

Laporan Hasil Praktikum
Algoritma Dan Struktur Data
Jobsheet 11



Angel Chelssa Leoni Eka Permatasari
244107020202
1E
Program Studi Teknologi Informasi
Jurusan Teknik Informatika
POLINEMA
2025

PERCOBAAN 1

1. buat package dengan nama Pertemuan12
2. buat

Tambahkan class-class berikut:

- a. Mahasiswa00.java
- b. Node00.java
- c. SingleLinkedList00.java
- d. SLLMain00.java

Ganti 00 dengan nomer Absen Anda

```
J Mahasiswa03.java
J NodeMahasiswa03.java
J SingleLinkList03.java
J SLLMain03.java
```

3. Membuat class Mahasiswa03.java

```
public class Mahasiswa03 {
    String nim;
    String nama;
    String kelas;
    double ipk;

    public Mahasiswa03() {
    }

    public Mahasiswa03(String name, String nm, String kls, double ip) {
        nim = name;
        nama = nm;
        kelas = kls;
        ipk = ip;
    }

    public void tampilInformasi() {
        System.out.println(nama + "\t " + nim + "\t " + kelas + "\t" + ipk );
    }
}
```

4. Implementasi class Node seperti gambar berikut ini

```
public class NodeMahasiswa03 {
    Mahasiswa03 data;
    NodeMahasiswa03 next;

    public NodeMahasiswa03(Mahasiswa03 data, NodeMahasiswa03 next) {
        this.data = data;
        this.next = next;
    }
}
```

5. Tambahkan attribute head dan tail pada class SingleLinkedList

```
public class SingleLinkedList03 {
    NodeMahasiswa03 head;
    NodeMahasiswa03 tail;
```

6. Sebagai langkah berikutnya, akan diimplementasikan method-method yang terdapat pada SingleLinkedList.

7. Tambahkan method isEmpty().

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

8. Implementasi method untuk mencetak dengan menggunakan proses traverse.

```
public void print() {  
    if (!isEmpty()) {  
        NodeMahasiswa03 tmp = head;  
        System.out.println(x:"Isi Linked List:");  
        while (tmp != null) {  
            tmp.data.tampilInformasi();  
            tmp = tmp.next;  
        }  
        System.out.println(x:"");  
    } else {  
        System.out.println(x:"Linked List kosong");  
    }  
}
```

9. Implementasikan method addFirst().

```
public void addFirst(Mahasiswa03 input) {  
    NodeMahasiswa03 ndInput = new NodeMahasiswa03(input, next:null);  
    if (isEmpty()) {  
        head = ndInput;  
        tail = ndInput;  
    }else {  
        ndInput.next = head;  
        head = ndInput;  
    }  
}
```

10. Implementasikan method addLast().

```
public void addLast(Mahasiswa03 input ) {  
    NodeMahasiswa03 ndInput = new NodeMahasiswa03(input, next:null);  
    if (isEmpty()) {  
        head = ndInput;  
        tail = ndInput;  
    }else {  
        tail.next = ndInput;  
        tail = ndInput;  
    }  
}
```

11. Implementasikan method insertAfter, untuk memasukkan node yang memiliki data input setelah node yang memiliki data key.

```
public void insertAfter(String key, Mahasiswa03 input) {  
    NodeMahasiswa03 ndInput = new NodeMahasiswa03(input, next:null);  
    NodeMahasiswa03 tmp = head;  
    do {  
        if (tmp.data.nama.equalsIgnoreCase(key)) {  
            ndInput.next = tmp.next;  
            tmp.next = ndInput;  
            if (ndInput.next == null) {  
                tail = ndInput;  
            }  
            break;  
        }  
        tmp = tmp.next;  
    }while (tmp != null);  
}
```

12. Tambahkan method penambahan node pada indeks tertentu.

```
public void insertArt(int index, Mahasiswa03 input) {
    if (index < 0) {
        System.out.println(x:"indeks salah");
    } else if (index == 0){
        addFirst(input);
    } else {
        NodeMahasiswa03 tmp = head;
        for (int i = 0; i < index - 1; i++) {
            tmp = tmp.next;
        }
        tmp.next = new NodeMahasiswa03(input, tmp.next);
        if (tmp.next.next == null) {
            tail = tmp.next;
        }
    }
}
```

