

LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN

ARRAY LIST



OLEH:

AHMAD ZAHRAN

2311532004

DOSEN PENGAMPU:

DR. WAHYUDI, S.T, M.T

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

2025

A. Pendahuluan

Perkembangan teknologi informasi yang pesat menuntut efisiensi dalam pengolahan data, salah satunya melalui pemanfaatan struktur data yang tepat. Struktur data merupakan metode penyimpanan dan pengorganisasian data dalam suatu program komputer, yang dirancang untuk mempermudah akses, manipulasi, serta pengelolaan data tersebut. Dalam konteks pengembangan perangkat lunak, pemilihan struktur data yang sesuai sangat berpengaruh terhadap efisiensi algoritma dan performa aplikasi yang dihasilkan.

Bahasa pemrograman Java menyediakan berbagai pustaka yang mendukung implementasi struktur data, baik yang bersifat statis maupun dinamis. Di antaranya adalah ArrayList, LinkedList, Queue, dan Stack, yang masing-masing memiliki karakteristik serta keunggulan tersendiri dalam pengolahan data.

Penelitian atau praktik ini dilakukan untuk mengkaji secara langsung implementasi dari beberapa jenis struktur data tersebut melalui dua buah program. Program pertama merupakan simulasi dasar penggunaan ArrayList untuk melakukan operasi penambahan dan penghapusan elemen dalam list. Program kedua merupakan representasi dari sistem peminjaman dan pengembalian buku di sebuah perpustakaan digital sederhana, yang memanfaatkan struktur data LinkedList untuk menyimpan koleksi buku, Queue untuk mencatat antrian peminjaman, serta Stack dalam pencatatan pengembalian.

B. Tujuan praktikum

1. Memahami implementasi dan penggunaan ArrayList, LinkedList, Queue, dan Stack dalam Java.
2. Melatih kemampuan logika dalam pengelolaan data dinamis.
3. Mempelajari prinsip dasar manipulasi koleksi data dalam Java.

C. Langkah Kerja Praktikum

1. Class ArrayList1

- a. Buat package dan class baru, berikan nama package sebagai “pekan2”, dan nama class dengan nama ArrayListUTS.
- b. Selanjutnya kita mengimpor kelas ArrayList yang berasal dari package java.util. ArrayList agar bisa menggunakan class ArrayList
- c. Buatlah objek ArrayList dengan tipe data elemennya String dan diberi nama list.
`package pekan2;`

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

```
public class ArrayListUTS {
```

```
    public static void main(String[] args) {  
        ArrayList<String>list = new ArrayList<>();
```

- d. Lalu tambahkan element “Apple” dan “banana” ke dalam ArrayList.
- e. Kemudian tambahkan element “cherry” ke posisi indeks 1.

```
        list.add("Apple");  
        list.add("Banana");  
        list.add(1, "Cherry");
```

- f. Melakukan perulangan for untuk setiap elemen dalam ArrayList lalu mencetak elemen tersebut ke konsol.

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

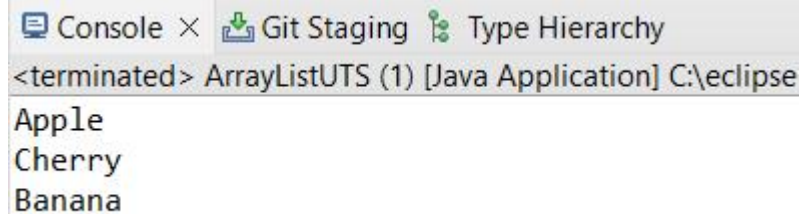
- g. Kode program lengkap

```

1 package pekan2;
2
3 import java.util.ArrayList;
4
5 public class ArrayListUTS {
6
7     public static void main(String[] args) {
8         ArrayList<String>list = new ArrayList<>();
9         list.add("Apple");
10        list.add("Banana");
11        list.add(1, "Cherry");
12        for (String fruit : list) {
13            System.out.println(fruit);
14        }
15    }
16 }

```

- h. jalankan program dan akan menghasilkan output sebagai berikut.



