

LAPORAN TUGAS ALGORITMA DAN STRUKTUR DATA
JOBSHEET 11



NAMA : RADITYA RIEFKI
KELAS : TI 1E
ABSEN : 23

2. Praktikum

2.1 Pembuatan Single Linked List

Kode Program Mahasiswa

```
package jobsheet11;

public class Mahasiswa23 {
    String nama, nim, kelas;
    double ipk;

    Mahasiswa23(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("\n%-10s %-10s %-3s %.1f", nama, nim, kelas, ipk);
    }
}
```

Kode Program NodeMahasiswa23

```
package jobsheet11;

import org.w3c.dom.Node;

public class NodeMahasiswa23 {
    Mahasiswa23 data;
    NodeMahasiswa23 next;

    public NodeMahasiswa23(Mahasiswa23 data, NodeMahasiswa23 next){
        this.data = data;
        this.next = next;
    }
}
```

Kode Program SingleLinkedList23

```
package jobsheet11;

import org.w3c.dom.Node;

public class SingleLinkedList23 {
    NodeMahasiswa23 head;
    NodeMahasiswa23 tail;

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

    public void print(){
        if (!isEmpty()) {
            NodeMahasiswa23 tmp = head;
            System.out.print("Isi linked list: \t");
            while (tmp != null) {
                tmp.data.tampilInformasi();
                tmp = tmp.next;
            }
            System.out.println("");
        }else{
            System.out.println("Linked list kosong");
        }
        System.out.println();
    }

    public void addFirst(Mahasiswa23 input){
        NodeMahasiswa23 ndInput = new NodeMahasiswa23(input, null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        }else{
            ndInput.next = head;
            head = ndInput;
        }
    }

    public void addLast(Mahasiswa23 input){
        NodeMahasiswa23 ndInput = new NodeMahasiswa23(input, null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        }else{
            tail.next = ndInput;
            tail = ndInput;
        }
    }

    public void insertAfter(String key, Mahasiswa23 input){
        NodeMahasiswa23 ndInput = new NodeMahasiswa23(input, null);
        NodeMahasiswa23 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);
    }

    public void insertAt(int index, Mahasiswa23 input){
        if (index < 0) {
            System.out.println("Index Salah");
        }else if (index == 0){
            addFirst(input);
        }else{
            NodeMahasiswa23 temp = head;
            for (int i = 0; i < index -1 ; i++){
                temp = temp.next;
            }
            temp.next = new NodeMahasiswa23(input, temp.next);
            if (temp.next.next == null) {
                tail = temp.next;
            }
        }
    }
}

```

Kode Program SLLMain

```

package jobsheet11;

import org.w3c.dom.Node;

public class SLLMain23 {
    public static void main(String[] args) {
        SingleLinkedList23 sll = new SingleLinkedList23();
        Mahasiswa23 mhs4 = new Mahasiswa23("21212203", "Dirga", "$D", 3.6);
        Mahasiswa23 mhs2 = new Mahasiswa23("22212202", "Cintia", "TI 1E",
3.5);
        Mahasiswa23 mhs3 = new Mahasiswa23("23212201", "Bimon", "TI 1E",
3.8);
        Mahasiswa23 mhs1 = new Mahasiswa23("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();  
}  
}
```

2.1.1 Verifikasi Hasil Percobaan

```
Linked list kosong  
  
Isi linked list:  
Dirga      21212203  $D  3.6  
  
Isi linked list:  
Dirga      21212203  $D  3.6  
Alvaro     24212200  1A  4.0  
  
Isi linked list:  
Dirga      21212203  $D  3.6  
Bimon      23212201  TI 1E 3.8  
Cintia     22212202  TI 1E 3.5  
Alvaro     24212200  1A  4.0
```

2.1.2 Pertanyaan

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

- karena pada pemanggilan method pertama belum menambahkan data apapun

```
sll.print();  
sll.addFirst(mhs4);
```

Data ditambahkan setelah print

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

- temp digunakan sebagai data sementara dalam linkedlist memungkinkan melakukan operasi tanpa mengubah struktur utama

3. Lakukan modifikasi agar data dapat ditambahkan dari keyboard!

Perubahan pada kode main

```
for (int i = 0; i < lanjut ; i++) {  
    System.out.println("Masukkan data mahasiswa ke- " + (i + 1) + " : ");  
    System.out.print(s:"NIM: ");  
    String nim = sc.nextLine();  
    System.out.print(s:"Nama: ");  
    String nama = sc.nextLine();  
    System.out.print(s:"Kelas: ");  
    String kelas = sc.nextLine();  
    System.out.print(s:"IPK: ");  
    double ipk = sc.nextDouble();  
    sc.nextLine();  
  
    Mahasiswa23 m = new Mahasiswa23(nim, nama, kelas, ipk);  
    sll.addFirst(m);  
    System.out.println();  
    System.out.print(s:"Apakah mau lanjut? (y/n) : ");  
    String jawab = sc.nextLine();  
    if (jawab.equalsIgnoreCase(anotherString:"n")) {  
        System.out.println(x:"Terima kasih.");  
        break;  
    }  
}
```

