Laporan Hasil Praktikum Algoritma Struktur Data

Jobsheet 11

Febryan Akhmad Taajuddin

244107020180

Kelas 1E

Program Studi Teknik Informatika

Jurusan Teknologi Informasi

Politeknik Negeri Malang

2025

Percobaan 1

1. Class Mahasiswa10

```
public class Mahasiswa10 {
    String nim;
    String nama;
    String kelas;
    double ipk;

    Mahasiswa10(){

    }
    Mahasiswa10(String nm, String name, String kls, double ip){
        nim = nm;
        nama = name;
        kelas = kls;
        ipk = ip;
    }
    void tampilInformasi(){
        System.out.println(nama + "\t"+ nim+ "\t"+kelas+"\t"+ ipk);

    }
}
```

2. Class NodeMahasiswa10

```
public class NodeMahasiswa10 {
    Mahasiswa10 data;
    NodeMahasiswa10 next;

    public NodeMahasiswa10(Mahasiswa10 data, NodeMahasiswa10 next){
        this.data = data;
        this.next = next;
    }

}
```

3. Class SingleLinkedList

```
public class SingleLinkedList10 {
    NodeMahasiswa10 head;
    NodeMahasiswa10 tail;

    boolean isEmpty(){
        return (head == null);
    }
    public void print(){
        if (!isEmpty()) {
            NodeMahasiswa10 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(Mahasiswa10 input){
        NodeMahasiswa10 ndInput = new NodeMahasiswa10(input, null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        }else{
            ndInput.next = head;
            head = ndInput;
        }
    }
    public void addLast(Mahasiswa10 input){
        NodeMahasiswa10 ndInput = new NodeMahasiswa10(input, null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        }else{
            tail.next = ndInput;
            tail = ndInput;
        }
    }
    public void insertAfter(String key, Mahasiswa10 input){
        NodeMahasiswa10 ndInput = new NodeMahasiswa10(input, null);
        NodeMahasiswa10 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, Mahasiswa10 input){
        if (index < 0) {
            System.out.println("indeks salah");
        }else if(index == 0){
            addFirst(input);
        }else{
            NodeMahasiswa10 temp = head;
            for (int i = 0; i < index; i++) {
                temp = temp.next;
            }
            temp.next = new NodeMahasiswa10(input, temp.next);
            if (temp.next.next == null) {
                tail = temp.next;
            }
        }
    }
}
```

4. Class SLLMain

```
public class SLLMain10 {
    public static void main(String[] args) {
        SingleLinkedList10 sll = new SingleLinkedList10();

        Mahasiswa10 mhs1 = new Mahasiswa10("24212200","Alvaro","1A",4.0);
        Mahasiswa10 mhs2 = new Mahasiswa10("23212201","Bimon","2B",3.8);
        Mahasiswa10 mhs3 = new Mahasiswa10("22212202","Cintia","3C",3.5);
        Mahasiswa10 mhs4 = new Mahasiswa10("21212203","Dirga","4D",3.6);

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

5. Run kode progam

```
Linked list kosong
Isi Linked List:
Dirga   21212203       4D      3.6

Isi Linked List:
Dirga   21212203       4D      3.6
Alvaro  24212200       1A      4.0

Isi Linked List:
Dirga   21212203       4D      3.6
Cintia  22212202       3C      3.5
Alvaro  24212200       1A      4.0
Bimon   23212201       2B      3.8
```

Pertanyaan percobaan 1

1. Mengapa hasil compile kode program di baris pertama menghasilkan “Linked List Kosong”?
2. Jelaskan kegunaan variable temp secara umum pada setiap method!
3. Lakukan modifikasi agar data dapat ditambahkan dari keyboard!

Jawaban percobaan 1

1. Linked list kosong pada baris pertama karena belum ada data atau node yang dimasukkan, hanya method saja, jadi outputnya akan kosong
2. Variabel temp sebagai penunjuk sementara jika kita mau menambah data, menghapus, dan update

3. Modifikasi kode program

```
import java.util.Scanner;
public class SLLMain10 {
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        SingleLinkedList10 sll = new SingleLinkedList10();

        System.out.print("Masukkan jumlah data: ");
        int input = sc.nextInt();
        sc.nextLine();

        for (int i = 0; i < input; i++) {
             System.out.println("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();