13. Pada class SLLMain00, buatlah fungsi main, kemudian buat object dari class SingleLinkedList.

```
SingleLinkedList03 sll = new SingleLinkedList03();
```

14. Buat empat object mahasiswa dengan nama mhs1, mhs2, mhs3, mhs4 kemudian isi data setiap object melalui konstruktor.

```
Mahasiswa03 mhs1 = new Mahasiswa03(name:"24212200", nm:"Alvaro", kls:"1A", ip:4.0);
Mahasiswa03 mhs2 = new Mahasiswa03(name:"23212201", nm:"Bimon", kls:"2B", ip:3.8);
Mahasiswa03 mhs3 = new Mahasiswa03(name:"22212202", nm:"Cintia", kls:"3C", ip:3.5);
Mahasiswa03 mhs4 = new Mahasiswa03(name:"21212203", nm:"Dirga", kls:"4D", ip:3.6);
```

15. Tambahkan Method penambahan data dan pencetakan data di setiap penambahannya agar terlihat perubahannya

```
sll.print();
sll.addFirst(mhs4);
sll.print();
sll.addLast(mhs1);
sll.print();
sll.insertAfter(key:"Dirga", mhs3);
sll.insertArt(index:2, mhs2);
sll.print();
```

Kode program

Kode program class Mahasiswa03.java

```
public class Mahasiswa03 {
    String nim;
    String nama;
    String kelas;
    double ipk;

    public Mahasiswa03() {
    }

    public Mahasiswa03(String name, String nm, String kls, double ip) {
        nim = name;
        nama = nm;
        kelas = kls;
        ipk = ip;
    }
}
```

```

        public void tampilInformasi() {
            System.out.println(nama + "\t " + nim + "\t " + kelas +
"\t" + ipk );
        }
    }
}

```

Kode program class Node03.java

```

public class Node03 {

    Mahasiswa03 data;
    Node03 next;

    public Node03(Mahasiswa03 data, Node03 next) {
        this.data = data;
        this.next = next;
    }
}

```

Kode program class Singlelinkedlist03.java

```

public class SingleLinkedList03 {;
    NodeMahasiswa03 head;
    NodeMahasiswa03 tail;

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

    public void print() {
        if (!isEmpty()) {
            NodeMahasiswa03 tmp = head;
            System.out.println("Isi Linked List:");
            while (tmp != null) {
                tmp.data.tampilInformasi();
                tmp = tmp.next;
            }
            System.out.println("");
        } else {
            System.out.println("Linked List kosong");
        }
    }

    public void addFirst(Mahasiswa03 input) {
        NodeMahasiswa03 ndInput = new NodeMahasiswa03(input, null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            ndInput.next = head;
            head = ndInput;
        }
    }

    public void addLast(Mahasiswa03 input ) {
        NodeMahasiswa03 ndInput = new NodeMahasiswa03(input, null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            tail.next = ndInput;
            tail = ndInput;
        }
    }
}

```

```

public void insertAfter(String key, Mahasiswa03 input) {
    NodeMahasiswa03 ndInput = new NodeMahasiswa03(input, null);
    NodeMahasiswa03 tmp = head;
    do {
        if (tmp.data.nama.equalsIgnoreCase(key)) {
            ndInput.next = tmp.next;
            tmp.next = ndInput;
            if (ndInput.next == null) {
                tail = ndInput;
            }
            break;
        }
        tmp = tmp.next;
    }while (tmp != null);
}

public void insertArt(int index, Mahasiswa03 input) {
    if (index < 0 ) {
        System.out.println("indeks salah");
    } else if (index == 0){
        addFirst(input);
    } else {
        NodeMahasiswa03 tmp = head;
        for (int i = 0; i < index - 1; i++) {
            tmp = tmp.next;
        }
        tmp.next = new NodeMahasiswa03(input, tmp.next);
        if (tmp.next.next == null) {
            tail = tmp.next;
        }
    }
}
}

