

LAPORAN PRAKTIKUM STRUKTUR DATA

PEKAN 2: Penerapan Struktur Data ArrayList dan Implementasi Studi

Kasus Manajemen Perpustakaan Menggunakan Java



OLEH

Abdullah Al Ramadhani (2411533016)

DOSEN PENGAMPU

DR. Wahyudi, S.T, M.T

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

2025

I. Pendahuluan

Struktur data yang umum digunakan dalam bahasa pemrograman Java adalah ArrayList. Dan praktikum ini memahami konsep dasar ArrayList serta cara penggunaannya dalam menyimpan, menambah, menghapus, dan menelusuri elemen data.

Selain itu, praktikum juga mencakup studi kasus berupa implementasi sistem perpustakaan sederhana yang memanfaatkan beberapa struktur data seperti LinkedList, Queue, dan Stack. Studi kasus ini bertujuan untuk memberikan gambaran nyata bagaimana struktur data dapat diintegrasikan dalam skenario dunia nyata, khususnya dalam pengelolaan koleksi buku, peminjaman, dan pengembalian.

II. Tujuan

Tujuan dari praktikum ini adalah sebagai berikut:

- a. untuk memahami konsep dasar dan cara kerja struktur data ArrayList, termasuk operasi dasar seperti penambahan, penghapusan, dan pencarian data.
- b. untuk mengembangkan keterampilan dalam mengimplementasikan ArrayList melalui studi kasus sederhana yang berkaitan dengan pengelolaan data, seperti perpustakaan.

III. Langkah-langkah pengerjaan

A. Program ArrayList1

1. Buatlah sebuah package baru dengan nama “**pekan2**”, lalu tambahkan sebuah class di dalamnya yang diberi nama ArrayList1.
2. Setelah itu, lakukan import terhadap kelas ArrayList yang berada dalam package java.util. Kelas ArrayList ini berfungsi untuk menyimpan data yang sifatnya fleksibel, artinya jumlah data bisa bertambah atau berkurang tergantung kebutuhan program.

```
package Pekan2;  
import java.util.ArrayList;
```

3. Buatlah sebuah class dengan nama **ArrayList1** sebagai tempat untuk menuliskan kode program. Dalam bahasa Java, setiap program setidaknya harus memiliki satu class atau lebih yang berfungsi sebagai wadah pengelolaan kode. Untuk menunjukkan titik awal eksekusi program, tambahkan method **main**, karena method ini merupakan bagian pertama yang akan dijalankan ketika program dieksekusi. Seluruh perintah yang ingin dijalankan harus ditulis di dalam method tersebut.

4. Lalu tambahkan method main yang digunakan sebagai titik awal program Java dijalankan. Semua perintah yang akan dieksekusi ditulis di dalam method ini.

```
ArrayList<String> list = new ArrayList<>();
```

5. Buat sebuah method **add()** untuk memasukkan data ke dalam objek list. Method ini merupakan fungsi bawaan dari class ArrayList yang berfungsi untuk menambahkan elemen baru ke akhir daftar. Dalam contoh ini, tiga buah string dimasukkan, yaitu "Apple", "Banana", dan "Cherry". Setiap kali **add()** dipanggil, elemen baru akan langsung disimpan di posisi indeks paling akhir dari ArrayList. Hal ini menunjukkan fleksibilitas ArrayList dalam menyimpan data tanpa perlu menentukan jumlah elemen sejak awal seperti pada array biasa.

```
list.add("Apple");  
list.add("Banana");  
list.add("Cherry");
```

6. Lalu untuk mencetak semua elemen dalam list, digunakan perulangan **for-each** yang akan mengambil setiap item dari list dan mencetaknya menggunakan **System.out.println()**.

```
for (String fruit : list) {  
    System.out.println(fruit);  
}
```

7. Setelah itu jalankan codingan, dan akan muncul output/tampilan dari codingan yang sudah dibuat tadi.

```
Apple  
Banana  
Cherry
```

B. Program Perpustakaan.

1. Buatlah sebuah class baru dengan nama "Perpustakaan".
2. Lakukan import **java.util.***.

```
package Pekan2;  
import java.util.*;
```

3. Setelah itu buat class dengan nama “Buku”, yang berfungsi untuk merepresentasikan objek buku. Dalam class ini, terdapat tiga atribut penting, yaitu **judul**, **pengarang**, dan **isbn**. Dan fungsi ini memakai kata kunci **this** untuk menyimpan data tersebut ke dalam class.

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

4. Buat class baru lagi dengan nama “Perpustakaan”, yang dimana akan menjadi pusat pengelolaan data koleksi **buku**, **peminjaman**, dan **pengembalian**. **LinkedList<Buku>** untuk menyimpan koleksi buku. Lalu **Queue<Buku>** untuk menyimpan daftar buku yang sedang dipinjam. Dan **Stack<Buku>** untuk menyimpan daftar buku yang dikembalikan.

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

5. Tambahkan method **void** baru “tambahBuku”, digunakan untuk menambahkan objek buku baru ke dalam “koleksi buku”.

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

6. Tambahkan method **void** baru “PinjamBuku”, yang digunakan untuk pengguna meminjam buku berdasarkan judul. Jika ditemukan buku dengan judul yang cocok di koleksiBuku, maka buku tersebut akan ditambahkan ke antrian Peminjaman.

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

7. Tambahkan method **void** baru lagi “kembalikanBuku”, yang dimana berfungsi untuk memindahkan buku dari antrian **Peminjaman** ke **Pengembalian**.

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

8. Setelah itu kita tulis method **main** sebagai entry point program dalam method main, dibuat **Perpustakaan** dan **Scanner** untuk menerima input dari pengguna. Lalu dibuat while loop yang terus berjalan sampai pengguna memilih keluar.

```
public static void main (String[] args) {
    Perpustakaan perpustakaan = new Perpustakaan();
    Scanner scanner = new Scanner(System.in);
    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.print("Masukkan judul: ");
            String judul = scanner.nextLine();
            System.out.print("Masukkan pengarang: ");
            String pengarang = scanner.nextLine();
            System.out.print("Masukkan ISBN: ");
            String isbn = scanner.nextLine();
            perpustakaan.tambahBuku(judul, pengarang, isbn);
        } else if (pilihan == 2) {
            System.out.print("Masukkan judul buku yang ingin dipinjam: ");
            String judul = scanner.nextLine();
            perpustakaan.PinjamBuku(judul);
        } else if (pilihan == 3) {
            System.out.print("Masukkan judul buku yang ingin dikembalikan: ");
            String judul = scanner.nextLine();
            perpustakaan.kembalikanBuku(judul);
        } else if (pilihan == 4) {
            break;
        }
    }
    scanner.close();
}
```

IV. Kesimpulan

Dari praktikum yang telah dilakukan, dapat disimpulkan bahwa ArrayList merupakan struktur data yang sangat berguna untuk menyimpan elemen data secara dinamis dan fleksibel. Selain itu, studi kasus sistem perpustakaan memperkenalkan cara integrasi berbagai struktur data dalam skenario nyata, seperti penggunaan LinkedList untuk koleksi buku, Queue untuk antrian peminjaman, dan Stack untuk proses pengembalian. Praktikum ini tidak hanya memperkuat pemahaman tentang struktur data di Java, tetapi juga mengembangkan kemampuan dalam merancang solusi programatis untuk permasalahan sederhana.