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The title bar of the console window reads '<terminated> ArrayListUTS (1) [Java Application] C:\eclipse'. The output text in the console is:

```

Apple
Cherry
Banana

```

2. Class Perpustakaan

- Buat *class* baru dengan nama “Perpustakaan”.
- Selanjutnya kita mengimpor `java.util.*` yang berasal dari package `java.util`. agar bisa menggunakan semua class dari java util

```
package pekan2;
```

```
import java.util.*;
```

- buat class dengan nama buku yang merupakan representasi dari entitas buku dalam perpustakaan.
- Buat tiga atribut utama yaitu judul, pengarang, dan isbn yang akan digunakan sebagai identifikasi. Konstruktor dari kelas ini digunakan untuk menginisialisasi atribut ketika objek Buku dibuat. Desain ini mencerminkan prinsip *object-oriented programming* di mana setiap buku diwakili sebagai objek mandiri

```

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

```

- e. Buat struktur Data dalam Kelas Perpustakaan yang akan di gunakan secara bersamaan seperti LinkedList digunakan sebagai koleksi utama buku karena mendukung penambahan dan penghapusan dengan efisiensi pada berbagai posisi tanpa perlu menggeser elemen seperti pada array, Queue (antrean) digunakan untuk merepresentasikan proses peminjaman buku yang bersifat First In First Out (FIFO), yakni siapa yang meminjam dulu akan diproses lebih dahulu, dan Stack (tumpukan) digunakan dalam proses pengembalian karena mencerminkan mekanisme Last In First Out (LIFO), di mana buku yang terakhir dikembalikan akan berada di atas tumpukan.

```

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

```

- f. Buat metode tambahBuku yang memungkinkan penambahan buku baru ke dalam koleksi perpustakaan. Objek baru dari kelas Buku dibuat berdasarkan input pengguna dan ditambahkan ke LinkedList koleksiBuku. Fungsi ini menggambarkan proses inventarisasi.

```

// use the linkedlist
void tambahBuku(String judul, String pengarang, String isbn) {
    koleksiBuku.add (new Buku(judul, pengarang, isbn));
}

```

- g. Buat metode pinjamBuku yang mana peminjaman buku dilakukan dengan mencocokkan judul buku dari input pengguna dengan daftar buku yang tersedia. Apabila ditemukan kecocokan, buku tersebut dimasukkan ke antrean Peminjaman.

```

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

```

- h. Untuk mengembalikan buku, buat metode kembalikanBuku yang mencerminkan proses pencarian buku di antrean Peminjaman, kemudian menambahkan ke

tumpukan Pengembalian. Hal ini menggambarkan bagaimana data dapat berpindah dari satu struktur ke struktur lain dalam sistem terintegrasi.

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

- i. Selanjutnya buat fungsi main yang nanti akan menyediakan antarmuka berbasis teks untuk interaksi pengguna dengan sistem perpustakaan. Struktur perulangan while digunakan agar menu terus tampil hingga pengguna memilih opsi keluar. Penggunaan Scanner memungkinkan input data secara dinamis.
- j. Buat sebuah objek yang berfungsi sebagai wadah utama untuk menyimpan data buku, peminjaman, dan pengembalian. Juga gunakan scanner untuk membaca input dari pengguna melalui konsol. System.in berarti input diambil dari keyboard pengguna.

```
public static void main (String[] args) {
    Perpustakaan perpustakaan = new Perpustakaan();
```

- k. buat perulangan tak terbatas yang akan digunakan supaya menu yang ditampilkan terus-menerus sampai pengguna memilih keluar dari program. Hanya bisa dihentikan dengan perintah break.

```
while (true) {
```

- l. Gunakan "System.out.println" untuk mencetak menu pilihan ke layar agar pengguna dapat memilih aksi yang diinginkan. Dan untuk menampilkan teks permintaan input ke pengguna agar mereka tahu harus mengetikkan angka pilihan.

```
System.out.println(" 1. Tambah Buku\n 2. Pinjam Buku\n " + "3. Kembalikan Buku\n 4. keluar");
System.out.print("Pilih opsi: ");
```