```

Kode program class SLLMain03.java

```

public class SLLMain03 {
    public static void main(String[] args) {
        SingleLinkedList03 sll = new SingleLinkedList03();

        Mahasiswa03 mhs1 = new Mahasiswa03("24212200",
        "Alvaro", "1A", 4.0);
        Mahasiswa03 mhs2 = new Mahasiswa03("23212201",
        "Bimon", "2B", 3.8);
        Mahasiswa03 mhs3 = new Mahasiswa03("22212202",
        "Cintia", "3C", 3.5);
        Mahasiswa03 mhs4 = new Mahasiswa03("21212203",
        "Dirga", "4D", 3.6);

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

```

HASIL KODE

```
_ws\josbheet 11\src\bin - SLLMain03
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
Bimon    23212201      2B      3.8
Alvaro   24212200      1A      4.0

PS C:\Users\User\Documents\SEMESTER 2\josbheet 11>
```

PERTANYAAN

1. Karena belum ada data / node yang masuk ke linkList saat fungsi print() pertama dipanggil. Maka akan mencetak "Link List Kosong", karena isinya masih kosong.
`sll.print();`
2. Temp adalah variable sementara yang di pakai untuk berpindah dari kode satu ke kode lainnya tanpa mengubah posisi head atau tail.
3. Modifikasi kode agar dapat memasukkan data secara manual

```
import java.util.Scanner;

public class SLLMain03 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        SingleLinkedList03 sll = new SingleLinkedList03();

        System.out.print("Berapa banyak data mahasiswa? ");
        int jumlah = Integer.parseInt(sc.nextLine());

        for (int i = 0; i < jumlah; i++) {
            System.out.println("Data Mahasiswa ke-" + (i +
1));

            System.out.print("Nama : ");
            String nama = sc.nextLine();
            System.out.print("NIM : ");
            String nim = sc.nextLine();
            System.out.print("Kelas : ");
            String kelas = sc.nextLine();
            System.out.print("IPK : ");
            double ipk = Double.parseDouble(sc.nextLine());

            Mahasiswa03 mhs = new Mahasiswa03(nama, nim,
kelas, ipk);
            sll.addLast(mhs);
        }

        System.out.println("\n== Hasil Linked List ==");
        sll.print();
    }
}
```

HASIL KODE MODIFIKASI

```
jumlah data mahasiswa: 2
Data Mahasiswa ke-1
Nama : angel
NIM : 1234235
Kelas : 1E
IPK : 3.5
Data Mahasiswa ke-2
Nama : eka
NIM : 1243524
Kelas : 1D
IPK : 3.9

== Hasil Linked List ==
Isi Linked List:
1234235 angel 1E 3.5
1243524 eka 1D 3.9

PS C:\Users\User\Documents\SEME
```

PERCOBAAN 2

1. Implementasikan method untuk mengakses data dan indeks pada linked list
2. Tambahkan method untuk mendapatkan data pada indeks tertentu pada class Single Linked List

```
public void getData(int index) {
    NodeMahasiswa03 tmp = head;
    for (int i = 0; i < index; i++) {
        tmp = tmp.next;
    }
    tmp.data.tampilInformasi();
}
```

3. Implementasikan method indexOf.

```
public int indexOf(String key){
    NodeMahasiswa03 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;
    }
}
```

4. Tambahkan method removeFirst pada class SingleLinkedList

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


5. Tambahkan method untuk menghapus data pada bagian belakang pada class SingleLinkedList

```
public void removeLast() {
    if (isEmpty()) {
        System.out.println(x:"Linked List kosong, tidak dapat dihapus");
    } else if (head == tail) {
        head = tail = null;
    } else {
        NodeMahasiswa03 tmp = head;
        while (tmp.next != tail) {
            tmp = tmp.next;
        }
        tmp.next = null;
        tail = tmp;
    }
}
```

