

LAPORAN PRAKTIKUM STRUKTUR DATA
SISTEM PERPUSTAKAAN SEDERHANA



Oleh :

HABIBI IKRAMUL HUDA

NIM 2411532008

MATA KULIAH STRUKTUR DATA

DOSEN PENGAMPU : DR. WAHYUDI, S.T., M.T.

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

PADANG, MEI 2025

A. Pendahuluan

LinkedList pada Java adalah struktur data linier di mana setiap elemen, yang dikenal sebagai node, berisi data dan penunjuk ke node berikutnya dalam urutan. LinkedList tidak memiliki persyaratan untuk ukuran tetap seperti larik dan menawarkan penyisipan dan penghapusan node yang dinamis, terutama di ujung atau tengah daftar. Java memiliki class LinkedList yang sudah ditentukan sebelumnya di bawah paket java.util, yang bisa digunakan untuk mendefinisikan dan mengoperasikan linked list dengan mudah. Mengakses elemen melalui indeks, bagaimanapun, lebih lambat dibandingkan dengan larik, karena harus melalui daftar dari awal.

Stack adalah struktur data linier yang mengikuti prinsip Last In, First Out (LIFO). Ini berarti item terakhir yang ditambahkan ke dalam tumpukan akan menjadi item yang pertama kali dihapus. Hal ini mirip dengan tumpukan piring, yaitu hanya piring yang berada di atas yang dapat dihapus terlebih dahulu. Tumpukan di Java diimplementasikan melalui kelas Stack dalam paket java.util. Operasi utamanya adalah push() untuk menambahkan elemen, pop() untuk menghapus elemen paling atas, dan peek() untuk melihat elemen paling atas tanpa melakukan popping.

Queue mengikuti prinsip First In, First Out (FIFO). Elemen pertama yang ditambahkan adalah elemen pertama yang akan dihapus, seperti orang yang berdiri dalam antrian. Java menyediakan interface Queue, dan implementasi yang umum digunakan adalah LinkedList. Operasi utamanya adalah add() atau offer() untuk menyisipkan elemen pada akhir antrian, dan remove() atau poll() untuk mengambil elemen dari depan. Pengguna bisa menggunakan peek() untuk melihat elemen depan tanpa menghapusnya. Queue berguna dalam aplikasi seperti penjadwalan tugas atau melayani permintaan secara berurutan.

B. Tujuan

1. Mengetahui cara mengimplementasikan linkedlist, queue, dan stack untuk membuat sistem perpustakaan sederhana.

C. Langkah-Langkah

1. Buatlah package dan class baru dengan nama package pekan2 dan nama class Perpustakaan.
2. Import java util untuk linkedlist, stack, dan queue. Selain itu, import juga scanner agar pengguna bisa menginputkan pilihan.

```
package pekan2;

import java.util.LinkedList;
import java.util.Queue;
import java.util.Stack;
import java.util.Scanner;
```

3. Ketikkan class baru yaitu Buku. Kelas Buku berisi tiga variabel yaitu judul, pengarang, dan isbn) yang semuanya berupa string. Kelas ini juga menyertakan sebuah konstruktor. Konstruktor ini mengambil tiga parameter yaitu judul, pengarang, dan isbn dan menggunakan kata kunci ini untuk menetapkan nilai parameter ke variabel-variabel turunan objek.

```
class Buku {  
    String judul, pengarang, isbn;  
    Buku(String judul, String pengarang, String isbn) {  
        this.judul = judul;  
        this.pengarang = pengarang;  
        this.isbn = isbn;  
    }  
}
```

4. Ketikkan kelas baru yaitu Perpustakaan. Deklarasikan linkedlist LinkedList<Buku> koleksiBuku untuk menyimpan seluruh koleksi buku yang ada di perpustakaan. Setelah itu, deklarasikan queue Queue<Buku> Peminjaman yang akan merepresentasikan antrian peminjaman. Lalu, deklarasikan juga stack Stack<Buku> Pengembalian yang merepresentasikan tumpukan pengembalian buku.

```
class Perpustakaan {  
    LinkedList<Buku> koleksiBuku = new LinkedList<>();  
    Queue<Buku> Peminjaman = new LinkedList<>();  
    Stack<Buku> Pengembalian = new Stack<>();  
}
```

5. Ketikkan void tambahBuku(). Ketika method tambahBuku() dipanggil, method ini akan membuat objek Buku baru dan menambahkannya ke koleksi ini, yang merepresentasikan penambahan buku baru ke dalam inventaris perpustakaan.

```
void tambahBuku(String judul, String pengarang, String isbn) {  
    koleksiBuku.add(new Buku(judul, pengarang, isbn));  
}
```

