

LAPORAN PRAKTIKUM STRUKTUR DATA  
IMPLEMENTASI ARRAYLIST, LINKEDLIST, QUEUE, DAN STACK  
PADA PEMROGRAMAN JAVA



Oleh :

DERIEL CHAERAHMAN

NIM 2411533007

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

ASISTEN PRAKTIKUM : RAHMAT DWIRIZKI OLDERS

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

2025

## A. Pendahuluan

Praktikum ini dilakukan untuk membuat program yang dapat menyimpan/mengelola data dengan implementasi struktur data dalam java yaitu ArrayList, Stack, Queue dan LinkedList. Mengimplementasikan dengan program java yang sudah dipelajari sebelumnya seperti if statement, looping, scanner class dan lainnya untuk menerima input yang nantinya disimpan pada ArrayList dalam pemrograman Java.

### 1. ArrayList

Merupakan bagian dari Java class yang digunakan untuk menyimpan data berupa list objek berbentuk array yang ukurannya dapat berubah secara dinamis. Mengimport dari package `java.util.ArrayList` sebagai bagian dari API Collection java. Memiliki banyak method untuk mengakses dan memodifikasi elemen yang tersimpan di dalam ArrayList. Elemen dalam ArrayList disimpan di lokasi memori bersebelahan.

Method :

- `add()` : untuk menambahkan data ke dalam ArrayList
- `get()` : untuk mengambil data yang tersimpan pada index tertentu
- `set()` : untuk menimpa data pada indeks tertentu
- `remove()` : untuk menghapus data pada indeks tertentu
- `size()` : untuk mengetahui panjang ArrayList
- `isEmpty()` : untuk memeriksa apakah ArrayList kosong atau tidak
- `indexOf()` : untuk mengetahui indeks dari suatu nilai
- `contains()` : untuk memeriksa apakah suatu nilai ada dalam ArrayList

### 2. Stack (Tumpukan)

Merupakan struktur data linear yang mengikuti aturan tertentu, yaitu LIFO (Last In First Out) atau FILO (First In Last Out). Jenisnya yaitu stack ukuran tetap dan dinamis. Dapat diimplementasikan menggunakan array ataupun LinkedList. Dapat digunakan untuk mengimplementasikan mekanisme 'pembatalan' untuk kembali ke kondisi sebelumnya/untuk membuat algoritma depth-first search dalam graf untuk backtracking. Kelebihannya yaitu implementasi linkedList dari sebuah stack dapat tumbuh dan menyusut sesuai kebutuhan saat runtime dan banyak digunakan pada JVM. Kekurangannya yaitu membutuhkan memori ekstra dan tidak mendukung pengaksesan secara acak.

Method :

- `Push()` : Menambahkan elemen baru pada tumpukan
- `Pop()` : Menghapus/ mengembalikan elemen teratas dari tumpukan
- `Peek()` : Mengembalikan elemen teratas pada stack, seperti `get()`.
- `isEmpty()` : Memeriksa apakah tumpukan kosong
- `Size()` : Menemukan jumlah elemen dalam tumpukan

### 3. Queue (Antrian)

Struktur data FIFO (First-In-First-Out) untuk pemrosesan berurutan, dimana data pertama yang ditambahkan ke dalam antrian akan dikeluarkan pertama. Visual operasinya dimana elemen ditambahkan pada satu ujung (`rear`) dan dihapus dari

ujung lainnya (front). Dapat diimplementasikan dengan array dan arraylist. Jenis - jenis antrian yaitu antrian masukan terbatas, output terbatas, melingkar, double ended dan antrian prioritas. Digunakan dalam algoritma dan aplikasi karena sederhana dan efisiensinya mengelola data.

Method :

- Enqueue() : Sisipkan, menambahkan elemen ke bagian belakang antrian
- Dequeue() : Hapus, menghapus dan mengembalikan elemen dari bagian depan antrian
- Peek() : Mengintip, mengembalikan elemen di bagian depan antrian tanpa menghapusnya, seperti get().
- Empty() : Kosong, memeriksa apakah antrian kosong
- Full() : Memeriksa apakah antrian penuh

#### 4. LinkedList (Senarai Berantai)

Merupakan struktur data berantai di mana setiap elemen (node) menyimpan referensi ke elemen berikutnya. Setiap node berisi data dan penunjuk ke node berikutnya dalam daftar. Cocok untuk operasi penambahan/penghapusan di tengah list, karena node-nodenya tidak disimpan secara berdekatan dalam memori. Jenisnya yaitu senarai berantai tunggal, ganda, melingkar dan ganda melingkar.

Operasi :

- add() : Menambah elemen pada node
- remove() : Menghapus elemen pada node
- get() : Mengambil elemen pada node
- set() : Mengganti elemen
- contains() : Mengecek isi node
- size() : Mengembalikan jumlah elemen dalam list

## B. Tujuan

Tujuan dari dilakukannya praktikum ini adalah :

1. Memahami dan mengaplikasikan ArrayList, LinkedList, Queue dan Stack dalam program java untuk menyimpan data.
2. Implementasi method ArrayList, LinkedList, Queue dan Stack pada program java.
3. Dapat mengaplikasikan statement (if, while), class (Scanner), dll. yang dipelajari sebelumnya ke dalam program.
4. Membuat program struktur data dengan metode OOP (Object Oriented Programming).

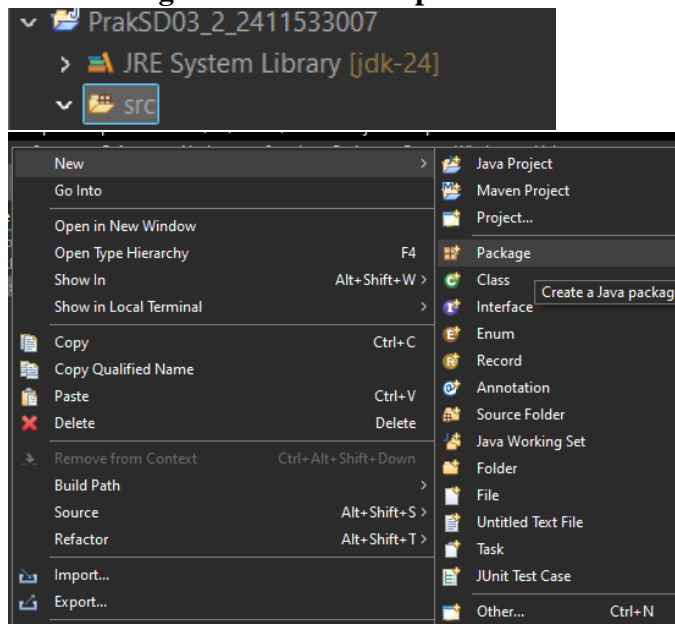
## C. Langkah kerja praktikum

### a. Alat dan Bahan

1. Perangkat computer atau laptop
2. Jaringan internet
3. IDE (Integrated Development Environment) direkomendasikan Eclipse IDE
4. Java JDK (Java Development Kit)

## b. Package pekan 2

1. Buat new package. **Buka java project** yang telah dibuat sebelumnya, lalu **klik kanan** pada folder 'src', setelahnya akan muncul list option, pilih 'new', lalu 'Package' dan beri nama 'pekan2'.



## c. Program ArrayList1

1. **Buat** terlebih dahulu new **class** dengan klik kanan pada package pekan2 dan beri nama 'ArrayList1', centang method public static void.
2. **Import** ArrayList class dari paket **Java.util.ArrayList** terlebih dahulu untuk menggunakan fungsi/method dari classnya.

```
1 package pekan2;
2 import java.util.ArrayList;
3
4 public class ArrayList1 {
```

3. **Deklarasi** Arraylist bertipe **string** dengan nama 'list'.

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

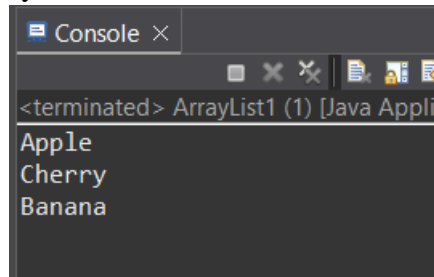
4. Gunakan **method add()** dari kelas ArrayList **untuk menambahkan data** ke dalam ArrayList. Gunakan method add() dengan spesifik lokasi index yaitu index ke-1 untuk menambahkan string.

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

5. Loop for-each untuk menelusuri setiap elemen String dalam list, dengan nama variabelnya 'fruit'. **Menampilkan** isi dari **ArrayList** melalui sysout menggunakan variabel for loop yaitu 'fruit'.

```
11    for (String fruit : list) {
12        System.out.println(fruit);
13    }
14 }
15 }
```

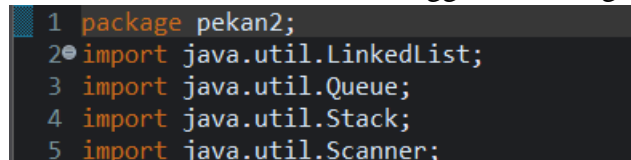
6. Output program. Isi dari ArrayList yang sebelumnya diinputkan menggunakan method add(), lalu ditampilkan menggunakan loop for each sysout.



```
<terminated> ArrayList1 (1) [Java Appli
Apple
Cherry
Banana
```

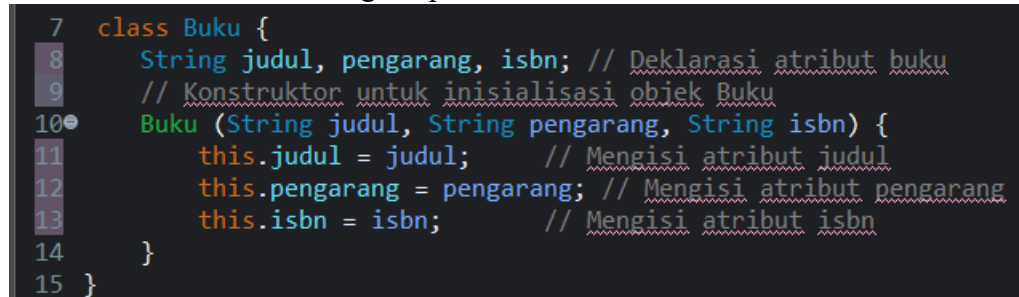
#### d. Program Perpustakaan

1. **Buat new class** dengan klik kanan pada package pekan2 dan beri nama 'Perpustakaan', centang method public static void.
2. **Import** LinkedList, Stack, Queue dan Scanner dari package **Java.util.** terlebih dahulu untuk menggunakan fungsi/method dari classnya.



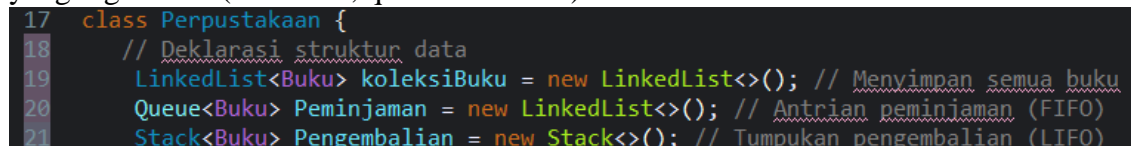
```
1 package pekan2;
2 import java.util.LinkedList;
3 import java.util.Queue;
4 import java.util.Stack;
5 import java.util.Scanner;
```

3. Deklarasi class dengan nama 'Buku'. Deklarasikan atribut judul, pengarang dan isbn bertipe String untuk menyimpan data buku. Konstruktor untuk inisialisasi objek buku yang didalamnya mengandung atribut instance, 'this' mengacu pada atribut instance.



```
7 class Buku {
8     String judul, pengarang, isbn; // Deklarasi atribut buku
9     // Konstruktor untuk inisialisasi objek Buku
10    Buku (String judul, String pengarang, String isbn) {
11        this.judul = judul; // Mengisi atribut judul
12        this.pengarang = pengarang; // Mengisi atribut pengarang
13        this.isbn = isbn; // Mengisi atribut isbn
14    }
15 }
```

4. Class 'Buku' berfungsi sebagai struktur data (data model) untuk mempresentasikan dan menyimpan informasi setiap objek 'Buku' dalam program 'perpustakaan'. Class ini menyimpan informasi berupa judul, pengarang dan isbn buku.
5. Deklarasi class dengan nama 'Perpustakaan'. Deklarasikan struktur data yang digunakan (linkedlist, queue dan stack).



```
17 class Perpustakaan {
18     // Deklarasi struktur data
19     LinkedList<Buku> koleksiBuku = new LinkedList<>(); // Menyimpan semua buku
20     Queue<Buku> Peminjaman = new LinkedList<>(); // Antrian peminjaman (FIFO)
21     Stack<Buku> Pengembalian = new Stack<>(); // Tumpukan pengembalian (LIFO)
```

6. Class 'Perpustakaan' berfungsi sebagai main class yang mengelola sistem perpustakaan seperti menambah, meminjam, dan mengembalikan buku. Kelas ini menyimpan struktur data dan metode-metode terkait operasional perpustakaan.

7. Method `tambahBuku`, menggunakan implementasi `LinkedList`. Menambahkan objek tersebut ke list `koleksiBuku` menggunakan method `add()`, yang mana membuat objek 'buku' dengan data dari parameter yaitu `judul`, `pengarang` dan `isbn`.

```
23 //Method menggunakan LinkedList
24 void tambahBuku(String judul, String pengarang, String isbn) {
25     koleksiBuku.add(new Buku(judul, pengarang, isbn));
26 }
```

8. Method `pinjamBuku`, menggunakan implementasi `Queue`. Menggunakan loop `for each` terhadap isi dari '`koleksiBuku`' untuk mencari buku yang sesuai dengan `judul`. Menambahkan buku tersebut ke antrian peminjaman (Deklarasi `Queue`) jika ditemukan (Bersifat `FIFO`).

```
28 //Method menggunakan queue
29 void pinjamBuku(String judul) {
30     for (Buku buku : koleksiBuku) { // Iterasi melalui semua buku
31         if (buku.judul.equals(judul)) { // Jika judul cocok
32             Peminjaman.add(buku); // Tambahkan ke antrian peminjaman
33             break; // Keluar dari loop setelah menemukan buku
34         }
35     }
36 }
```

9. Method `kembalikanBuku`, menggunakan implementasi `Stack`. Menggunakan loop `for each` terhadap isi dari '`Peminjaman`' untuk mencari buku yang sesuai dengan `judul`. Menambahkan buku tersebut ke antrian peminjaman (`Peminjaman`) jika ditemukan (Bersifat `LIFO`).

```
38 //Method menggunakan Stack
39 void kembalikanBuku(String judul) {
40     for (Buku buku : Peminjaman) { // Iterasi melalui buku yang dipinjam
41         if (buku.judul.equals(judul)) { // Jika judul cocok
42             Pengembalian.push(buku); // Taruh ke stack pengembalian
43             break; // Keluar dari loop setelah menemukan buku
44         }
45     }
46 }
```

10. Program utama yang di run. Membuat objek dari class `Perpustakaan` dan membuat objek untuk class `Scanner`.

```
48 public static void main(String[] args) {
49     Perpustakaan perpustakaan = new Perpustakaan(); // Buat objek perpustakaan
50     Scanner scanner = new Scanner(System.in); // Buat objek Scanner untuk input
```

11. While loop dengan kondisi '`true`' untuk melakukan perulangan yang menampilkan 'menu'. Meminta user untuk menginputkan yang ditampilkan pada variabel integer '`pilihan`' menggunakan fungsi dari class `Scanner`.

```
51 while (true) {
52     // Menampilkan menu
53     System.out.println("1. Tambah Buku\n2. Pinjam Buku\n" + "3. Kembalikan Buku\n");
54     System.out.println("Pilih opsi : ");
55     int pilihan = scanner.nextInt();
56     scanner.nextLine();
```

12. If Statement dengan kondisi jika input user equals 1, maka akan menjalankan method `tambahBuku`, yang meminta input berupa `judul`, `pengarang`, `isbn` terkait buku. Datanya ditambahkan ke dalam '`koleksiBuku`', yaitu implementasi `LinkedList` pada method `tambahBuku`.

```

57         if (pilihan == 1) {
58             // Proses tambah buku
59             System.out.println("Masukan judul: ");
60             String judul = scanner.nextLine();
61             System.out.println("Masukan pengarang: ");
62             String pengarang = scanner.nextLine();
63             System.out.println("Masukan ISBN: ");
64             String isbn = scanner.nextLine();
65             perpustakaan.tambahBuku(judul, pengarang, isbn);

```

13. Pada elif dengan kondisi jika input user equals 2, maka akan menjalankan method pinjamBuku, dimana user diminta untuk input judul buku yang ingin dipinjam, lalu menambahkan 'buku' tersebut ke dalam 'Peminjaman', yaitu implementasi queue pada method pinjamBuku.

```

66         } else if (pilihan == 2) {
67             // Proses pinjam buku
68             System.out.println("Masukan judul buku yang ingin dipinjam: ");
69             String judul = scanner.nextLine();
70             perpustakaan.pinjamBuku(judul);

```

14. Pada elif dengan kondisi jika input user equals 3, maka akan menjalankan method kembalikanBuku, dimana user diminta untuk input judul buku yang ingin dikembalikan, lalu menambahkan 'buku' tersebut ke dalam 'kembalikanBuku', yaitu implementasi Stack method kembalikanBuku.

```

71         } else if (pilihan == 3) {
72             // Proses kembalikan buku
73             System.out.println("Masukan judul buku yang ingin diikembalilkan: ");
74             String judul = scanner.nextLine();
75             perpustakaan.kembalikanBuku(judul);

```

15. Pada elif dengan kondisi jika input user equals 4, program akan mengakhiri loop menggunakan 'break'.

```

76         } else if (pilihan == 4) {
77             break; // Keluar dari loop dan program
78         }

```

16. Menutup objek Scanner setelah program diakhiri untuk mencegah memori leak.

```

79         }
80         scanner.close();
81     }
82 }

```

17. Output dari program perpustakaan.

```
Console X
<terminated> Perpustakaan [Java Application] C:\Program Files
1. Tambah Buku
2. Pinjam Buku
3. Kembalikan Buku
4. Keluar
Pilih opsi :
1
Masukan judul:
Laskar Pelangi
Masukan pengarang:
Andrea Hirata
Masukan ISBN:
9793062797
1. Tambah Buku
2. Pinjam Buku
3. Kembalikan Buku
4. Keluar
Pilih opsi :
2
Masukan judul buku yang ingin dipinjam:
Laskar Pelangi
1. Tambah Buku
2. Pinjam Buku
3. Kembalikan Buku
4. Keluar
Pilih opsi :
3
Masukan judul buku yang ingin dikembalikan:
Laskar Pelangi
1. Tambah Buku
2. Pinjam Buku
3. Kembalikan Buku
4. Keluar
Pilih opsi :
4
```

#### D. Kesimpulan

Setelah melakukan praktikum ini dapat memahami dan mengimplementasikan struktur data dari class java collections seperti ArrayList, Queue, Stack dan LinkedList serta methodnya. Membuat program yang lebih kompleks dan interaktif dengan konsep OOP berupa program Perpustakaan. Mengaplikasikan Scanner class untuk menerima input user lalu menyimpan datanya dalam ArrayList/Queue/Stack/LinkedList. Menggunakan while loop dalam method main untuk membuat program lebih interaktif.