

# **Laporan Jobsheet 11 Algoritma dan Struktur Data**

## **Linked List**



244107020151

Ghazwan Ababil

Teknik Informatika / TI – E (11)

Politeknik Negeri Malang

Jurusan Teknologi Informasi

2025

## 1. Praktikum

### 1.1. Percobaan 1: Pembuatan Single Linked List

1. Pada folder Praktikum-ASD membuat folder baru dengan nama Pertemuan12
2. Menambahkan class-class:
  - a. Mahasiswa11.java
  - b. Node11.java
  - c. SingleLingkedList.java
  - d. SLLMain11.java
3. Mengimplementasikan class Mahasiswa11 sesuai dengan class diagram

```
package Pertemuan12;

public class Mahasiswa11 {
    String nim,nama,kelas;
    double ipk;

    public Mahasiswa11(){
    public Mahasiswa11(String nim, String nama, String kelas,
double ipk) {
        this.nama = nama;
        this.nim = nim;
        this.kelas = kelas;
        this.ipk = ipk;
    }

    void tampilInformasi(){
        System.out.printf("%s \t %-18s %-8s \t %.1f\n", nim, nama,
kelas, ipk);
    }
}
```

4. Mengimplementasikan class Node11 sesuai dengan gambar pada langkah percobaan

```
package Pertemuan12;

public class Node11 {
    Mahasiswa11 data;
    Node11 next;

    public Node11(Mahasiswa11 data, Node11 next){
        this.data = data;
        this.next = next;
    }
}
```

5. Pada class SingleLinkedList11, menambahkan attribute head dan tail serta mengimplementasikan method-method yang terdapat pada konsep single linked list seperti:
  - isEmpty(), method untuk mengecek apakah linked list kosong.
  - print(), method untuk mencetak data pada linked list dengan proses traverse
  - addFirst(), untuk menambahkan data pada linked list dari depan

- addLast(), untuk menambahkan data pada linked list dari belakang
- insertAfter(), untuk memasukkan node yang memiliki data input setelah node yang memiliki data key.
- insertAt(), untuk memasukkan node setelah node pada index tertentu.

## 6. Class SingleLinkedList11

```
package Pertemuan12;

public class SingleLinkedList11 {
    Node11 head;
    Node11 tail;

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

    public void print(){
        if (isEmpty()){
            System.out.println("Linked List kosong");
            return;
        }
        Node11 tmp = head;
        System.out.println("Isi Linked List:\t");
        while (tmp != null) {
            tmp.data.tampilInformasi();
            tmp = tmp.next;
        }
        System.out.println();
    }

    public void addFirst(Mahasiswa11 input){
        Node11 nodeInput = new Node11(input, null);
        if(isEmpty()){
            head = nodeInput;
            tail = nodeInput;
        }else{
            nodeInput.next = head;
            head = nodeInput;
        }
    }

    public void addLast(Mahasiswa11 input){
        Node11 nodeInput = new Node11(input, null);
        if(isEmpty()){
            head = nodeInput;
            tail = nodeInput;
        }else{
            tail.next = nodeInput;
            tail = nodeInput;
        }
    }
}
```

```

public void insertAfter(String key, Mahasiswa11 input){
    Node11 nodeInput = new Node11(input, null);
    Node11 temp = head;
    do {
        if (temp.data.nama.equalsIgnoreCase(key)) {
            nodeInput.next = temp.next;
            temp.next = nodeInput;
            if (nodeInput.next == null) tail = nodeInput;
            break;
        }
    } while (temp.next != null);
}

public void insertAt(int index, Mahasiswa11 input){
    if (index < 0) System.out.println("Indeks salah");
    else if (index == 0) addFirst(input);
    else {
        Node11 temp = head;
        for (int i = 0; i < index-1; i++) {
            temp = temp.next;
        }
        temp.next = new Node11(input, temp.next);
        if (temp.next.next == null) tail = temp.next;
    }
}
}