6. Ketikkan void pinjamBuku(). Ketika pengguna menjalankan method pinjamBuku(judul), method ini akan mencari buku dengan judul yang diberikan di dalam koleksi. Jika ditemukan, method ini akan menambahkan buku tersebut ke antrian Peminjaman, mensimulasikan seseorang yang sedang meminjamnya.

```
void pinjamBuku(String judul) {  
    for (Buku buku : koleksiBuku) {  
        if (buku.judul.equals(judul)) {  
            Peminjaman.add(buku);  
            break;  
        }  
    }  
}
```

7. Ketikkan `void kembalikanBuku()`. Ketika `kembalikanBuku(judul)` dipanggil, method ini akan mencari antrian peminjaman untuk buku dengan judul yang cocok. Jika ditemukan, method ini akan mendorongnya ke tumpukan pengembalian, mensimulasikan bahwa buku tersebut telah dikembalikan dan menunggu untuk diproses kembali ke dalam sistem.

```
void kembalikanBuku(String judul) {  
    for (Buku buku : Peminjaman) {  
        if (buku.judul.equals(judul)) {  
            Pengembalian.push(buku);  
            break;  
        }  
    }  
}
```

8. Ketikkan main method untuk program ini. Main method ini merupakan titik masuk program Java dan berfungsi sebagai antarmuka berinteraksi dengan sistem Perpustakaan. Ketikkan objek Perpustakaan baru, yang berisi koleksi buku dan operasi peminjaman/pengembalian. Ketikkan juga scanner untuk membaca masukan pengguna dari console.

```
public static void main (String[] args) {  
    Perpustakaan perpustakaan = new Perpustakaan();  
    Scanner scanner = new Scanner(System.in);
```

9. Ketikkan perulangan while yang di mana di dalamnya terdapat kondisional if else. Di dalam perulangan while(true) tak terbatas, perulangan ini secara terus menerus menampilkan menu dengan empat pilihan kepada pengguna yaitu tambah buku, pinjam buku, kembalikan buku, dan keluar. Pengguna diminta untuk memilih salah satu dengan memasukkan nomor.

- Pilihan 1 akan meminta pengguna untuk memasukkan judul buku, pengarang, dan ISBN, lalu memanggil tambahBuku() untuk menambahkan buku tersebut ke dalam perpustakaan.
- Pilihan 2 meminta judul buku yang akan dipinjam, dan memanggil pinjamBuku() untuk menambahkannya ke dalam antrian peminjaman.
- Pilihan 3 meminta judul buku yang akan dikembalikan, dan memanggil kembalikanBuku() untuk mendorongnya ke tumpukan pengembalian.
- Pilihan 4 menghentikan perulangan dan mengakhiri program.

Setelah perulangan selesai, Pemindai ditutup untuk membebaskan sumber daya sistem.

```
while(true) {
    System.out.println("1. Tambah Buku\n2. Pinjam Buku\n3. Kembalikan Buku\n4. Keluar");
    System.out.print("Pilih opsi: ");
    int pilihan = scanner.nextInt();
    scanner.nextLine();
    if (pilihan == 1) {
        System.out.println("Masukkan Judul: ");
        String judul = scanner.nextLine();
        System.out.println("Masukkan Pengarang: ");
        String pengarang = scanner.nextLine();
        System.out.println("Masukkan ISBN: ");
        String isbn = scanner.nextLine();
        perpustakaan.tambahBuku(judul, pengarang, isbn);
    } else if (pilihan == 2) {
        System.out.println("Masukkan judul buku yang ingin dipinjam: ");
        String judul = scanner.nextLine();
        perpustakaan.pinjamBuku(judul);
    } else if (pilihan == 3) {
        System.out.println("Masukkan judul buku yang ingin dikembalikan: ");
        String judul = scanner.nextLine();
        perpustakaan.kembalikanBuku(judul);
    } else if (pilihan == 4) {
        break;
    }
}
scanner.close();
}
```

D. Kesimpulan

Program Java ini mensimulasikan sebuah sistem perpustakaan sederhana dengan menggunakan pemrograman berorientasi objek dan struktur data dasar. Program ini mendefinisikan kelas Buku untuk merepresentasikan objek buku dan kelas Perpustakaan untuk mengelola koleksi buku dengan menggunakan `LinkedList` untuk penyimpanan buku perpustakaan, `Queue` digunakan untuk buku-buku yang sedang dipinjam, dan `Stack` untuk buku-buku yang dikembalikan. Program ini dapat dijalankan pengguna untuk berinteraksi melalui menu konsol di mana mereka dapat menambahkan buku baru, meminjam buku berdasarkan judul, dan mengembalikan buku, dengan tindakan yang sesuai untuk memperbarui struktur data yang sesuai.