

LAPORAN PRAKTIKUM 2

STRUKTUR DATA



Oleh:
Ibrahim Mousa Dhani
2411532010

Dosen Pengampu : Dr. Wahyudi S.T, M.T.

Mata Kuliah : Struktur Data

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

PADANG

A. Pendahuluan

Struktur data merupakan salah satu fondasi penting dalam pemrograman yang digunakan untuk menyimpan dan mengelola data secara efisien. Java sebagai bahasa pemrograman berorientasi objek menyediakan berbagai struktur data dinamis melalui library *Collection Framework*, seperti ArrayList, LinkedList, Queue, dan Stack.

1. ArrayList adalah struktur data berbasis array dinamis yang termasuk dalam Java Collection Framework. ArrayList memungkinkan penambahan, penghapusan, dan pengaksesan elemen secara cepat menggunakan indeks.
2. LinkedList adalah struktur data berbasis node, di mana setiap elemen (node) menyimpan data dan referensi ke elemen berikutnya. LinkedList lebih efisien dalam operasi penambahan dan penghapusan dibandingkan ArrayList, terutama pada awal atau tengah daftar.
3. Queue adalah struktur data linear yang mengikuti prinsip FIFO (*First In, First Out*), artinya elemen yang pertama kali masuk akan menjadi yang pertama kali keluar.
4. Stack adalah struktur data linear yang mengikuti prinsip LIFO (*Last In, First Out*), yaitu elemen terakhir yang dimasukkan akan menjadi yang pertama keluar

B. Tujuan Praktikum

1. Memahami konsep dan penggunaan struktur data dinamis seperti ArrayList, LinkedList, Queue, dan Stack di Java.
2. Mengimplementasikan operasi dasar seperti penambahan, peminjaman, dan pengembalian data menggunakan struktur data tersebut.
3. Melatih kemampuan mahasiswa dalam memahami konsep dasar struktur data serta mengaplikasikannya dalam situasi pemrograman nyata.

C. Langkah Langkah

a. ArrayList1

1. Pertama buat class baru terlebih dahulu pada package pekan 2, dan beri nama ArrayList1
2. Setelah itu Import library ArrayList, dengan cara mengetikkan 'import.java.util.ArrayList;'

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

3. Lalu buat ArrayList dengan nama 'list' dan bertipe data String

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

4. Baru kemudian kita mulai menambahkan elemen yang kita mau ke dalam ArrayList, disini saya menambahkan elemen pertama yaitu Apple, lalu elemen yang kedua Banana setelah Apple, dan yang ketiga menyisipkan Cherry di indeks (1) atau dengan kata lain diantara Apple dan Banana

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

5. Selanjutnya kita akan menampilkan data dalam Arraylist yang diberi nama list tadi, dengan menggunakan *for-each* untuk menampilkan setiap element yang ada di dalam ArrayList nya satu per satu. Lalu buat variable sementara, disini misalnya fruit untuk menyimpan isi dari setiap elemen saat iterasi berlangsung, dan terakhir tampilkan isi fruit ke layer. Bisa dilihat pada gambar berikut

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

6. Setelah semua benar dan tidak ada eror, maka jalankan program, lalu akan menghasilkan output sebagai berikut

```
<terminated>  
Apple  
Cherry  
Banana
```

7. Berikut kode program versi lengkapnya

```
1 package pekan2;
2 import java.util.ArrayList;
3 public class ArrayList1 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         ArrayList<String> list = new ArrayList<>();
8         list.add("Apple");
9         list.add("Banana");
10        list.add(1, "Cherry");
11
12        for (String fruit : list) {
13            System.out.println(fruit);
14        }
15    }
```

b. Perpustakaan

1. Buat class baru pada package 2, dan beri nama Perpustakaan
2. Lalu Import semua kelas yang dibutuhkan di dalam paket java.util, yaitu linked list, queue, dan stack. Atau bisa ditulis juga dengan *

```
1 package pekan2;
2 import java.util.*;
```

3. Selanjutnya Deklarasi kelas buku yang digunakan untuk merepresentasikan objek buku. Buku memiliki tiga atribut: judul, pengarang, dan isbn.. Buat juga konstruktor kelas buku untuk menginisialisasi ketiga atribut tersebut.

```
class Buku {
    String judul, pengarang, isbn;

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

4. Deklarasi kelas Perpustakaan, yang berfungsi untuk mengelola koleksi buku (LinkedList) dan memfasilitasi peminjaman (Queue) serta pengembalian buku (Stack).

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

- Setelah itu kita akan membuat method `tambahBuku`, Method ini menerima judul, pengarang, dan isbn, lalu membuat objek `Buku` baru dan menambahkannya ke dalam `koleksiBuku`.

