Laporan Praktikum

Struktur Data



Disusun Oleh :

**Ahda Rindang Al-Amin (2311531003)**

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

Departemen Informatika

Fakultas Teknologi Informasi

Universitas Andalas

Tahun 2024

**LinkedList**

1. **Tujuan Praktikum**
2. Memahami penggunaan Single LinkedList untuk struktur data pada bahasa pemograman Java
3. Memahami penggunaan Double LinkedList untuk struktur data pada bahasa pemograman Java
4. **Pendahuluan**

Linked List is a part of the Collection framework present in java.util package. This class is an implementation of the LinkedList data structure which is a linear data structure where the elements are not stored in contiguous locations and every element is a separate object with a data part and address part. The elements are linked using pointers and addresses. Each element is known as a node.

Due to the dynamicity and ease of insertions and deletions, they are preferred over the arrays. It also has a few disadvantages like the nodes cannot be accessed directly instead we need to start from the head and follow through the link to reach a node we wish to access.

Since a LinkedList acts as a dynamic array and we do not have to specify the size while creating it, the size of the list automatically increases when we dynamically add and remove items. And also, the elements are not stored in a continuous fashion. Therefore, there is no need to increase the size. Internally, the LinkedList is implemented using the doubly linked list data structure.

The main difference between a normal linked list and a doubly LinkedList is that a doubly linked list contains an extra pointer, typically called the previous pointer, together with the next pointer and data which are there in the singly linked list.

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

public class ListNode {

public int data; //data stored in this node

public ListNode next; //link to next node in the list

//post: constructs a node with data 0 and null link

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;

}

}

Kelas ListNode adalah representasi dari sebuah node dalah struktur data Single Linked List. Setiap simpul menyimpan sebuah nilai integer (data) dan sebuah referensi ke simpul berikutnya (next). Variabel public int data menyimpan data yang ada di simpul tersebut. Variabel public ListNode next sebagai referensi ke simpul berikutnya.

Konstruktor pertama ListNode() membuat simpul dengan data 0 dan referensi next bernilai null. this(0, null) memanggil konstruktor ketiga untuk menginisialisasi nilai tersebut. Konstruktor kedua, ListNode(int data) membuat simpul dengan nilai data yang diberikan dan referensi next bernilai null. konstruktor ketiga, ListNode (int data, ListNode next) membuat simpul dengan nilai data dan referensi next yang diberikan. Ini memungkinkan pembuatan simpul yang langsung terhubung ke simpul lain.

Berikut contoh program yang mengimplementasi 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);

}

}

Output:

42 -3 17

1. **Java.util.LinkedList**

Pada Java, terdapat library untuk struktur data LinkedList, yaitu java.util.LinkedList. Program berikut mengimplementasikan library 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***.print("After converted LinkedList to Array: ");

for (Object element : a)

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

}

}

LinkedList<Integer> list = new LinkedList<Integer>(); adalah untuk memanggil konstruktor LinkedList. Pada progam tersebut digunakan beberapa method dari LinkedList yang terdaftar pada library. Method add() adalah untuk menambahkan elemen ke dalam LinkedList. Method toArray() adalah untuk mengembalikan sebuah array dengan elemen-elemen yang sama dengan urutan yang sama dengan LinkedList.

Berikut output dari program tersebut:

LinkedList: [42, -3, 17]

After converted LinkedList to Array: 42

-3

17

1. **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;

}

public void display(Node head) {

Node temp = head;

while (temp != null) {

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

temp = temp.next;

}

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

}

}

Kelas DoubleLinkedList3 merepresentasi struktur data double linked list dimana setiap node menyimpan referensi ke simpul berikutnya dan simpul sebelumnya. Variabel static Node head menyimpan referensi ke simpul pertama (head), data menyimpan data dalam node, prev menyimpan referensi ke simpul sebelumnya dalam daftar, dan next menyimpan referensi ke simpul berikutnya.

Terdapat method push untuk menambahkan simpul baru di awal linked list dengan data yang diberikan (new\_data) dan method display untuk menampilkan semua simpul dalam linked list mulai dari simpul head.

Berikut method main yang mengimplementasikan kelas DoubleLinkedList3:

public static void main (String[] args) {

DoubleLinkedList3 dll = new DoubleLinkedList3();

dll.push(42);

dll.push(-3);

dll.push(17);

dll.display(*head*);

}

Output :

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

1. **Kesimpulan Praktikum**

LinkedList adalah salah satu struktur data yang digunakan untuk menyimpan koleksi elemen dalam urutan tertentu. LinkedList menyimpan elemen dalam node-node yang saling terhubung dengan pointer. LinkedList adalah struktur data yang fleksibel dan efisien untuk operasi penyisipan dan penghapusan elemen, terutama ketika operasi tersebut sering terjadi. Namun, untuk akses elemen yang lebih cepat, array mungkin lebih efisien. Karena pada LinkedList, untuk mengakses suatu node tidak bisa secara langsung, melainkan harus mulai dari head node dan terus hingga mencapai node yang diinginkan.