LAPORAN HASIL PRATIKUM

ANALISIS DAN STRUKTUR DATA

JOBSHEET 11

Raihan Akbar Putra Prasetyo/244107020087

Kelas: TI-1E

D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

PRAKTIKUM 25

Percobaan 1

Kode program
Mahasiswa24.java

```
package jobsheet11;

public class Mahasiswa24 {
    String nim,nama,kelas;
    double ipk;

     public Mahasiswa24(String nim, String nama, String kelas, double
ipk) {
        this.nim = nim;
       this.nama = nama;
        this.kelas = kelas;
        this.ipk = ipk;
    }

   public void tampilInformasi() {
    System.out.printf("%-10s %-10s %-4s %.1f\n", nama, nim, kelas, ipk);
}

}
```

NodeMahasiswa24.java

```
package jobsheet11;

import java.lang.classfile.components.ClassPrinter.Node;

public class NodeMahasiswa24 {
    Mahasiswa24 data;
    NodeMahasiswa24 next;

    public NodeMahasiswa24(Mahasiswa24 data, NodeMahasiswa24
next){
        this.data= data;
        this.next=next;
    }
}
```

SingleLinkedList24.java

```
package jobsheet11;

public class SingleLinkedList24 {
    NodeMahasiswa24 head;
    NodeMahasiswa24 tail;

    boolean isEmpty (){
        return (head== null);
    }

    public void print(){
        if (!isEmpty()) {
            NodeMahasiswa24 tmp = head;
            System.out.println("isi linked list:\t");
            while (tmp != null) {
                tmp.data.tampilInformasi();
                tmp = tmp.next;

            }
            System.out.println("");
        } else {
            System.out.println("Linked list kosong");
        }
    }

    public void addFirst (Mahasiswa24 input){
        NodeMahasiswa24 ndInput = new
NodeMahasiswa24(input,null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;

        }else {
            ndInput.next = head;
            head = ndInput;
        }
    }
    public void addLast(Mahasiswa24 input){
        NodeMahasiswa24 ndInput = new
NodeMahasiswa24(input,null);
         if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            tail.next = ndInput;
            tail = ndInput;
        }
    }
    public void insertAfter(String key, Mahasiswa24 input) {
        NodeMahasiswa24 ndInput = new NodeMahasiswa24(input,
null);
        NodeMahasiswa24 temp = head;
        do {
            if (temp.data.nama.equalsIgnoreCase(key)) {
                ndInput.next = temp.next;
                temp.next = ndInput;
                if (ndInput.next == null) {
                    tail = ndInput;
                }
                break;
            }
            temp = temp.next;
        } while (temp != null);
```

SLLMain24.java

```
package jobsheet11;

public class SLLMain24 {
    public static void main(String[] args) {
          SingleLinkedList24 sll = new SingleLinkedList24();

        Mahasiswa24 mhs3 = new Mahasiswa24("21212203", "Dirga", "4D", 3.6);
        Mahasiswa24 mhs1 = new Mahasiswa24("22212202", "Cintia", "3C",
3.5);
        Mahasiswa24 mhs2 = new Mahasiswa24("23212201", "Bimon", "2B", 3.8);
        Mahasiswa24 mhs4 = new Mahasiswa24("24212200", "Alvaro", "1A",
4.0);

        sll.print();
        sll.addFirst(mhs4);
        sll.print();
        sll.addLast(mhs1);
        sll.print();
        sll.insertAfter("Dirga", mhs3);
        sll.insertAt(2, mhs2);
        sll.print();

}
}
```

Output

```
bin   jobsheet11.SLLMain24
Linked list kosong
isi linked list:
Alvaro     24212200   1A   4,0

isi linked list:
Alvaro     24212200   1A   4,0
Cintia     22212202   3C   3,5

isi linked list:
Alvaro     24212200   1A   4,0
Cintia     22212202   3C   3,5
Bimon      23212201   2B   3,8

PS D:\kuliah\PRAKTIKUM-ASD>
```

Pertanyaan :

1. Mengapa hasil compile kode program di baris pertama menghasilkan “Linked List Kosong”?

Karena pada fungsi print terdapat pengecekan apakah linked list kosong atau tidak, pada kondisi ini linked list masih kosong maka dari itu di print linked list kosong

