Agiftsany Azhar
152011513020/D3-Sistem Informasi

```
-----Main-----
package tugas6_152011513020;

public class Tugas6_152011513020 {

    public static void main(String[] args) {
        System.out.println("-------------------------------------------------");
        System.out.println("Praktikum 6");
        System.out.println("-------------------------------------------------");
        System.out.println("-------------------------------------------------");
        System.out.println("Nomor 1 - Membuat Method untuk Membalik Node (Single
"
                + "Linked List)");
        System.out.println("-------------------------------------------------");

        List a;

        a = new List();

        a.addFront(5);
        a.addFront(10);
        a.addFront(15);
        a.addFront(20);
        a.addFront(25);
        a.print();

        a.reverse();
        a.print();

        System.out.print("\n");
        System.out.println("-------------------------------------------------");
        System.out.println("Nomor 2 - Membuat Circular Single Linked List");
        System.out.println("-------------------------------------------------");

        Circular b;

        b = new Circular();

        b.addFront(30);
        b.addFront(35);
        b.addFront(40);
        b.addFront(45);
        b.addFront(50);
        b.addRear(55);
        b.addRear(60);
        b.addRear(65);
        b.addRear(70);
        b.addRear(75);
        b.print();
    }
}
```

-----Class-----

```java
package tugas6_152011513020;

public class Node {
    int info;       // memuat satu data integer
    Node next;      // pointer to next node
    Node prev;

    /**
    * constructor dengan parameter info
    * @param info
    */
    public Node(int info){
        this.info   = info;
        this.next   = null;
        this.prev   = null;
    }

    /**
    * mengubah nilai variable info dengan nilai tertentu yang dimasukkan
    * dari luar melalui parameter input
    * @param info
    */
    public void setInfo(int info){
        this.info    = info;
    }

    /**
    * mengubah variable pointer next menunjuk ke object tertentu sesuai nilai
    * parameter input
    * @param next
    */
    public void setNext(Node next){
        this.next   = next;
    }

    public void setPrev(Node prev){
        this.prev   = prev;
    }

    /**
    * mengambil nilai info dari sebuah node
    * mengembalikan sebuah nilai integer
    * @return
    */
    public int getInfo(){
        return this.info;
    }

    /**
    * mengambil nilai pointer next, nilainya mungkin null atau merefers
    * pada address/ alamat yang merujuk pada node lain
    * mengembalikan nilai pointer of Node
    * @return
    */
    public Node getNext(){
        return this.next;
    }

    public Node getPrev(){
        return this.prev;
    }

    /**
    * menyetak nilai yang termuat di dalam info
    */
    public void print(){
        System.out.println("Info    = " + this.info);
    }
}

-----Class-----
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package tugas6_152011513020;

/**
 *
 * @author Agiftsany Azhar
 */
public class List {
    Node head; // pointer menunjuk node terdepan
    Node tail; // pointer menunjuk node terakhir
    int size; // jumlah node yang terdapat dalam list

    public List(){
        this.head=null;
        this.tail=null;
        this.size=0;
    }

    public boolean isEmpty(){
        return this.size==0;
    }

    public void addFront(int info){
        if (isEmpty()==true){
            this.head=new Node(info);
            this.tail=this.head;
            this.size++;
        }

        else{
            Node t=new Node(info);
            t.setNext(this.head);
            this.head=t;
            this.size++;
        }
    }

    public void addRear(int info){
        if (isEmpty()==true){
            this.head=new Node(info);
            this.tail=new Node(info);
            this.size++;
        }

        else{
            Node t=new Node(info);
            this.tail.setNext(t);
            this.tail=t;
            this.size++;
        }
    }

    public void delFront(){
        if (isEmpty()==true){
        }
        else{
            this.head=this.head.getNext();
            this.size--;
        }
    }

    public void delRear(){
        if (isEmpty()==true){
        }
        else{
            Node t=this.head;
            for(int i=1;i<(this.size-1);i++){
                t=t.getNext();
            }
            this.tail=t;
            this.size--;
        }
    }

    public boolean found(int info){
        Node t=this.head;
        for(int i=1;i<=this.size;i++){
            if(t.getInfo()==info){
                return true;
            }
            t=t.getNext();
        }
        return false;
    }
```

```
public Node getPosition(int info){
    Node t=this.head;
    for(int i=1;i<=this.size;i++){
        if(t.getInfo()==info){
            return t;
        }
        t=t.getNext();
    }
    return null;
}

public void delete(int info){
    Node t=this.head;
    int pos=0;
    boolean parm=false;
    do{
        if (t.getInfo()==info){
            t=t.getNext();
            Node a=this.head;
            for(int i=1;i<(pos-1);i++){
                a=a.getNext();
            }
            a.setNext(t);
            parm=true;
            this.size--;
        }
        pos++;
        t=t.getNext();
    }while (parm==false || pos==size);
}

public void reverse(){
    Node a=this.tail;
    Node b=this.head;
    this.head=a;
    int s=this.size;
    while(s!=0){
        Node c=b;
        for (int i=1;i<s-1;i++){
            c=c.getNext();
        }
        a.setNext(c);
        a=a.getNext();
        s--;
    }
    this.tail=b;
}

public void print(){
    Node t=this.head;
    for(int i=1;i<=this.size;i++){
        System.out.print(t.getInfo()+" ");
        t=t.getNext();
    }
```

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

-----Class-----
package tugas6_152011513020;

public class Circular {
    Node head; // pointer menunjuk node terdepan
    Node tail; // pointer menunjuk node terakhir
    int size;  // jumlah node yang terdapat dalam list

    public Circular(){
        this.head=null;
        this.tail=null;
        this.size=0;
    }

    public boolean isEmpty(){
        return this.size==0;
    }

    public void addFront(int info){
        if (isEmpty()==true){
            this.head=new Node(info);
            this.tail=this.head;
            this.head.setNext(this.head);
            this.size++;
        }

        else{
            Node t=new Node(info);
            t.setNext(this.head);
            this.head=t;
            this.tail.setNext(this.head);
            this.size++;
        }
    }

    public void addRear(int info){
        if (isEmpty()==true){
            this.head=new Node(info);
            this.tail=this.head;
            this.head.setNext(this.head);
            this.size++;
        }

        else{
            Node t=new Node(info);
            this.tail.setNext(t);
            this.tail=t;
            this.tail.setNext(this.head);
            this.size++;
        }
    }

    public void delFront(){
        if (isEmpty()==true){
        }
        else{
            this.head=this.head.getNext();
            this.tail.setNext(this.head);
            this.size--;
        }
    }

    public void delRear(){
        if (isEmpty()==true){
        }
        else{
            Node t=this.head;
            for(int i=1;i<(this.size-1);i++){
                t=t.getNext();
            }
            this.tail=t;
            this.tail.setNext(this.head);
            this.size--;
        }
    }

    public boolean found(int info){
        Node t=this.head;
        for(int i=1;i<=this.size;i++){
            if(t.getInfo()==info){
                return true;
            }
            t=t.getNext();
        }
        return false;
    }

    public Node getPosition(int info){
        Node t=this.head;
        for(int i=1;i<=this.size;i++){
            if(t.getInfo()==info){
                return t;
            }
            t=t.getNext();
        }
        return null;
    }

    public void delete(int info){
        if (isEmpty()==false){
            Node t=this.head;
            int pos=1;
            while (t.info!=info&&pos<this.size){
                t=t.getNext();
                pos++;
            }
            if (t.getInfo()==info){
                if(t==this.head){
                    delFront();
                }
                else if (t==this.tail){
                    delRear();
                }
                else{
                    t=t.getNext();
                    Node a=this.head;
                    for(int i=1;i<(pos-1);i++){
                        a=a.getNext();
                    }
                    a.setNext(t);
                    this.size--;
                }
            }
        }
    }

    public void print(){
        Node t=this.head;
        for(int i=1;i<=this.size+1;i++){
            System.out.print(t.getInfo()+" ");
            t=t.getNext();
        }
        System.out.println("");
    }
}
```