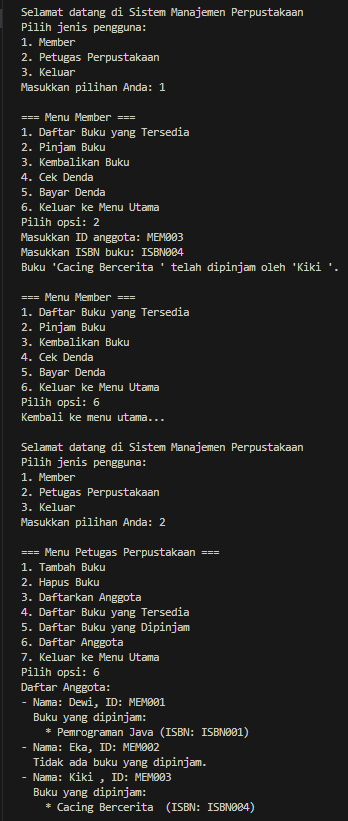
**Hasil Running**

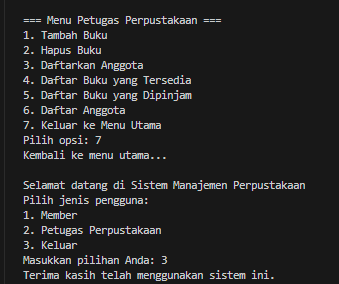
**No. 2**

****

**No. 3**

** **

****

****

**Link GitHub:** https://github.com/icen6/Tugas-Praktikum-PBO.git