LAPORAN PRAKTIKUM

STRUKTUR DATA LINIER

MODUL IX

Dosen Pengampu

JB. Budi Darmawan S.T., M.Sc.

DISUSUN OLEH :

AGUSTINUS KEVIN YUDIPRATAMA

235314029

PROGRAM STUDI INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS SANATA DHARMA

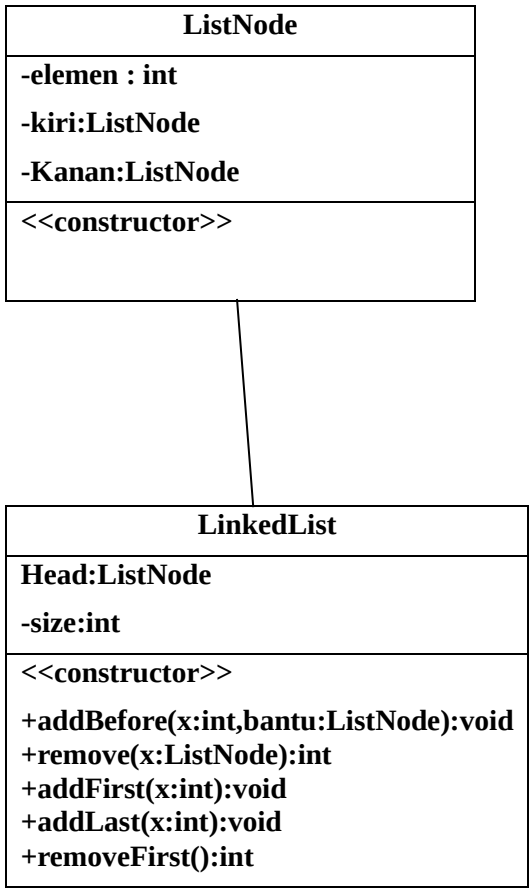
YOGYAKARTA

2024

A. TUJUAN PRAKTIKUM

- Mahasiswa mampu membuat program struktur data Queue (antrian), dengan struktur data statis (array)

B. DIAGRAM UML



C. SOURCE CODE

Class mainnya

```java
package vscode.Modul9;

import java.util.NoSuchElementException;

public class Main {
    Run | Debug
    public static void main(String[] args) {
        LinkedList senarai = new LinkedList();
        senarai.addFirst(x:8);
        senarai.addFirst(x:15);
        senarai.addLast(x:26);
        senarai.addLast(x:14);
        senarai.print();

        int cari = 26;
        ListNode hasil = senarai.search(cari);
        if (hasil != null) {
            System.out.println("Data: " + hasil.getElemen() + "ditemnukan");
        }else{
            System.out.println(x:"data tidak ditemukan ");
        }
        System.out.println();
        try{
            senarai.removeLast();
            System.out.println("Isinya: " + senarai + " ");
            senarai.removeLast();
            System.out.println("isinya: "+ senarai + " ");
            senarai.removeFirst();
            System.out.println("Isinya "+ senarai + " ");
            senarai.removeFirst();
            System.out.println("Isinya "+ senarai + " ");
            senarai.removeFirst();
            System.out.println("isinya " + senarai + " ");
        }catch(NoSuchElementException e){
            System.out.println(x:"Senarai ksosong");
        }
    }
}
```