6. Sebagai langkah berikutnya, akan diimplementasikan method remove

```
public void remove(String key) {
    if (isEmpty()) {
        System.out.println(x:"Linked List kosong, tidak dapat dihapus");
    } else {
        NodeMahasiswa03 tmp = head;
        while (tmp != null) {
            if ((tmp.data.nama.equalsIgnoreCase(key)) && (tmp == head)) {
                removeFirst();
                break;
            } else if (tmp.data.nama.equalsIgnoreCase(key)) {
                tmp.next = tmp.next.next;
                if (tmp.next == null) {
                    tail = tmp;
                }
                break;
            }
            tmp = tmp.next;
        }
    }
}
```

7. Implementasi method untuk menghapus node dengan menggunakan index.

```
public void removeAt(int index) {
    if (index == 0) {
        removeFirst();
    } else {
        NodeMahasiswa03 tmp = head;
        for (int i = 0; i < index - 1; i++) {
            tmp = tmp.next;
        }
        tmp.next = tmp.next.next;
        if (tmp.next == null) {
            tail = tmp;
        }
    }
}
```

8. Kemudian, coba lakukan pengaksesan dan penghapusan data di method main pada class SLLMain dengan menambahkan kode berikut

```
System.out.println(x:"data index 1 :");
sll.getData(index:1);
System.out.println("data mahasiswa an Bimon berada pada index :" + sll.indexOf(key:"Bimon"));
System.out.println();
sll.removeFirst();
sll.removeLast();
sll.print();
sll.removeAt(index:0);
sll.print();
```

Kode program

Kode program class Mahasiswa03.java

```
public class Mahasiswa03 {
    String nim;
    String nama;
    String kelas;
    double ipk;

    public Mahasiswa03() {
    }

    public Mahasiswa03(String name, String nm, String kls,
double ip) {
        nim = name;
        nama = nm;
        kelas = kls;
        ipk = ip;
    }

    public void tampilInformasi() {
        System.out.println(nama + "\t " + nim + "\t " + kelas
+ "\t" + ipk );
    }
}
```

Kode program class NodeMahasiswa03.java

```
public class NodeMahasiswa03 {

    Mahasiswa03 data;
    NodeMahasiswa03 next;

    public NodeMahasiswa03(Mahasiswa03 data, NodeMahasiswa03
next) {
        this.data = data;
        this.next = next;
    }
}
```

Kode program class SingleLinkedList03.java

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

public class SingleLinkedList03 {;
    NodeMahasiswa03 head;
    NodeMahasiswa03 tail;

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

    public void print() {
        if (!isEmpty()) {
            NodeMahasiswa03 tmp = head;
            System.out.println("Isi Linked List:");
            while (tmp != null) {
                tmp.data.tampilInformasi();
                tmp = tmp.next;
            }
            System.out.println("");
        } else {
            System.out.println("Linked List kosong");
        }
    }
}
```

```

    }

    public void addFirst(Mahasiswa03 input) {
        NodeMahasiswa03 ndInput = new NodeMahasiswa03(input,
null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        }else {
            ndInput.next = head;
            head = ndInput;
        }
    }

    public void addLast(Mahasiswa03 input ) {
        NodeMahasiswa03 ndInput = new NodeMahasiswa03(input,
null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        }else {
            tail.next = ndInput;
            tail = ndInput;
        }
    }

    public void insertAfter(String key, Mahasiswa03 input) {
        NodeMahasiswa03 ndInput = new NodeMahasiswa03(input,
null);
        NodeMahasiswa03 tmp = head;
        do {
            if (tmp.data.nama.equalsIgnoreCase(key)) {
                ndInput.next = tmp.next;
                tmp.next = ndInput;
                if (ndInput.next == null) {
                    tail = ndInput;
                }
                break;
            }
            tmp = tmp.next;
        }while (tmp != null);
    }

    public void insertArt(int index, Mahasiswa03 input) {
        if (index < 0 ) {
            System.out.println("indeks salah");
        } else if (index == 0){
            addFirst(input);
        } else {
            NodeMahasiswa03 tmp = head;
            for (int i = 0; i < index - 1; i++) {
                tmp = tmp.next;
            }
            tmp.next = new NodeMahasiswa03(input, tmp.next);
            if (tmp.next.next == null) {
                tail = tmp.next;
            }
        }
    }

    public void getData(int index) {

```

```

        NodeMahasiswa03 tmp = head;
        for (int i = 0; i < index; i++) {
            tmp = tmp.next;
        }
        tmp.data.tampilInformasi();
    }

    public int indexOf(String key){
        NodeMahasiswa03 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 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 kosong, tidak
dapat dihapus");
        } else if (head == tail) {
            head = tail = null;
        } else {
            NodeMahasiswa03 tmp = head;
            while (tmp.next != tail) {
                tmp = tmp.next;
            }
            tmp.next = null;
            tail = tmp;
        }
    }

    public void remove(String key) {
        if (isEmpty()) {
            System.out.println("Linked List kosong, tidak
dapat dihapus");
        } else {
            NodeMahasiswa03 tmp = head;
            while (tmp != null) {
                if ((tmp.data.nama.equalsIgnoreCase(key)) &&
(tmp == head)) {
                    removeFirst();
                    break;
                }
            }
        }
    }

```

```

        } else if
        (tmp.data.nama.equalsIgnoreCase(key)) {
            tmp.next = tmp.next.next;
            if (tmp.next == null) {
                tail = tmp;
            }
            break;
        }
        tmp = tmp.next;
    }
}

public void removeAt(int index) {
    if (index == 0) {
        removeFirst();
    } else {
        NodeMahasiswa03 tmp = head;
        for (int i = 0; i < index - 1; i++) {
            tmp = tmp.next;
        }
        tmp.next = tmp.next.next;
        if (tmp.next == null) {
            tail = tmp;
        }
    }
}
}

```

Kode program class SLLMain03.java

```

public class SLLMain03 {
    public static void main(String[] args) {
        SingleLinkedList03 sll = new SingleLinkedList03();

        Mahasiswa03 mhs1 = new Mahasiswa03("24212200",
        "Alvaro", "1A", 4.0);
        Mahasiswa03 mhs2 = new Mahasiswa03("23212201",
        "Bimon", "2B", 3.8);
        Mahasiswa03 mhs3 = new Mahasiswa03("22212202",
        "Cintia", "3C", 3.5);
        Mahasiswa03 mhs4 = new Mahasiswa03("21212203",
        "Dirga", "4D", 3.6);

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

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

```

Hasil kode program

```

PS C:\Users\User\Documents\SEMESTER 2\joshheet 11> java -cp .\src\Linkedlist\LinkedListMain.class
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
Bimon 23212201 2B 3.8
Alvaro 24212200 1A 4.0

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

```

PERTANYAAN

1. Kode break unruk menghentikan perulangan Ketika kode yuang akan dihapus ketemu dan berhasil di hapus. Jika tidak ada break akan terus lanjut mengecek node berikutnya, meskipun data yang dicari sudah ketemu. Yang dapat mengakibatkan kode bisa eror jika tidak sengaja ngeakses yang sudah ada. Jadi break penting agar proses efisien dan aman
2. Kode itu dipakai buat hapus node di tengah/akhir, lalu update tail kalo yang dihapus itu node terakhir.

TUGAS

Kode Program

Kode program class MahasiswaTugas03.java

```

public class MahasiswaTugas03 {

    String nama;
    String nim;
    String keperluan;

    public MahasiswaTugas03(String nim, String nama,
String keperluan) {
        this.nim = nim;
        this.nama = nama;
        this.keperluan = keperluan;
    }

    void tampil() {
        System.out.println("NIM : " + nim + " | Nama : " +
nama + " | Keperluan : " + keperluan);
    }
}

```

Kode program class NodeTugas03.java

```

public class NodeTugas03 {

```

```

        MahasiswaTugas03 data;
        NodeTugas03 next;

        public NodeTugas03(MahasiswaTugas03 data) {
            this.data = data;
            this.next = null;
        }
    }

```

Kode program class QueueLinkedListTugas03.java

