POLITEKNIK NEGERI MALANG

TEKNOLOGI INFORMASI

TEKNIK INFORMATIKA



#### Nama : Muhammad Nuril Huda

#### Kelas : TI-1A

#### No : 19

#### Mata Kuliah : Algoritma dan Struktur Data

## 12.2 Kegiatan Praktikum 1

12.2.1 Kode Program

* Kode Program Node

public class Node {

    int data;

    Node prev, next;

    Node (Node prev, int data, Node next){

        this.prev = prev;

        this.data = data;

        this.next = next;

    }

}

* Kode Program Doublelinkedlist

public class Doublelinkedlist {

    Node head;

    int size;

    Doublelinkedlist(){

        head = null;

        size = 0;

    }

    public boolean isEmpty(){

        return head == null;

    }

    public void addFirst(int item) {

        if (isEmpty()) {

            head = new Node(null, item, null);

        } else {

            Node newNode = new Node(null, item, head);

            head.prev = newNode;

            head = newNode;

        }

        size++;

    }

public void addLast(int item) {

        if (isEmpty()) {

            addFirst(item);

        } else {

            Node current = head;

            while (current.next != null) {

                current = current.next;

            }

            Node newNode = new Node(current, item, null);

            current.next = newNode;

        }

        size++;

    }

    public void add(int item, int index) throws Exception {

        if (isEmpty()) {

            addFirst(item);

        } else if (index < 0 || index > size) {

            throw new Exception("Nilai indeks di luar batas");

        } else {

            Node current = head;

            int i = 0;

            while (i < index) {

                current = current.next;

                i++;

            }

            if (current.prev == null) {

                Node newNode = new Node(null, item, current);

                current.prev = newNode;

                head = newNode;

            } else {

                Node newNode = new Node(current.prev, item, current);

                newNode.prev = current.prev;

                newNode.next = current;

                current.prev.next = newNode;

                current.prev = newNode;

            }

        }

        size++;

    }

public int size(){

        return size;

    }

    public void clear(){

        head = null;

        size = 0;

    }

    public void print() {

        if (!isEmpty()) {

            Node tmp = head;

            while (tmp != null) {

                System.out.print(tmp.data + "\t");

                tmp = tmp.next;

            }

            System.out.println("\nberhasil diisi");

        } else {

            System.out.println("Linked Lists Kosong");

        }

    }

}

* Kode Program Doublelinkedlistmain

ublic class Doublelinkedlistmain {

    public static void main(String[] args) throws Exception {

        Doublelinkedlist dll = new Doublelinkedlist();

        dll.print();

        System.out.println("Size : " + dll.size());

        System.out.println("=========================");

        dll.addFirst(3);

        dll.addLast(4);

        dll.addFirst(7);

        dll.print();

        System.out.println("Size : " + dll.size());

        System.out.println("=========================");

dll.add(40, 1);

        dll.print();

        System.out.println("Size : " + dll.size());

        System.out.println("=========================");

        dll.clear();

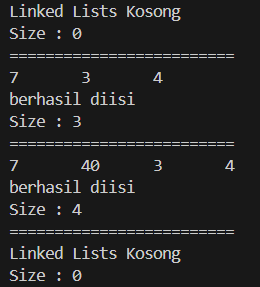
        dll.print();

        System.out.println("Size : " + dll.size());

    }

}

12.2.2 Hasil Kode Program



12.2.3 Pertanyaan

1. Jelaskan perbedaan antara single linked list dengan double linked lists!

* Single linked list hanya bisa bergerak ke depan yaitu dari head ke tail dan tidak bisa mundur ke node sebelumnya secara langsung. Sedangkan double linked list bisa bergerak ke depan dan ke belakang karena ada prev dan next.

1. Perhatikan class Node, didalamnya terdapat atribut next dan prev. Untuk apakah atribut tersebut?

* Next merupakan referensi ke node berikutnya dalam list, digunakan saat ingin maju ke node setelahnya. Sedangkan prev merupakan referensi ke node sebelumnya dalam list, digunakan saat ingin mundur ke node sebelumnya.