Classnya

LindkedList

```java
package vscode.Modul9;

import java.util.NoSuchElementException;

public class LinkedList {
    ListNode head;
    int size;

    public LinkedList() {
        head = new ListNode();
        head.next = head;
        head.prev = head;
        size =0;
    }

    public ListNode getHead() {
        return head;
    }

    public void setHead(ListNode head) {
        this.head = head;
    }

    public int getSize() {
        return size;
    }

    public void setSize(int size) {
        this.size = size;
    }

    public void addBefore(int x, ListNode bantu) {
        ListNode baru = new ListNode(x);
        baru.next = bantu;
        baru.prev = bantu.prev;
        bantu.prev.next = baru;
        bantu.prev = baru;
        size++;
    }
    public int remove(ListNode bantu){
        if (isEmpty()) {
            throw new NoSuchElementException();
        }
        bantu.prev.next = bantu.next;
        bantu.next.prev = bantu.prev;
        size--;
        return bantu.elemen;
    }

    public void print() {
        ListNode bantu = head.next;
        while (bantu != head) {
            System.out.println(bantu.elemen);
            bantu = bantu.next;
        }
    }

    public void addFirst(int x) {
        addBefore(x, head.next);
    }

    public void addLast(int x) {
        addBefore(x, head);
    }

    public int removeFirst() {
        if (head.next != head) { // Check if the list is not empty
            int value = head.next.elemen;
            head.next = head.next.next;
            if (head.next != head) {
                head.next.prev = head;
            }
            size--;
            return value;
        } else {
            return -1; // Return -1 or any value that indicates an error
        }
    }
    public int removeLast(){
        return remove(head.prev);
    }
    public ListNode search(int key){
        ListNode bantu = head;
        while (bantu!= head) {
            if (bantu.elemen == key) {
                return bantu;
            }
            bantu = bantu.next;
        }
        return null;
    }
    public int size(){
        return size;
    }
    public boolean isEmpty(){
        return size == 0;
    }
    public String toString(){
        String cetak = " ";    The value of the local variable cetak is not used
        ListNode bantu = head.next;
        while (bantu != head) {
            System.out.println(bantu.elemen);
            bantu = bantu.next;
        }
        return null;
```

List Node

```java
package vscode.Modul9;

public class ListNode {
    int elemen;
    ListNode next;
    ListNode prev;

    public ListNode(){
        this(x:0);
    }
    public ListNode(int x) {
        this.elemen = x;
        this.next = null;
        this.prev = null;
    }
    public String getElemen() {
        // TODO Auto-generated method stub    TODO Auto-generated method stub
        throw new UnsupportedOperationException(message:"Unimplemented method 'getElemen'");
    }

    }
```