```

public class QueueLinkedListTugas03 {

    NodeTugas03 front;
    NodeTugas03 rear;
    int size;

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

    public void enqueue(MahasiswaTugas03 data) {
        NodeTugas03 newNode = new NodeTugas03(data);
        if (isEmpty()) {
            front = newNode;
            rear = newNode;
        } else {
            rear.next = newNode;
            rear = newNode;
        }
        size++;
    }

    public MahasiswaTugas03 dequeue() {
        if (isEmpty()) {
            System.out.println("Queue is empty");
            return null;
        }
        MahasiswaTugas03 data = front.data;
        front = front.next;
        size--;
        if (isEmpty()) {
            rear = null;
        }
        return data;
    }

    public MahasiswaTugas03 peekFront() {
        if (isEmpty()) {
            System.out.println("Queue is empty");
            return null;
        }
        return front.data;
    }

    public MahasiswaTugas03 peekRear() {
        if (isEmpty()) {
            System.out.println("Queue is empty");
            return null;
        }
        return rear.data;
    }
}

```

```

        public int getSize() {
            return size;
        }

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

        public void tampilAntrian() {
            if (isEmpty()) {
                System.out.println("Queue is empty");
                return;
            }
            NodeTugas03 current = front;
            System.out.println("Daftar Antrian:");
            while (current != null) {
                current.data.tampil();
                current = current.next;
            }
        }
    }
}

```

Kode program class AntrianMain03.java

```

import java.util.Scanner;
public class AntrianMain03 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        QueueLinkedListTugas03 queue = new
        QueueLinkedListTugas03();
        int pilih;

        do {
            System.out.println("\n===== MENU ANTRIAN
LAYANAN =====");
            System.out.println("1. Tambah Antrian");
            System.out.println("2. Panggil Antrian");
            System.out.println("3. Lihat Antrian Depan");
            System.out.println("4. Lihat Antrian
Belakang");
            System.out.println("5. Lihat Semua Antrian");
            System.out.println("6. Lihat Jumlah Antrian");
            System.out.println("7. Kosongkan Antrian");
            System.out.println("8. Keluar Program");
            System.out.print("Pilih menu: ");
            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();
                    System.out.print("Masukkan Keperluan:
");
                    String keperluan = sc.nextLine();
                    MahasiswaTugas03 mhs = new
                    MahasiswaTugas03(nim, nama, keperluan);

```



```

        queue.enqueue(mhs);
        break;

        case 2:
            MahasiswaTugas03 keluar =
queue.dequeue();
            if (keluar != null) {
                System.out.println("Mahasiswa
berikut telah dipanggil:");
                keluar.tampil();
            }
            break;

        case 3:
            MahasiswaTugas03 depan =
queue.peekFront();
            if (depan != null) {
                System.out.println("Antrian
Terdepan:");
                depan.tampil();
            }
            break;

        case 4:
            MahasiswaTugas03 belakang =
queue.peekRear();
            if (belakang != null) {
                System.out.println("Antrian
Terakhir:");
                belakang.tampil();
            }
            break;

        case 5:
            queue.tampilAntrian();
            break;

        case 6:
            System.out.println("Jumlah Mahasiswa
Dalam Antrian: " + queue.getSize());
            break;

        case 7:
            queue.clear();
            break;

        case 8:
            System.out.println("Terima kasih telah
menggunakan layanan.");
            break;

        default:
            System.out.println("Pilihan tidak
valid.");
    }

    } while (pilih != 8);

    sc.close();
}
}

```

Hasil kode program

```
===== MENU ANTRIAN LAYANAN =====
```

```
1. Tambah Antrian
2. Panggil Antrian
3. Lihat Antrian Depan
4. Lihat Antrian Belakang
5. Lihat Semua Antrian
6. Lihat Jumlah Antrian
7. Kosongkan Antrian
8. Keluar Program
Pilih menu: 1
Masukkan NIM: 12345
Masukkan Nama: Angel
Masukkan Keperluan: KRS
```