            Mahasiswa10 mhs = new Mahasiswa10(nim, nama, kelas, ipk);
            sll.addLast(mhs);
            System.out.println("Setelah penambahan:");
            sll.print();
        }
        sc.close();
    }
}
```

- Output

```
Masukkan jumlah data: 5
Data Mahasiswa ke-1
NIM: 1
Nama: adi
Kelas: 1e
IPK: 4
Setelah penambahan:
Isi Linked List:
adi     1       1e      4.0

Data Mahasiswa ke-2
NIM: 2
Nama: fadhil
Kelas: 1e
IPK: 4
Setelah penambahan:
Isi Linked List:
adi     1       1e      4.0
fadhil  2       1e      4.0

Data Mahasiswa ke-3
NIM: 3
Nama: abil
Kelas: 1e
IPK: 4
Setelah penambahan:
Isi Linked List:
adi     1       1e      4.0
fadhil  2       1e      4.0
abil    3       1e      4.0
```

```
Data Mahasiswa ke-4
NIM: 4
Nama: nawaf
Kelas: 1e
IPK: 4
Setelah penambahan:
Isi Linked List:
adi     1       1e      4.0
fadhil  2       1e      4.0
abil    3       1e      4.0
nawaf   4       1e      4.0

Data Mahasiswa ke-5
NIM: 5
Nama: adid
Kelas: 1e
IPK: 4
Setelah penambahan:
Isi Linked List:
adi     1       1e      4.0
fadhil  2       1e      4.0
abil    3       1e      4.0
nawaf   4       1e      4.0
adid    5       1e      4.0
```

Modifikasi Elemen pada Single Linked List

1. Class SingleLinkedList10

```
    public void getData (int index){
        NodeMahasiswa10 tmp = head;
        for (int i = 0; i < index; i++) {
            tmp = tmp.next;
        }
        tmp.data.tampilInformasi();
    }
    public int indexOf (String key){
        NodeMahasiswa10 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{
            NodeMahasiswa10 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{
            NodeMahasiswa10 temp = head;
            while (temp != null) {
                if ((temp.data.nama.equalsIgnoreCase(key)) && (temp == head))
{
                    this.removeFirst();
                    break;
                }else if (temp.data.nama.equalsIgnoreCase(key)){
                    if (temp.next == null) {
                        tail = temp;
                    }
                    break;
                }
                temp = temp.next;
            }
        }
    }
    public void removeAt (int index){
        if (index == 0) {
            removeFirst();
        }else{
            NodeMahasiswa10 temp = head;
            for (int i = 0; i < index -1; i++) {
                temp = temp.next;
            }
            temp.next = temp.next.next;
            if (temp.next == null) {
                tail = temp;
            }
        }
    }
}
```

2. Class SLLMain10

```
public class SLLMain10 {
    public static void main(String[] args) {
        SingleLinkedList10 sll = new SingleLinkedList10();

        Mahasiswa10 mhs1 = new Mahasiswa10("22212202", "Cintia", "3C", 3.5);
        Mahasiswa10 mhs2 = new Mahasiswa10("23212201", "Bimon", "2B", 3.8);

        sll.addLast(mhs1);
        sll.addLast(mhs2);

        System.out.println("data index 1 :");
        sll.getData(0);

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

        sll.print();

        sll.removeFirst();
        sll.print();
    }
}
```

3. Run kode program

```
data index 1 :
Cintia  22212202        3C      3.5
data mahasiswa an Bimon berada pada index : 1

Isi Linked List:
Cintia  22212202        3C      3.5
Bimon   23212201        2B      3.8

Isi Linked List:
Bimon   23212201        2B      3.8
```

Pertanyaan

1. Mengapa digunakan keyword break pada fungsi remove? Jelaskan!
2. Jelaskan kegunaan kode dibawah pada method remove

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

Jawaban

1. Break digunakan untuk menghentikan perulangan while setelah data yang dicari key telah ditemukan dan dihapus
2. temp.next = tem.next.next untuk menghapus temp.next dengan cara melompati dua setelahnya, lalu jika sudah dihapus temp.next menjadi null maka tail akan berpindah ke temp

Latihan Praktikum

1. Berikut kode programnya

a. Mahasiswa10

```
package Latihan;

public class Mahasiswa10 {
    String nim, nama;

    Mahasiswa10(String nim, String nama) {
        this.nim = nim;
        this.nama = nama;
    }

    public String tampil() {
        return "NIM: " + nim + ", Nama: " + nama;
    }
}
```

b.

```
package Latihan;

public class NodeMahasiswa10 {
    Mahasiswa10 data;
    NodeMahasiswa10 next;

    NodeMahasiswa10(Mahasiswa10 data) {
        this.data = data;
        this.next = null;
    }
}
```

c. Class QueueMahasiswa10

```
package Latihan;

public class QeueuMahasiswa10 {
    NodeMahasiswa10 front, rear;
    int jumlah;

    public QeueuMahasiswa10() {
        front = null;
        rear = null;
        jumlah = 0;
    }

    public boolean isEmpty() {
        return front == null;
    }
```

```
    public void tambahAntrian(Mahasiswa10 mhs) {
        NodeMahasiswa10 nodeBaru = new NodeMahasiswa10(mhs);
        if (isEmpty()) {
            front = nodeBaru;
            rear = nodeBaru;
        } else {
            rear.next = nodeBaru;
            rear = nodeBaru;
        }
        jumlah++;
        System.out.println("Mahasiswa " + mhs.nama + " berhasil
ditambahkan ke antrian");
    }

    public void panggilAntrian() {
        if (isEmpty()) {
            System.out.println("Antrian kosong. Tidak ada yang
dipanggil");
        } else {
            System.out.println("Memanggil: " + front.data.nim + " - "
+ front.data.nama);
            front = front.next;
            jumlah--;
            if (front == null) {
                rear = null;
            }
        }
    }

    public void tampilkanDepan() {
        if (isEmpty()) {
            System.out.println("Antrian kosong");
        } else {
            System.out.println("Antrian terdepan: " + front.data.nim +
" - " + front.data.nama);
        }
    }

    public void tampilkanBelakang() {
        if (isEmpty()) {
            System.out.println("Antrian kosong");
        } else {
            System.out.println("Antrian terakhir: " + rear.data.nim +
" - " + rear.data.nama);
        }
    }

    public void tampilkanSemua() {
        if (isEmpty()) {
            System.out.println("Antrian kosong");
        } else {
            System.out.println("Daftar antrian:");
            NodeMahasiswa10 bantu = front;
            while (bantu != null) {
                System.out.println(bantu.data.nim + " - " +
bantu.data.nama);
                bantu = bantu.next;
            }
        }
    }
```

```
    public void kosongkanAntrian() {
        front = null;
        rear = null;
        jumlah = 0;
        System.out.println("Antrian telah dikosongkan");
    }

    public void tampilkanJumlah() {
        System.out.println("Jumlah mahasiswa dalam antrian: " +
jumlah);
    }

    public void cekKosong() {
        if (isEmpty()) {
            System.out.println("Antrian kosong");
        } else {
            System.out.println("Antrian tidak kosong");
        }
    }
}
```

d. MainQueue10

```
package Latihan;
import java.util.Scanner;
public class MainQueue10 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        QeueuMahasiswa10 antrian = new QeueuMahasiswa10();
        int pilih = -1;

        while (pilih != 0) {
            System.out.println();
            System.out.println("=== Menu Antrian Layanan Kemahasiswaan
===");
            System.out.println("1. Daftar Antrian");
            System.out.println("2. Panggil Antrian");
            System.out.println("3. Tampilkan Antrian Terdepan");
            System.out.println("4. Tampilkan Antrian Terakhir");
            System.out.println("5. Tampilkan Semua Antrian");
            System.out.println("6. Tampilkan Jumlah Antrian");
            System.out.println("7. Cek Antrian Kosong");
            System.out.println("8. Kosongkan Antrian");
            System.out.println("0. Keluar");
            System.out.print("Pilih: ");
            pilih = sc.nextInt();
            sc.nextLine();

            switch (pilih) {
                case 1:
                    System.out.print("Masukkan NIM: ");
                    String nim = sc.nextLine();
                    System.out.print("Masukkan Nama: ");
                    String nama = sc.nextLine();
                    Mahasiswa10 mhs = new Mahasiswa10(nim, nama);
                    antrian.tambahAntrian(mhs);
                    break;
```