- m. tambahkan "int pilihan = scanner.nextInt();" untuk Membaca input pengguna dalam bentuk bilangan bulat, yang merepresentasikan pilihan menu. Dan untuk membersihkan sisa karakter newline di buffer yang tersisa setelah nextInt(). Ini penting supaya input string setelahnya tidak langsung terbaca kosong gunakan "scanner.nextLine();".

```
int pilihan = scanner.nextInt();
scanner.nextLine();
```

- n. Jika pengguna memilih menu nomor 1 maka program akan meminta pengguna mengetikkan judul buku, nama pengarang, dan nomor ISBN. Untuk ketiga itu program akan membacanya sebagai input string. Terakhir, program akan memanggil tambahBuku untuk menambahkan data buku baru ke dalam daftar koleksi yang disimpan di dalam objek perpustakaan.


```

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);
}

```

- o. Jika pengguna memilih pilihan 2 maka program akan meminta pengguna mengetikkan judul buku yang ingin dipinjam, lalu dibaca dan disimpan ke variabel judul. Dan akan memanggil metode pinjamBuku untuk memindahkan buku dengan judul tersebut dari koleksi ke daftar antrean peminjaman.

```

}else if (pilihan == 2) {
    System.out.print("Masukkan judul buku yang ingin dipinjam: ");
    String judul = scanner.nextLine();
    perpustakaan.pinjamBuku(judul);
}

```

- p. Jika pengguna memilih menu nomor 3 maka pengguna diminta mengetikkan judul buku yang akan dikembalikan. Input dibaca dan disimpan seperti sebelumnya. Dan akan memanggil metode kembalikanBuku dipanggil untuk memindahkan buku dari daftar peminjaman ke stack pengembalian.

```

}else if (pilihan == 3) {
    System.out.print("Masukkan judul buku yang ingin dikembalikan: ");
    String judul = scanner.nextLine();
    perpustakaan.kembalikanBuku(judul);
}

```

- q. Scanner ditutup untuk mengakhiri proses pembacaan input. Ini dilakukan untuk membebaskan resource setelah input tidak lagi diperlukan.

```

}else if (pilihan == 4) {
    break;
}
}
scanner.close();
}

```

- r. Jalankan program dan coba semua menu pilihan apakah sudah sesuai dengan program yang di buat tadi.

```

1. Tambah Buku
2. Pinjam Buku
3. Kembalikan Buku
4. keluar
Pilih opsi: 1
Masukkan judul: BUKU APA AJA
Masukkan pengarang: SIAPA SAJA
Masukkan ISBN: 19216811
1. Tambah Buku
2. Pinjam Buku
3. Kembalikan Buku
4. keluar
Pilih opsi: 2
Masukkan judul buku yang ingin dipinjam: BUKU APA AJA
1. Tambah Buku
2. Pinjam Buku
3. Kembalikan Buku
4. keluar
Pilih opsi: 3
Masukkan judul buku yang ingin dikembalikan: BUKU APA AJA
1. Tambah Buku
2. Pinjam Buku
3. Kembalikan Buku
4. keluar
Pilih opsi: 4

```

D. Kesimpulan

Dalam praktikum ini digunakan berbagai struktur data Java seperti ArrayList, LinkedList, Queue, dan Stack, masing-masing sesuai dengan karakteristik kebutuhan program. ArrayList dipakai untuk menyimpan data secara dinamis dan mudah diakses lewat indeks. LinkedList digunakan karena kemampuannya menambahkan dan menghapus elemen secara efisien tanpa perlu menggeser data lainnya. Queue diterapkan pada sistem peminjaman buku karena sesuai dengan prinsip FIFO, sementara Stack digunakan untuk proses pengembalian karena mengikuti prinsip LIFO.

Melalui program sistem perpustakaan sederhana ini, mahasiswa dapat memahami bagaimana pemilihan struktur data yang tepat memengaruhi efisiensi dan logika kerja program sesuai kebutuhan nyata..