```
// Menambahkan buku ke linked list
void tambahBuku(String judul, String pengarang, String isbn) {
    koleksiBuku.add(new Buku(judul, pengarang, isbn));
}
```

- Buat method `pinjamBuku` untuk meminjam buku berdasarkan judul. Nanti pencarian buku dilakukan pada `koleksiBuku`. Jika ditemukan buku yang sesuai, buku tersebut ditambahkan ke dalam `Peminjaman` (Queue).

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

- yang ketiga buat method `kembalikanBuku` untuk mengembalikan buku yang sudah dipinjam. Dan pencariannya akan dilakukan di `Peminjaman`, dan jika ditemukan buku yang sesuai, buku tersebut dipindahkan ke dalam `Pengembalian` (Stack).

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

- Selanjutnya buat program utama nya (main), pertama kita inisialisasi objek `Perpustakaan` dan `Scanner`, `Perpustakaan` dibuat dengan memanggil konstruktor dari kelas `Perpustakaan` yang telah kita buat sebelumnya.
- Lalu buat perulangan tak terbatas menggunakan `while` selama kondisinya tetap benar (`true`), dan program akan terus meminta input lewat scanner hingga user memilih untuk 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\n" + "3. Kembalikan Buku\n4. Keluar");
        System.out.print("Pilih opsi: ");

        int pilihan = scanner.nextInt();
        scanner.nextLine(); // consume newline
    }
}
```

10. Jika user memasukkan angka 1, maka program akan meminta pengguna untuk memasukkan tiga informasi tentang buku yang ingin ditambahkan yaitu Judul buku, Pengarang, dan ISBN. Setelah itu, metode tambahBuku() pada objek perpustakaan dipanggil untuk menambahkan buku baru ke dalam koleksi 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);
}
```

11. Jika user memasukkan angka 2, maka program akan meminta pengguna untuk memasukkan judul buku yang ingin dipinjam. Kemudian, pinjamBuku() dipanggil untuk menambahkan buku ke dalam antrian peminjaman (Peminjaman), jika buku ditemukan di koleksi perpustakaan.

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

12. Jika user memasukkan angka 3, maka program akan meminta pengguna untuk memasukkan judul buku yang ingin dikembalikan. Kemudian, kembalikanBuku() dipanggil untuk memindahkan buku dari antrian peminjaman ke stack pengembalian (Pengembalian).

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

13. Jika user memasukkan angka 4, maka program akan berhenti/berakhir, dan setelah selesai jangan lupa untuk menutup scanner

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

14. Setelah semua benar dan tidak ada eror, maka jalankan program, lalu akan menghasilkan output sebagai berikut

- Disini saya menambahkan buku Madilog dengan pengarang tan Malaka dan isbn 1001, lalu buku jungle book dengan pengarang tom alex dan isbn 2002, kemudian masuk ke program 2, saya meminjam kedua buku tersebut. Masuk ke program 3 saya mengembalikan buku madilog, lalu saya mengakhiri/keluar program dengan masuk ke program ke 4.

```
1. Tambah Buku
2. Pinjam Buku
3. Kembalikan Buku
4. Keluar
Pilih opsi: 1
Masukkan judul: Madilog
Masukkan pengarang: Tan Malaka
Masukkan ISBN: 1001
1. Tambah Buku
2. Pinjam Buku
3. Kembalikan Buku
4. Keluar
Pilih opsi: 1
Masukkan judul: Jungle Book
Masukkan pengarang: Tom Alex
Masukkan ISBN: 2002
```

```
1. Tambah Buku
2. Pinjam Buku
3. Kembalikan Buku
4. Keluar
Pilih opsi: 3
Masukkan judul buku yang ingin dikembalikan: Madilog
```

```
1. Tambah Buku
2. Pinjam Buku
3. Kembalikan Buku
4. Keluar
Pilih opsi: 4
```

```
1. Tambah Buku
2. Pinjam Buku
3. Kembalikan Buku
4. Keluar
Pilih opsi: 2
Masukkan judul buku yang ingin dipinjam: Madilog
1. Tambah Buku
2. Pinjam Buku
3. Kembalikan Buku
4. Keluar
Pilih opsi: 2
Masukkan judul buku yang ingin dipinjam: Jungle Book
1. Tambah Buku
2. Pinjam Buku
3. Kembalikan Buku
4. Keluar
```

15. Berikut kode program versi lengkapnya

```
1 package pekan2;
2 import java.util.*;
3
4 class Buku {
5     String judul, pengarang, isbn;
6
7     Buku(String judul, String pengarang, String isbn) {
8         this.judul = judul;
9         this.pengarang = pengarang;
10        this.isbn = isbn;
11    }
12 }
13
14 class Perpustakaan {
15     LinkedList<Buku> koleksiBuku = new LinkedList<>();
16     Queue<Buku> Peminjaman = new LinkedList<>();
17     Stack<Buku> Pengembalian = new Stack<>();
18
19     // Menambahkan buku ke linked list
20     void tambahBuku(String judul, String pengarang, String isbn) {
21         koleksiBuku.add(new Buku(judul, pengarang, isbn));
22     }
23
24     // Menggunakan queue
25     void pinjamBuku(String judul) {
26         for (Buku buku : koleksiBuku) {
27             if (buku.judul.equals(judul)) {
28                 Peminjaman.add(buku);
29                 break;
30             }
31         }
32     }
33
34     // menggunakan stack
35     void kembalikanBuku(String judul) {
36         for (Buku buku : Peminjaman) {
37             if (buku.judul.equals(judul)) {
38                 Pengembalian.push(buku);
39                 break;
40             }
41         }
42     }
43
44     public static void main(String[] args) {
45         Perpustakaan perpustakaan = new Perpustakaan();
46         Scanner scanner = new Scanner(System.in);
47
48         while (true) {
49             System.out.println("1. Tambah Buku\n2. Pinjam Buku\n" + "3. Kembalikan Buku\n4. Keluar");
50             System.out.print("Pilih opsi: ");
51
52             int pilihan = scanner.nextInt();
53             scanner.nextLine(); // consume newline
54
55             if (pilihan == 1) {
56                 System.out.print("Masukkan judul: ");
57                 String judul = scanner.nextLine();
58                 System.out.print("Masukkan pengarang: ");
59                 String pengarang = scanner.nextLine();
60                 System.out.print("Masukkan ISBN: ");
61                 String isbn = scanner.nextLine();
62
63                 perpustakaan.tambahBuku(judul, pengarang, isbn);
64             } else if (pilihan == 2) {
65                 System.out.print("Masukkan judul buku yang ingin dipinjam: ");
66                 String judul = scanner.nextLine();
67                 perpustakaan.pinjamBuku(judul);
68             } else if (pilihan == 3) {
69                 System.out.print("Masukkan judul buku yang ingin dikembalikan: ");
70                 String judul = scanner.nextLine();
71                 perpustakaan.kembalikanBuku(judul);
72             } else if (pilihan == 4) {
73                 break;
74             }
75         }
76         scanner.close();
77     }
78 }
```


D. Kesimpulan

Berdasarkan praktikum yang telah dilakukan dapat disimpulkan bahwa penggunaan struktur data dinamis seperti ArrayList, LinkedList, Queue, dan Stack memiliki peran penting dalam pengelolaan data pada pemrograman Java. ArrayList sangat berguna untuk menyimpan dan mengakses data secara berurutan melalui indeks, sementara LinkedList lebih fleksibel dalam penambahan dan penghapusan elemen. Queue efektif untuk mengelola proses antrian seperti peminjaman buku, karena mengikuti prinsip FIFO (First In, First Out). Sedangkan Stack digunakan dalam proses yang memerlukan urutan LIFO (Last In, First Out), seperti pengembalian buku.

Melalui penerapan langsung dalam simulasi sistem perpustakaan, praktikum ini memberikan pemahaman yang lebih nyata tentang bagaimana masing-masing struktur data bekerja dan bagaimana memilih struktur yang sesuai untuk kebutuhan tertentu. Dengan demikian, mahasiswa diharapkan mampu mengembangkan program yang efisien dan terstruktur dengan baik menggunakan struktur data yang tepat.