1. Perhatikan konstruktor pada class DoubleLinkedLists. Apa kegunaan inisialisasi atribut head dan size seperti pada gambar berikut ini?

* Atribut head diinisialisasi dengan nilai null dan size dengan nilai 0 bertujuan untuk menandai bahwa linked list masih dalam keadaan kosong saat pertama kali dibuat.

1. Pada method addFirst(), kenapa dalam pembuatan object dari konstruktor class Node prev dianggap sama dengan null?

Node newNode = new Node(null, item, head);

* Prev diisi dengan null karena node tersebut akan menjadi node pertama (head) dalam double linked list. Pada struktur double linked list, node pertama tidak memiliki node sebelumnya, sehingga referensi ke node sebelumnya (prev) harus diatur menjadi null.

1. Perhatikan pada method addFirst(). Apakah arti statement head.prev = newNode?

* Statement head.prev = newNode; digunakan untuk menghubungkan node lama (head sebelumnya) dengan node baru yang ditambahkan di depan, agar pointer prev dari node lama mengarah ke node baru.

1. Perhatikan isi method addLast(), apa arti dari pembuatan object Node dengan mengisikan parameter prev dengan current, dan next dengan null?

Node newNode = new Node(current, item, null);

* Pembuatan objek digunakan untuk menghubungkan node baru ke node terakhir sebelumnya (current) lewat prev dan Menandai bahwa node baru adalah **node terakhir** dengan next = null.

## 12.3 Kegiatan Praktikum 2

12.3.1 Kode Program

* Perubahan Kode Program Doublelinkedlist

public void removeFirst() throws Exception {

        if (isEmpty()) {

            throw new Exception("Linked List masih kosong, tidak dapat dihapus!");

        } else if (size == 1) {

            removeLast();

        } else {

            head = head.next;

            head.prev = null;

            size--;

        }

    }

    public void removeLast() throws Exception {

        if (isEmpty()) {

            throw new Exception("Linked List masih kosong, tidak dapat dihapus!");

        } else if (head.next == null) {

            head = null;

            size--;

            return;

        }

        Node current = head;

        while (current.next.next != null) {

            current = current.next;

        }

        current.next = null;

        size--;

    }

* Perubahan Kode Program Doublelinkedlistmain

dll.addLast(50);

        dll.addLast(40);

        dll.addLast(10);

        dll.addLast(20);

        dll.print();

        System.out.println("Size : " + dll.size());

        System.out.println("=========================");

public void remove(int index) throws Exception {

        if (isEmpty() || index >= size) {

            throw new Exception("Nilai indeks di luar batas");

        } else if (index == 0) {

            removeFirst();

        } else {

            Node current = head;

            int i = 0;

            while (i < index) {

                current = current.next;

                i++;

            }

            if (current.next == null) {

                current.prev.next = null;

            } else if (current.prev == null) {

                current = current.next;

                current.prev = null;

                head = current;

            } else {

                current.prev.next = current.next;

                current.next.prev = current.prev;

            }

            size--;

        }

    }

dll.removeFirst();

        dll.print();

        System.out.println("Size : " + dll.size());

        System.out.println("=========================");

        dll.removeLast();

        dll.print();

        System.out.println("Size : " + dll.size());

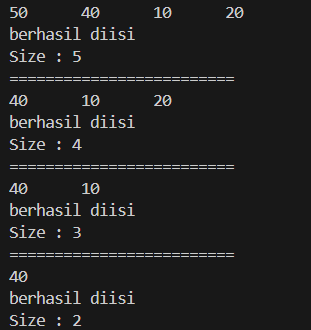
        System.out.println("=========================");

        dll.remove(1);

        dll.print();

        System.out.println("Size : " + dll.size());

12.3.2 Hasil Kode Program



12.2