```
                case 2:
                    antrian.panggilAntrian();
                    break;
                case 3:
                    antrian.tampilkanDepan();
                    break;
                case 4:
                    antrian.tampilkanBelakang();
                    break;
                case 5:
                    antrian.tampilkanSemua();
                    break;
                case 6:
                    antrian.tampilkanJumlah();
                    break;
                case 7:
                    antrian.cekKosong();
                    break;
                case 8:
                    antrian.kosongkanAntrian();
                    break;
                case 0:
                    System.out.println("Program selesai");
                    break;
                default:
                    System.out.println("Pilihan tidak valid");
            }
        }
        sc.close();
    }
```

- Output

```
=== Menu Antrian Layanan Kemahasiswaan ===
1. Daftar Antrian
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Cek Antrian Kosong
8. Kosongkan Antrian
0. Keluar
Pilih: 1
Masukkan NIM: 1
Masukkan Nama: Zidan
Mahasiswa Zidan berhasil ditambahkan ke antrian

=== Menu Antrian Layanan Kemahasiswaan ===
1. Daftar Antrian
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Cek Antrian Kosong
8. Kosongkan Antrian
0. Keluar
Pilih: 1
Masukkan NIM: 2
Masukkan Nama: Savero
Mahasiswa Savero berhasil ditambahkan ke antrian
```

```
=== Menu Antrian Layanan Kemahasiswaan ===
1. Daftar Antrian
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Cek Antrian Kosong
8. Kosongkan Antrian
0. Keluar
Pilih: 1
Masukkan NIM: 3
Masukkan Nama: Akmal
Mahasiswa Akmal berhasil ditambahkan ke antrian

=== Menu Antrian Layanan Kemahasiswaan ===
1. Daftar Antrian
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Cek Antrian Kosong
8. Kosongkan Antrian
0. Keluar
Pilih: 5
Daftar antrian:
1 - Zidan
2 - Savero
3 - Akmal
```

```
=== Menu Antrian Layanan Kemahasiswaan ===
1. Daftar Antrian
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Cek Antrian Kosong
8. Kosongkan Antrian
0. Keluar
Pilih: 3
Antrian terdepan: 1 - Zidan

=== Menu Antrian Layanan Kemahasiswaan ===
1. Daftar Antrian
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Cek Antrian Kosong
8. Kosongkan Antrian
0. Keluar
Pilih: 4
Antrian terakhir: 3 - Akmal
```

```
=== Menu Antrian Layanan Kemahasiswaan ===
1. Daftar Antrian
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Cek Antrian Kosong
8. Kosongkan Antrian
0. Keluar
Pilih: 6
Jumlah mahasiswa dalam antrian: 3

=== Menu Antrian Layanan Kemahasiswaan ===
1. Daftar Antrian
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Cek Antrian Kosong
8. Kosongkan Antrian
0. Keluar
Pilih: 7
Antrian tidak kosong
```

```
=== Menu Antrian Layanan Kemahasiswaan ===
1. Daftar Antrian
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Cek Antrian Kosong
8. Kosongkan Antrian
0. Keluar
Pilih: 2
Memanggil: 1 - Zidan

=== Menu Antrian Layanan Kemahasiswaan ===
1. Daftar Antrian
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Cek Antrian Kosong
8. Kosongkan Antrian
0. Keluar
Pilih: 5
Daftar antrian:
2 - Savero
3 - Akmal

=== Menu Antrian Layanan Kemahasiswaan ===
1. Daftar Antrian
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Cek Antrian Kosong
8. Kosongkan Antrian
0. Keluar
Pilih: 8
Antrian telah dikosongkan
```

```
=== Menu Antrian Layanan Kemahasiswaan ===
1. Daftar Antrian
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Cek Antrian Kosong
8. Kosongkan Antrian
0. Keluar
Pilih: 7
Antrian kosong

=== Menu Antrian Layanan Kemahasiswaan ===
1. Daftar Antrian
2. Panggil Antrian
3. Tampilkan Antrian Terdepan
4. Tampilkan Antrian Terakhir
5. Tampilkan Semua Antrian
6. Tampilkan Jumlah Antrian
7. Cek Antrian Kosong
8. Kosongkan Antrian
0. Keluar
Pilih: 0
Program selesai
```