2.2 Modifikasi Elemen pada Single Linked List

Kode program SingleLinkedList

```
package jobsheet11;

import org.w3c.dom.Node;

public class SingleLinkedList23 {
    NodeMahasiswa23 head;
    NodeMahasiswa23 tail;

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

    public void print(){
        if (!isEmpty()) {
            NodeMahasiswa23 tmp = head;
            System.out.print("Isi linked list: \t");
            while (tmp != null) {
                tmp.data.tampilInformasi();
                tmp = tmp.next;
            }
            System.out.println("");
        }else{
            System.out.println("Linked list kosong");
        }
        System.out.println();
    }

    public void addFirst(Mahasiswa23 input){
        NodeMahasiswa23 ndInput = new NodeMahasiswa23(input, null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        }else{
            ndInput.next = head;
            head = ndInput;
        }
    }

    public void addLast(Mahasiswa23 input){
        NodeMahasiswa23 ndInput = new NodeMahasiswa23(input, null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        }else{
            tail.next = ndInput;
            tail = ndInput;
        }
    }

    public void insertAfter(String key, Mahasiswa23 input){
```

```

NodeMahasiswa23 ndInput = new NodeMahasiswa23(input, null);
NodeMahasiswa23 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);
}

public void insertAt(int index, Mahasiswa23 input){
    if (index < 0) {
        System.out.println("Index Salah");
    }else if (index == 0){
        addFirst(input);
    }else{
        NodeMahasiswa23 temp = head;
        for (int i = 0; i < index -1 ; i++){
            temp = temp.next;
        }
        temp.next = new NodeMahasiswa23(input, temp.next);
        if (temp.next.next == null) {
            tail = temp.next;
        }
    }
}

public void getData(int index){
    NodeMahasiswa23 tmp = head;
    for (int i = 0; i < index; i++) {
        tmp = tmp.next;
    }
    tmp.data.tampilInformasi();
}

public int indexOf(String key){
    NodeMahasiswa23 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{
            NodeMahasiswa23 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{
            NodeMahasiswa23 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{

```

```

        NodeMahasiswa23 temp = head;
        for (int i = 0; i < index - 1; i++) {
            temp = temp.next;
        }
        temp.next = temp.next.next;
        if (temp.next == null) {
            tail = temp;
        }
    }
}
}

```

Kode Program Main

```

package jobsheet11;

import java.util.Scanner;

import org.w3c.dom.Node;

public class SLLMain23 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        SingleLinkedList23 sll = new SingleLinkedList23();

        Mahasiswa23 mhs1 = new Mahasiswa23("21212203", "Dirga", "4D", 3.6);
        Mahasiswa23 mhs3 = new Mahasiswa23("22212202", "Cintia", "3C", 3.5);
        Mahasiswa23 mhs2 = new Mahasiswa23("23212201", "Bimon", "2B", 3.8);
        Mahasiswa23 mhs4 = new Mahasiswa23("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();

    }
}

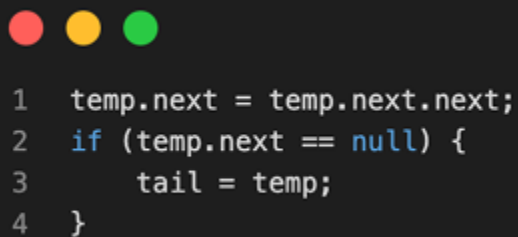
```

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:\CollegeFile\SMT 2\ALSD>
```

2.2.3 Pertanyaan

1. Mengapa digunakan keyword break pada fungsi remove? Jelaskan!
- Mencegah penghapusan lebih dari satu node setelah key ditemukan
2. Jelaskan kegunaan kode dibawah pada method remove



```
1 temp.next = temp.next.next;  
2 if (temp.next == null) {  
3     tail = temp;  
4 }
```

Pada baris pertama menghapus node dan baris kedua mejadikan tail tetap ad ajika node terakhir dihapus

TUGAS

Kode program Mahasiswa

```
package jobsheet11.TUGAS;

public class Mahasiswa {

    String nama, nim, kelas, prodi;

    public Mahasiswa(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();
    }

}
```

Kode Program LinkedListLayanan

```
package jobsheet11.TUGAS;

import jobsheet11.NodeMahasiswa23;

public class LinkedListLayananUnit {
    NodeMhs head;
    NodeMhs 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(Mahasiswa input){
    NodeMhs ndInput = new NodeMhs(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 tampilMahasiswaYangMengantre() {
    int count = 0;
    NodeMhs temp = head;