D. OUTPUT

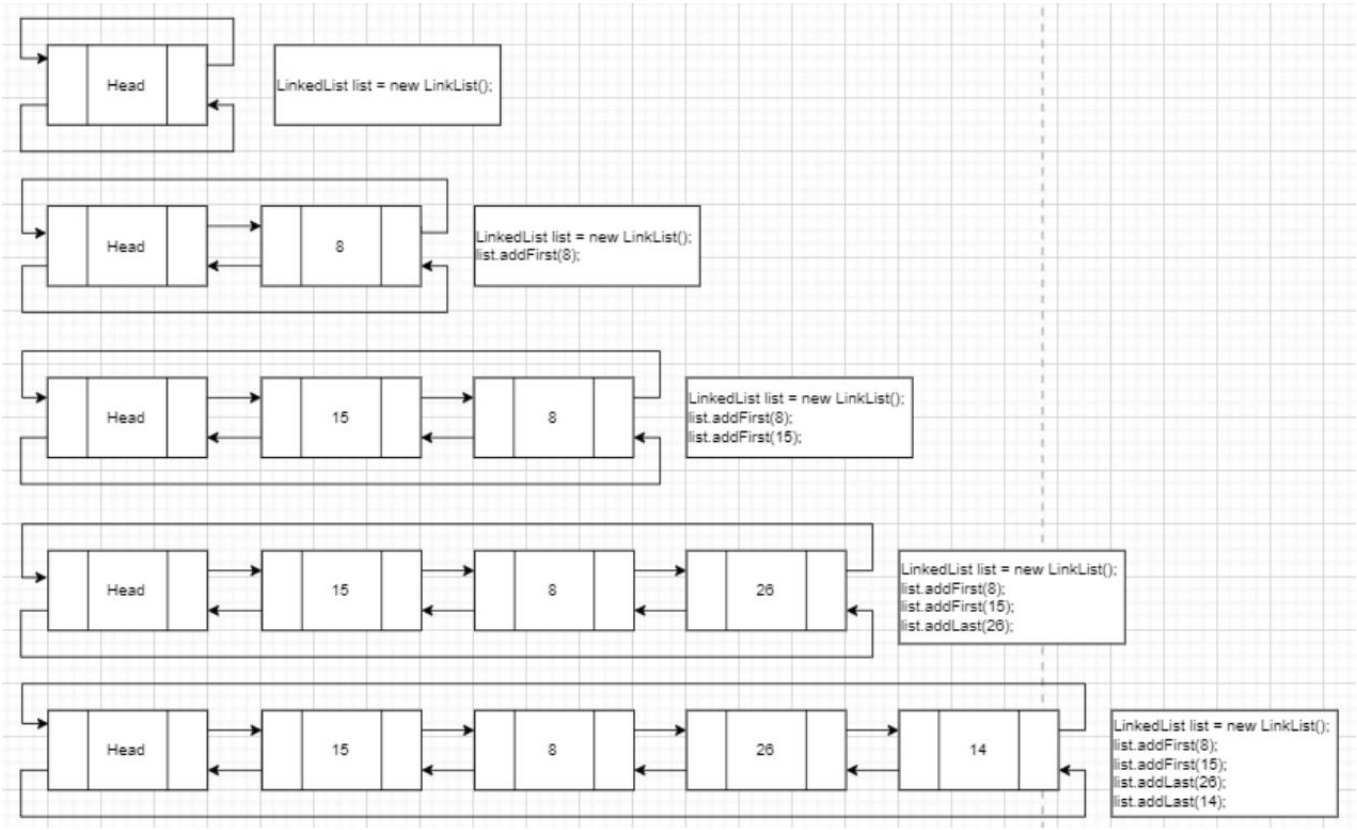
```
15
8
26
14
data tidak ditemukan

15
8
26
Isinya: null
15
8
isinya: null
8
Isinya null
Isinya null
isinya null
```

E. ANALISIS

a. LinkedList

- Contruxtor LinkedList
  Digunakan untuk menjuk ke dirinya sendiri untuk next dan prev.
- Method addBefore
  Mengatur referensi next dan prev dari simpul baru dan simpul sekitar.
- Method remove
  Mengatur ulang referensi next dan prev dari simpul sekitarnya. Jika linked list kosong, melempar pengecualian NoSuchElementException.
- Method print
  Mencetak semua elemen dalam linked list mulai dari simpul pertama hingga kembali ke kepala.
- Method addFirst
  Menyisipkan elemen baru di awal linked list dengan menggunakan addBefore
- Method addLast
  Menyisipkan elemen baru di akhir linked list dengan menggunakan addBefore
- Method removeFirst
  Menghapus dan mengembalikan elemen pertama dalam linked list dengan menggunakan remove
- Method removeLast
  Menghapus dan mengembalikan elemen terakhir dalam linked list dengan menggunakan remove
- Method search
  Mencari elemen dalam linked list. Jika ditemukan, mengembalikan referensi ke simpul tersebut, jika tidak, mengembalikan null.
- Method size
  Mengembalikan ukuran (jumlah elemen) dari linked list.
- Method isEmpty
  Memeriksa apakah linked list kosong.
- Method toString
  Mengembalikan representasi string dari linked list dengan semua elemen yang dipisahkan oleh spasi.

Ilustarsi


Head
LinkedList list = new LinkList();
Head
8
LinkedList list = new LinkList();
list.addFirst(8);
Head
15
8
LinkedList list = new LinkList();
list.addFirst(8);
list.addFirst(15);
Head
15
8
26
LinkedList list = new LinkList();
list.addFirst(8);
list.addFirst(15);
list.addLast(26);
Head
15
8
26
14
LinkedList list = new LinkList();
list.addFirst(8);
list.addFirst(15);
list.addLast(26);
list.addLast(14);