Laporan Praktikum Struktur Data

**Linked List**



Disusun Oleh :

**FIKHRI HANIF**

**2311533007**

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

Departemen Informatika

Fakultas Teknologi Informasi

Universitas Andalas

Tahun 2024

1. **Tujuan Praktikum**
2. Memahami konsep dan implementasi struktur data linked list
3. Menerapkan berbagai jenis linked list, yaitu single linked list dan double linked list.
4. Mempelajari cara konversi antara linked list dan array
5. **Pendahuluan**

Laporan ini menjelaskan dua program Java yang berfungsi untuk mendemonstrasikan penggunaan struktur data LinkedList. Program pertama (SingleLinkedList1) menggunakan implementasi ListNode buatan sendiri, sedangkan program kedua (SingleLinkedList2) menggunakan LinkedList dan program terakhir menggunakan DoubleLinkedList (DoubleLinkedList3).

1. **Metode Praktikum**
2. **ListNode**

package Pekan4;

public class ListNode {

public int data;

public ListNode next;

public ListNode() {

this (0, null);

}

public ListNode(int data) {

this.data = data;

this.next = null;

}

public ListNode(int data, ListNode next) {

this.data = data;

this.next = next;

}

}

List Node disini berfungsi sebagai menginisialisasi referensi **next** ke **null**, menunjukkan bahwa node ini tidak terhubung ke node lain saat dibuat. Di program tersebut juga terdapat data node dengan nilai yang telah diberikan dan referensi **next** dengan node yang diberikan.

Struktur di atas membentuk linked list dengan tiga node:

* Node pertama dengan data **1** dan mengarah ke node kedua.
* Node kedua dengan data **2** dan mengarah ke node ketiga.
* Node ketiga dengan data **3** dan tidak mengarah ke node lain (next = null).

Dengan kata lain, kelas **ListNode** mendefinisikan unit dasar dari linked list, yang bisa digunakan untuk membangun dan mengelola linked list dengan cara menghubungkan beberapa node melalui referensi **next** mereka.{

1. **SingleLinkedList1**

Program ini membuat dan mengelola sebuah linked list sederhana dengan tiga node. Setiap node diwakili oleh kelas **ListNode**.

public class SingleLinkedList1 {

public static void main(String[] args) {

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;

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

}

}

Program mencetak nilai dari masing-masing node yang tersimpan dalam linked list.

Berikut output dari program tersebut:

42 -3 17

1. **SingleLinkedList2**

Program ini menggunakan **LinkedList** dari Java Collection Framework untuk menyimpan tiga elemen, mengubah linked list menjadi array, dan mencetak elemen-elemen tersebut.

import java.util.LinkedList;

public class SingleLinkedList2 {

public static void main(String[] args) {

LinkedList<Integer> list= new LinkedList<Integer>();

list.add(42);

list.add(-3);

list.add(17);

System.***out***.println("LinkedList : "+ list);

Object[] a = list.toArray();

System.***out***.println("After converted LinkeList to Array: ");

for(Object element : a)

System.***out***.print(element+" ");

}

}

Program mencetak isi linked list, kemudian mencetak elemen-elemen array yang berasal dari linked list tersebut.

Berikut output dari program tersebut:

LinkedList : [42, -3, 17]

After converted LinkeList to Array:

42 -3 17

1. **DoubleLinkedList**

Pada implementasi ini, kita membuat kelas **DoubleLinkedList** untuk merepresentasikan double linked list.

public class DoubleLinkedList3 {

static Node *head*;

class Node {

int data;

Node prev;

Node next;

Node (int d) { data = d;}

}

public void push(int new\_data) {

Node new\_Node = new Node (new\_data);

new\_Node.next = *head*;

new\_Node.prev = null;

if (*head* != null)*head*.prev = new\_Node;

*head* = new\_Node;

}

while (last.next != null)last = last.next;

last.next = new\_Node;

new\_Node.prev = last ;

}

public void display (Node head) {

Node temp = head;

while (temp != null) {

System.***out***.print(temp.data + " --> ");

temp = temp.next;

}

System.***out***.println("Null");

}

public static void main(String[] args) {

DoubleLinkedList3 dll = new DoubleLinkedList3();

dll.push(42);

dll.push(-3);

dll.push(17);

dll.display(*head*);

}

}

Di program ini kita membuat Double Linked List dengan program paling akhir sebagai *head*

Berikut output dari peogram tersebut

17 --> -3 --> 42 --> Null

public class DoubleLinkedList3 {

static Node *head*;

class Node {

int data;

Node prev;

Node next;

Node (int d) { data = d;}

}

public void push2 (int new\_data) {

Node new\_Node = new Node (new\_data);

Node last = *head*;

new\_Node.next = null;

if (*head* == null ) {

new\_Node.prev = null;

*head* = new\_Node;

return;

}

while (last.next != null)last = last.next;

last.next = new\_Node;

new\_Node.prev = last ;

}

public void display (Node head) {

Node temp = head;

while (temp != null) {

System.***out***.print(temp.data + " --> ");

temp = temp.next;

}

System.***out***.println("Null");

}

public static void main(String[] args) {

DoubleLinkedList3 dll = new DoubleLinkedList3();

dll.push(42);

dll.push(-3);

dll.push2(17);

dll.display(*head*);

}

Di bentuk kedua ini kita dapat melihat bahwa *head* di mulai dari -3 yaitu dari Tengah

Berikut output program tersebut:

-3 --> 42 --> 17 --> Null

1. **Kesimpulan Praktikum**

Pada A dan B menunjukkan cara berbeda untuk mengelola linked list di java. Program pertama menggunakan implementasi manual dari linked list dengan kelas **ListNode**, sementara program kedua memanfaatkan **LinkedList.** Pada program C menerapkan Double Linked List yaitu dengan menambahkan referensi ke node sebelumnya, double linked list memberikan fleksibilitas lebih dalam manipulasi data dibandingkan single linked list.