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

```

```

        temp = temp.next;
    }

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

```

Kode Progam LayananUnitMahasiswa

```

package jobsheet11.TUGAS;

import java.util.Scanner;

public class LayananUnitKemahasiswaan {
    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();
        Mahasiswa mhs = new Mahasiswa(nim, nama, prodi, kelas);
        antrian.daftarkanData(mhs);
        break;
    }
    case 5:
        antrian.memanggilAntrian();
        break;
    case 6:
        antrian.tampilAntrianDepanTerakhir();
        break;
    case 7:
        antrian.tampilMahasiswaYangMengantre();
        break;
    case 0 :
        System.out.println("Terima kasih");
        break;
    default:
        System.out.println("Pilihan tidak valid");
    }
}while(pilihan != 0);
}
}

```

OUTPUT

=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN

1. Cek Antrian Kosong
2. Cek Antrian Penuh
3. Kosongkan Antrian
4. Tambah Antrian
5. Panggil Antrian
6. Lihat Antrian Terdepan Dan Paling Akhir
7. Lihat Jumlah Antrian Mahasiwa
0. Keluar

Pilih menu: 4

Nama: Dirga

NIM: 21212203

Prodi: TI

Kelas: 4D

Dirga Berhasil masuk ke antrian

=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN

1. Cek Antrian Kosong
2. Cek Antrian Penuh
3. Kosongkan Antrian
4. Tambah Antrian
5. Panggil Antrian
6. Lihat Antrian Terdepan Dan Paling Akhir
7. Lihat Jumlah Antrian Mahasiwa
0. Keluar

Pilih menu: 4

Nama: Cintia

NIM: 22212202

Prodi: SIB

Kelas: 3C

Cintia Berhasil masuk ke antrian

=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN

1. Cek Antrian Kosong
2. Cek Antrian Penuh
3. Kosongkan Antrian
4. Tambah Antrian
5. Panggil Antrian
6. Lihat Antrian Terdepan Dan Paling Akhir
7. Lihat Jumlah Antrian Mahasiwa
0. Keluar

Pilih menu: 4

Nama: Bimon

NIM: 23212201

Prodi: SIB

Kelas: 2B

Bimon Berhasil masuk ke antrian

=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN

1. Cek Antrian Kosong
2. Cek Antrian Penuh
3. Kosongkan Antrian
4. Tambah Antrian
5. Panggil Antrian
6. Lihat Antrian Terdepan Dan Paling Akhir
7. Lihat Jumlah Antrian Mahasiwa
0. Keluar

Pilih menu: 4

Nama: Alvaro

NIM: 24212200

Prodi: TI

Kelas: 1A

Alvaro Berhasil masuk ke antrian

=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN

1. Cek Antrian Kosong
2. Cek Antrian Penuh
3. Kosongkan Antrian
4. Tambah Antrian
5. Panggil Antrian
6. Lihat Antrian Terdepan Dan Paling Akhir
7. Lihat Jumlah Antrian Mahasiwa
0. Keluar

Pilih menu: 7

Jumlah data dalam linked list: 4

=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN

1. Cek Antrian Kosong
2. Cek Antrian Penuh
3. Kosongkan Antrian
4. Tambah Antrian
5. Panggil Antrian
6. Lihat Antrian Terdepan Dan Paling Akhir
7. Lihat Jumlah Antrian Mahasiwa
0. Keluar

Pilih menu: 6

Antrian terdepan:

Nama: Dirga

NIM: 21212203

Prodi: 4D

Kelas: TI

Antrian Terakhir:

Nama: Alvaro

NIM: 24212200

Prodi: 1A

Kelas: TI

=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN

1. Cek Antrian Kosong
2. Cek Antrian Penuh
3. Kosongkan Antrian
4. Tambah Antrian
5. Panggil Antrian
6. Lihat Antrian Terdepan Dan Paling Akhir
7. Lihat Jumlah Antrian Mahasiwa
0. Keluar

Pilih menu: 5

Antrian Kosong tidak ada yang bisa dipanggil

=== MENU ANTRIAN LAYANAN UNIT KEMAHASISWAAN

1. Cek Antrian Kosong
2. Cek Antrian Penuh
3. Kosongkan Antrian
4. Tambah Antrian
5. Panggil Antrian
6. Lihat Antrian Terdepan Dan Paling Akhir
7. Lihat Jumlah Antrian Mahasiwa
0. Keluar

Pilih menu: