

LAPORAN PRAKTIKUM STRUKTUR DATA  
*“Linked List”*



OLEH :  
EZZA ADDINI  
2311532001

DOSEN PENGAMPU: Dr. WAHYUDI, M. T.

DEPARTEMEN INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS ANDALAS  
PADANG  
2024

## A. PENDAHULUAN

Linked list adalah salah satu jenis struktur data yang digunakan untuk menyimpan kumpulan data secara dinamis. Dalam linked list, setiap elemen adalah sebuah node yang menyimpan dua bagian utama: data dan referensi (atau pointer) ke node berikutnya dalam daftar. Dalam Java, linked list biasanya diimplementasikan menggunakan kelas dan objek.

1. Node Class: Kelas ini mendefinisikan struktur dasar dari setiap elemen dalam linked list, yang berisi data dan referensi ke node berikutnya
2. Linked List Class: Kelas ini berisi metode untuk menambah, menghapus, dan menampilkan elemen dalam linked list.
3. Main Method: Bagian ini digunakan untuk menguji kelas LinkedList dengan menambah dan menghapus elemen serta menampilkan isi linked list.

Dalam Java, kita bisa membuat linked list sendiri atau menggunakan linked list bawaan dari pustaka Java Collections Framework (JCF). Berikut adalah perbedaan utama antara linked list buatan sendiri dengan linked list bawaan dari Java:

1. Linked list buatan sendiri

Linked list buatan sendiri memberikan kontrol penuh atas struktur dan perilaku linked list. Namun, ini juga berarti lebih banyak pekerjaan yang perlu dilakukan untuk mengimplementasikan dan memelihara linked list tersebut. Lebih banyak barisan kode yang akan dibuat untuk menginputkan sebuah nilai ke dalam list-nya.

2. Linked list bawaan dari Java

Java Collections Framework menyediakan implementasi yang efisien dan serbaguna dari linked list melalui kelas `java.util.LinkedList`. Ini adalah linked list ganda (doubly linked list) yang menyediakan berbagai metode untuk manipulasi koleksi. Kemudahan penggunaan, efisiensi, dan berbagai fitur yang disediakan sangat berguna untuk pengembangan cepat dan mengurangi kompleksitas kode.

## B. TUJUAN

Tujuan dari praktikum ini adalah:

1. Membuat program Single Linked List buatan manual.
2. Membuat program Single Linked List bawaan dari Java.
3. Membuat program Double Linked List.

### C. LANGKAH KERJA

#### a. Program Single Linked List Buatan Manual

1. Buat package dan class baru untuk membuat list node, namakan sesuai dengan yang diperintahkan atau yg diinginkan.

```
1 package Pekan4;  
2  
3 public class ListNode {
```

2. Deklarasikan variabel data.

```
public int data; //data stored in this node  
public ListNode next; // link to next node in the list
```

3. Buat konstruktor default untuk kelas list node tersebut.

```
public ListNode() {  
    this(0, null);  
}
```

4. Deklarasikan konstruktor default dan inisialisasi variabel instance.

```
public ListNode(int data) {  
    this.data = data;  
    this.next = null;  
}
```

5. Deklarasikan konstruktor default dengan dua parameter, serta inisialisasi variabel instance.

```
public ListNode(int data, ListNode next) {  
    this.data = data;  
    this.next = next;  
}
```

6. Buat class baru untuk single linked list.

```
1 package Pekan4;  
2  
3 public class SingleLinkedList1 {
```

7. Deklarasikan nama single linked list sesuai dengan nama class list node sebelumnya.

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

8. Buat sintaks untuk menginputkan nilai ke dalam linked list.

```
ListNode list = new ListNode();  
list.data = 42;  
list.next = new ListNode();  
list.next.data = -3;  
list.next.next = new ListNode();  
list.next.next.data = 17;  
list.next.next.next = null;
```