```

7. Menambahkan fungsi main pada class SLLMain11, kemudian membuat object dari class SingleLingkedList. Lalu membuat empat object mahasiswa dengan nama mhs1, mhs2, mhs3, mhs4 dan mengisi data setiap object melalui konstruktor serta menambahkan method penambahan data dan pencetakan data pada setiap penambahannya agar terlihat perubahannya.

```

package Pertemuan12;

public class SLLMain11 {
    public static void main(String[] args) {
        SingleLinkedList11 sll = new SingleLinkedList11();
        Mahasiswa11 mhs1 = new Mahasiswa11("24212200", "Alvaro", "1A", 4.0);
        Mahasiswa11 mhs2 = new Mahasiswa11("23212201", "Bimon", "2B", 3.8);
        Mahasiswa11 mhs3 = new Mahasiswa11("22212202", "Cintia", "3C", 3.5);
        Mahasiswa11 mhs4 = new Mahasiswa11("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();
    }
}

```

## 8. Hasil Percobaan

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

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

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

## 9. Pertanyaan

1. Hasil compile program menghasilkan “Linked List kosong” karena, pada class `SingleLinkedList11` masih belum ada data / belum menambahkan data sama sekali pada class `SingleLinkedList11` tetapi dilakukan method `print` (menampilkan data), sehingga menampilkan pesan “Linked List kosong”.
2. Variabel `temp` pada beberapa method seperti method `insertAt`, `insertAfter`, dan `print` berfungsi untuk melakukan pengaksesan seperti indeks pada data dalam linked list yang dimulai dari head (awal) hingga tail (akhir). Pada method `insertAt`, variabel `temp` digunakan untuk mengakses data pada linked list secara linier seperti indeks untuk menyisipkan data pada indeks yang diinginkan. Kemudian pada method `insertAfter`, variabel `temp` digunakan untuk melakukan pencarian pada data dalam linked list secara linier seperti indeks untuk mencari data yang sesuai dengan kata kunci yang diberikan dan menginputkan datanya. Sedangkan pada method `print`, variabel `temp` digunakan untuk mengakses data pada linked list dan menampilkan data tersebut.

3. Melakukan modifikasi agar data dapat ditambahkan dari keyboard.

Menambahkan fungsi untuk menginput mahasiswa pada class SLLMain11

```
static Scanner sc = new Scanner(System.in);
public static Mahasiswa11 inputMhs() {
    String nim,nama,kelas;
    double ipk;
    do {
        System.out.println("Input Data Mahasiswa");
        System.out.print("NIM \t : ");
        nim = sc.nextLine();
        System.out.print("Nama \t : ");
        nama = sc.nextLine();
        System.out.print("Kelas \t : ");
        kelas = sc.nextLine();
        System.out.print("IPK \t : ");
        ipk = sc.nextDouble();
        if(ipk>4.0 || ipk<0.0)System.out.println("ipk tidak valid");
    } while (ipk > 4.0 || ipk < 0.0);
    return new Mahasiswa11(nim,nama,kelas,ipk);
}
```

### Memodifikasi fungsi main pada class SLLMain11

```
public static void main(String[] args) {
    SingleLinkedList11 sll = new SingleLinkedList11();
    int menu;
    do {
        System.out.println("=== Menu Input Data Mahasiswa ===");
        System.out.println("1. Menambahkan Data Mahasiswa di Depan");
        System.out.println("2. Menambahkan Data Mahasiswa di
Belakang");
        System.out.println("3. Menambahkan Data Mahasiswa Setelah
Data Tertentu");
        System.out.println("4. Menambahkan Data Mahasiswa pada Indeks
Tertentu");
        System.out.println("5. Menampilkan Daftar Mahasiswa");
        System.out.println("6. Keluar");
        System.out.println("=====");
        System.out.print("Pilih Menu: ");
        menu = sc.nextInt(); sc.nextLine();
        switch (menu) {
            case 1:
                sll.addFirst(inputMhs());
                break;
            case 2:
                sll.addLast(inputMhs());
                break;
            case 3:
                System.out.print("Masukkan Nama mahasiswa yang
dicari:");
                String nama = sc.nextLine();
                sll.insertAfter(nama, inputMhs());
                break;
            case 4:
                System.out.print("Masukkan Indeks yang diinginkan:");
                int index = sc.nextInt();
                sc.nextLine();
                sll.insertAt(index, inputMhs());
                break;
            case 5:
                sll.print();
                break;
            case 6:
                System.out.println("Terima kasih");
                break;
            default:
                System.out.println("Menu tidak valid");
                break;
        }
    } while (menu != 6);
}
```