2. Jelaskan kegunaan variable temp secara umum pada setiap method!

Digunakan sebagai data sementara/temporary

3. Lakukan modifikasi agar data dapat ditambahkan dari keyboard

```
int lanjut = 100;

        for (int i = 0; i < lanjut ; i++) {
            System.out.println("Masukkan data mahasiswa ke- " + (i + 1)
+ " : ");
            System.out.print("NIM: ");
            String nim = sc.nextLine();
            System.out.print("Nama: ");
            String nama = sc.nextLine();
            System.out.print("Kelas: ");
            String kelas = sc.nextLine();
            System.out.print("IPK: ");
            double ipk = sc.nextDouble();
            sc.nextLine();

            Mahasiswa24 m = new Mahasiswa24(nim, nama, kelas, ipk);
            sll.addFirst(m);
            System.out.println();
            System.out.print("Apakah mau lanjut? (y/n) : ");
            String jawab = sc.nextLine();
            if (jawab.equalsIgnoreCase("n")) {
                System.out.println("Terima kasih.");
                break;
                }
```

Percobaan 2

Modifikasi Singlelinkedlist24.java

```
public void getdata (int index){
        NodeMahasiswa24 tmp = head;
        for (int i= 0; i <index; i++){
            tmp =tmp.next;
        }
        tmp.data.tampilInformasi();
    }
    public int indexOf(String key){
        NodeMahasiswa24 tmp = head;
        int index = 0;
        while (tmp != null && !tmp.data.nama.equalsIgnoreCase(key)) {
            tmp = tmp.next;
            index++;
        }

        if (tmp == null) {
            return -1;
        }else{
            return index;
        }
    }

    public void removeFirst(){
        if (isEmpty()) {
            System.out.println("Linked List masih kosong, tidak dapat dihapus!");
        }else if(head == tail){
            head = tail = null;
        }else{
            head = head.next;
        }
    }

    public void removeLast(){
        if (isEmpty()) {
            System.out.println("Linked List masih kosong, tidak dapat dihapus!");
        }else if ( head == tail){
            head = tail = null;
        }else{
            NodeMahasiswa24 temp = head;
            while (temp.next != tail) {
                temp = temp.next;
            }
            temp.next = null;
            tail = temp;
        }
    }
```

```
public void remove (String key){
        if (isEmpty()) {
            System.out.println("Linked List masih kosong, tidak dapat dihapus!");
        }else{
            NodeMahasiswa24 temp = head;
            while (temp != null) {
                if (temp.data.nama.equalsIgnoreCase(key) && temp == head) {
                    this.removeFirst();
                    break;
                }else if (temp.data.nama.equalsIgnoreCase(key)){
                    temp.next = temp.next.next;
                    if (temp.next == null) {
                        tail = temp;
                    }
                    break;
                }
                temp = temp.next;
            }
        }
    }

    public void removeAt(int index){
        if (index == 0) {
            removeFirst();
        }else{
            NodeMahasiswa24 temp = head;
            for (int i = 0; i < index - 1; i++) {
                temp = temp.next;
            }
            temp.next = temp.next.next;
            if (temp.next == null) {
                tail = temp;
            }
        }
    }
}
```

Modifikasi main

```
Mahasiswa24 mhs1 = new Mahasiswa24("21212203", "Dirga", "4D", 3.6);
        Mahasiswa24 mhs3 = new Mahasiswa24("22212202", "Cintia", "3C", 3.5);
        Mahasiswa24 mhs2 = new Mahasiswa24("23212201", "Bimon", "2B", 3.8);
        Mahasiswa24 mhs4 = new Mahasiswa24("24212200", "Alvaro", "1A", 4.0);
        sll.addFirst(mhs1);
        sll.addFirst(mhs2);
        sll.addFirst(mhs3);
        sll.addFirst(mhs4);
        System.out.println("data index 1 :");
        sll.getdata(1);
        System.out.println();

        System.out.println("Data mahasiswa an Bimon berada pada index : " +
sll.indexOf("bimon"));
        System.out.println();

        sll.removeFirst();
        sll.removeLast();
        sll.print();
        sll.removeAt(0);
        sll.print();
```

Output :

```
data index 1 :
Cintia     22212202   3C   3,5

Data mahasiswa an Bimon berada pada index : 2

isi linked list:
Cintia     22212202   3C   3,5
Bimon      23212201   2B   3,8

isi linked list:
Bimon      23212201   2B   3,8

PS D:\kuliah\PRAKTIKUM-ASD>
```

Tugas :

MahasiswaTugas.Java

```
package jobsheet11;

public class MahasiswaTugas {
      String nama, nim, kelas, prodi;

     public MahasiswaTugas(String nim, String nama, String kelas, String prodi) {
        this.nim = nim;
        this.nama = nama;
        this.kelas = kelas;
        this.prodi = prodi;
    }

     public void tampilInformasi() {
        System.out.printf("Nama: %s\n", nama);
        System.out.printf("NIM: %s\n", nim);
        System.out.printf("Prodi: %s\n", prodi);
        System.out.printf("Kelas: %s\n", kelas);
        System.out.println();
    }


}
```