9. Buat sintaks untuk menampilkan output.

```
System.out.println(list.data + " " + list.next.data + " " + list.next.next.data);
```

10. Jalankan program.

```
Console ×  
<terminated> Sir  
42 -3 17
```

b. Program Single Linked List Bawaan Java

1. Buat class baru dan namakan sesuai yang diinginkan, lalu buat "import java.util.LinkedList;" pada baris setelah package.

```
1 package Pekan4;  
2  
3 import java.util.LinkedList;  
4  
5 public class SingleLinkedList2 {
```

2. Deklarasikan tipe data yang akan digunakan.

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

3. Buat sintaks untuk menginputkan nilai ke dalam linked list dengan metode *add*, kemudian buat sintaks untuk menampilkan output.

```
list.add(42);  
list.add(-3);  
list.add(17);  
System.out.println("LinkedList: " + list);
```

4. Buat sintaks untuk mengonversi sebuah objek LinkedList menjadi sebuah array, kemudian buat sintaks untuk menampilkan elemen-elemennya.

```
Object[] a = list.toArray();  
System.out.println("After converted LinkedList to Array: ");
```

5. Buat sintaks untuk mencetak setiap elemen dari array a menggunakan loop for-each. Kemudian buat sintaks untuk menampilkan outputnya.

```
for (Object element : a)  
    System.out.println(element + " ");
```

6. Jalankan program.

```
Console x  
<terminated> SingleLinkedList2 [Java Application] C:\Users  
LinkedList: [42, -3, 17]  
After converted LinkedList to Array:  
42  
-3  
17
```

c. Program Double Linked List

1. Buat class baru dan namakan sesuai perintah atau yang diinginkan.

```
1 package Pekan4;  
2  
3 public class DoubleLinkedList1 {
```

2. Deklarasikan variabel head.

```
static Node head;
```

3. Deklarasikan kelas node.

```
class Node {  
    int data;  
    Node prev;  
    Node next;  
    Node(int d) { data = d;  
    }  
}
```

4. Deklarasikan metode push.

```
public void push (int new_data) {
```

5. Buat sintaks untuk node baru, kemudian tentukan yang mana next dan prev.

```
Node new_Node = new Node(new_data);  
new_Node.next = head;  
new_Node.prev = null;
```

6. Buat sintaks if untuk memeriksa apakah head tidak null, kemudian tentukan nilai prev node dan head untuk node barunya.

```
if (head != null)  
    head.prev = new_Node;  
head = new_Node;
```

7. Deklarasi metode display dan variabel sementara.

```
public void display(Node head) {  
    Node temp = head;
```

8. Buat sintaks untuk menampilkan isi dari sebuah doubly linked list dengan cara mencetak setiap elemen secara berurutan.

```
while (temp != null) {  
    System.out.println(temp.data + " --> ");  
    temp = temp.next;  
}
```

9. Buat sintaks untuk menampilkan output Null apabila while loop bernilai false.

```
}  
System.out.println("Null");
```

10. Deklarasikan nama double linked list.

```
public static void main(String[] args) {  
    DoubleLinkedList1 dll = new DoubleLinkedList1();
```

11. Inisialisasi nilai yang akan diinputkan.

```
        dll.push(42);  
        dll.push(-3);  
        dll.push(17);  
        dll.display(head);  
    }
```

12. Jalankan program.

```
Console ×  
<terminated> Do  
17 -->  
-3 -->  
42 -->  
Null
```

#### D. KESIMPULAN

Pada praktikum kali ini, kita membuat program single linked list dengan metode buatan dan bawaan dari Java, serta membuat program double linked list. Untuk menjalankan program single linked list buatan, dibutuhkan kelas yang berisi node untuk dijalankan pada class single linked list buatan. Baru setelahnya dibuat class untuk menjalankan single linked list-nya. Kemudian, dibuat juga program single linked list menggunakan linked list bawaan dari pustaka Java Collections Framework (JCF). Terakhir, membuat program double linked list.