```
===== MENU ANTRIAN LAYANAN =====
```

```
1. Tambah Antrian
2. Panggil Antrian
3. Lihat Antrian Depan
4. Lihat Antrian Belakang
5. Lihat Semua Antrian
6. Lihat Jumlah Antrian
7. Kosongkan Antrian
8. Keluar Program
Pilih menu: 1
Masukkan NIM: 678910
Masukkan Nama: Chelssaa
Masukkan Keperluan: NIM
```

```
===== MENU ANTRIAN LAYANAN =====
```

```
1. Tambah Antrian
2. Panggil Antrian
3. Lihat Antrian Depan
4. Lihat Antrian Belakang
5. Lihat Semua Antrian
6. Lihat Jumlah Antrian
7. Kosongkan Antrian
8. Keluar Program
Pilih menu: 5
Daftar Antrian:
NIM : 12345 | Nama : Angel | Keperluan : KRS
NIM : 678910 | Nama : Chelssaa | Keperluan : NIM
```

```
===== MENU ANTRIAN LAYANAN =====
```

```
1. Tambah Antrian
2. Panggil Antrian
3. Lihat Antrian Depan
4. Lihat Antrian Belakang
5. Lihat Semua Antrian
6. Lihat Jumlah Antrian
7. Kosongkan Antrian
8. Keluar Program
Pilih menu: 6
Jumlah Mahasiswa Dalam Antrian: 1
```

```
===== MENU ANTRIAN LAYANAN =====
```

```
1. Tambah Antrian
2. Panggil Antrian
3. Lihat Antrian Depan
4. Lihat Antrian Belakang
5. Lihat Semua Antrian
6. Lihat Jumlah Antrian
7. Kosongkan Antrian
8. Keluar Program
Pilih menu: 7
Antrian telah dikosongkan.
```

```
===== MENU ANTRIAN LAYANAN =====
```

```
1. Tambah Antrian
2. Panggil Antrian
3. Lihat Antrian Depan
4. Lihat Antrian Belakang
5. Lihat Semua Antrian
6. Lihat Jumlah Antrian
7. Kosongkan Antrian
8. Keluar Program
Pilih menu: 2
Mahasiswa berikut telah dipanggil:
NIM : 12345 | Nama : Angel | Keperluan : KRS
```

```
===== MENU ANTRIAN LAYANAN =====
```

```
1. Tambah Antrian
2. Panggil Antrian
3. Lihat Antrian Depan
4. Lihat Antrian Belakang
5. Lihat Semua Antrian
6. Lihat Jumlah Antrian
7. Kosongkan Antrian
8. Keluar Program
Pilih menu: 3
Antrian Terdepan:
NIM : 678910 | Nama : Chelssaa | Keperluan : NIM
```

```
===== MENU ANTRIAN LAYANAN =====
```

```
1. Tambah Antrian
2. Panggil Antrian
3. Lihat Antrian Depan
4. Lihat Antrian Belakang
5. Lihat Semua Antrian
6. Lihat Jumlah Antrian
7. Kosongkan Antrian
8. Keluar Program
Pilih menu: 4
Antrian Terakhir:
NIM : 678910 | Nama : Chelssaa | Keperluan : NIM
```

```
===== MENU ANTRIAN LAYANAN =====
```

```
1. Tambah Antrian
2. Panggil Antrian
3. Lihat Antrian Depan
4. Lihat Antrian Belakang
5. Lihat Semua Antrian
6. Lihat Jumlah Antrian
7. Kosongkan Antrian
8. Keluar Program
Pilih menu: 5
Queue is empty
```

```
===== MENU ANTRIAN LAYANAN =====
```

```
1. Tambah Antrian
2. Panggil Antrian
3. Lihat Antrian Depan
4. Lihat Antrian Belakang
5. Lihat Semua Antrian
6. Lihat Jumlah Antrian
7. Kosongkan Antrian
8. Keluar Program
Pilih menu: 8
Terima kasih telah menggunakan layanan.
```

PS: C:\Users\User\Documents\SEMESTER 2\iosh