NodeMahasiswaTugas.java

```
package jobsheet11;

import java.lang.classfile.components.ClassPrinter.Node;

public class NodeMahasiswa24 {
    Mahasiswa24 data;
    NodeMahasiswa24 next;

    public NodeMahasiswa24(Mahasiswa24 data, NodeMahasiswa24 next){
        this.data= data;
        this.next=next;
    }
}
```

LinkedlistLayananUnit.java

```
package jobsheet11;

public class LinkedListLayananUnit {
     NodeMhsTugas head;
    NodeMhsTugas tail;

    int maxSize = 100;
    int size = 0;

    boolean isEmpty(){
        return (head == null);
    }
    public void kosongkan() {
        head = tail = null;
        System.out.println("Antrian telah dikosongkan.");
    }

    public boolean cekAntrianPenuh() {
        return size >= maxSize;
    }
    public void mengosongkanAntrian() {
        head = tail = null;
        System.out.println("Antrian Dikosongkan");
    }

    public void daftarkanData(MahasiswaTugas input){
        NodeMhsTugas ndInput = new NodeMhsTugas(input, null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        }else{
            tail.next = ndInput;
            tail = ndInput;
        }
        System.out.println(ndInput.data.nama + " Berhasil masuk ke antrian" );
    }

    public void memanggilAntrian(){
        if (isEmpty()) {
            System.out.println("Antrian Kosong tidak ada yang bisa dipanggil");
        }else if(head == tail){
            System.out.println(head.data.nama + " Dipanggil");
            head = tail = null;
        }else{
            System.out.println(head.data.nama + " Dipanggil");
            head = head.next;
        }
    }



    public void tampilAntrianDepanTerakhir(){
        if (isEmpty()) {
            System.out.println("Antrian Kosong tidak ada yang bisa ditampilkan");
        }else{
            System.out.println("Antrian terdepan: ");
            head.data.tampilInformasi();
            System.out.println("Antrian Terakhir: ");
            tail.data.tampilInformasi();
        }
    }
```

```
public void tampilMahasiswaYangMengantri() {
    int count = 0;
    NodeMhsTugas temp = head;

    while (temp != null) {
        count++;
        temp = temp.next;
    }

    System.out.println("Jumlah data dalam linked list: " + count);
    }
}
```

LayananUnitKemahasiswaan24

```
package jobsheet11;

import java.util.Scanner;

public class LayananUnitKemahasiswaan24 {
    public static void main(String[] args) {
         Scanner sc = new Scanner(System.in);
        LinkedListLayananUnit antrian = new LinkedListLayananUnit();
        int pilihan;

        do{
            System.out.println("\n=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN");
            System.out.println("1. Cek Antrian Kosong");
            System.out.println("2. Cek Antrian Penuh");
            System.out.println("3. Kosongkan Antrian");
            System.out.println("4. Tambah Antrian");
            System.out.println("5. Panggil Antrian");
            System.out.println("6. Lihat Antrian Terdepan Dan Paling Akhir");
            System.out.println("7. Lihat Jumlah Antrian Mahasiwa");
            System.out.println("0. Keluar");

            System.out.print("Pilih menu: ");
            pilihan = sc.nextInt();

            switch (pilihan) {
                case 1:
                    System.out.println(antrian.isEmpty() ? "Antrian Kosong." :
"Antrian Tidak kosong.");
                    break;
                    case 2:
                    System.out.println(antrian.cekAntrianPenuh()? "Antrian Penuh." :
"Antrian Belum Penuh");
                    break;
                    case 3:
                    antrian.kosongkan();
                    break;
                    case 4:
                {
                    sc.nextLine();
                    System.out.print("Nama: ");
                    String nama = sc.nextLine();
                    System.out.print("NIM: ");
                    String nim = sc.nextLine();
                    System.out.print("Prodi: ");
                    String prodi = sc.nextLine();
                    System.out.print("Kelas: ");
                    String kelas = sc.nextLine();
                    MahasiswaTugas mhs = new MahasiswaTugas(nim, nama, prodi, kelas);
                    antrian.daftarkanData(mhs);
                    break;
                }
                case 5:
                    antrian.memanggilAntrian();
                    break;
```

```
            case 6:
                   antrian.tampilAntrianDepanTerakhir();
                   break;
               case 7:
                   antrian.tampilMahasiswaYangMengantri();
                   break;
               case 0 :
                   System.out.println("Terima kasih");
                   break;
               default:
               System.out.println("Pilihan tidak valid");
            }
        }while(pilihan != 0);
    }
    }
```