## Hasil run program

```
=== Menu Input Data Mahasiswa ===
1. Menambahkan Data Mahasiswa di Depan
2. Menambahkan Data Mahasiswa di Belakang
3. Menambahkan Data Mahasiswa Setelah Data Tertentu
4. Menambahkan Data Mahasiswa pada Indeks Tertentu
5. Menampilkan Daftar Mahasiswa
6. Keluar
=====
Pilih Menu: 1
Input Data Mahasiswa
NIM      : 21212203
Nama     : Dirga
Kelas   : 4D
IPK      : 3.6
=== Menu Input Data Mahasiswa ===
1. Menambahkan Data Mahasiswa di Depan
2. Menambahkan Data Mahasiswa di Belakang
3. Menambahkan Data Mahasiswa Setelah Data Tertentu
4. Menambahkan Data Mahasiswa pada Indeks Tertentu
5. Menampilkan Daftar Mahasiswa
6. Keluar
=====
Pilih Menu: 5
Isi Linked List:
21212203      Dirga      4D      3.6
24212200      Alvaro     1A      4.0

=== Menu Input Data Mahasiswa ===
1. Menambahkan Data Mahasiswa di Depan
2. Menambahkan Data Mahasiswa di Belakang
3. Menambahkan Data Mahasiswa Setelah Data Tertentu
4. Menambahkan Data Mahasiswa pada Indeks Tertentu
5. Menampilkan Daftar Mahasiswa
6. Keluar
=====
Pilih Menu: 3
Masukkan Nama mahasiswa yang dicari:Dirga
Input Data Mahasiswa
NIM      : 22212202
Nama     : Cintia
Kelas   : 3C
IPK      : 3.5
```



```

=== Menu Input Data Mahasiswa ===
1. Menambahkan Data Mahasiswa di Depan
2. Menambahkan Data Mahasiswa di Belakang
3. Menambahkan Data Mahasiswa Setelah Data Tertentu
4. Menambahkan Data Mahasiswa pada Indeks Tertentu
5. Menampilkan Daftar Mahasiswa
6. Keluar
=====
Pilih Menu: 4
Masukkan Indeks yang diinginkan:2
Input Data Mahasiswa
NIM      : 23212201
Nama     : Bimon
Kelas   : 2B
IPK      : 3.8
=== Menu Input Data Mahasiswa ===
1. Menambahkan Data Mahasiswa di Depan
2. Menambahkan Data Mahasiswa di Belakang
3. Menambahkan Data Mahasiswa Setelah Data Tertentu
4. Menambahkan Data Mahasiswa pada Indeks Tertentu
5. Menampilkan Daftar Mahasiswa
6. Keluar
=====
Pilih Menu: 5
Isi Linked List:
21212203      Dirga      4D      3.6
22212202      Cintia     3C      3.5
23212201      Bimon      2B      3.8
24212200      Alvaro     1A      4.0

=== Menu Input Data Mahasiswa ===
1. Menambahkan Data Mahasiswa di Depan
2. Menambahkan Data Mahasiswa di Belakang
3. Menambahkan Data Mahasiswa Setelah Data Tertentu
4. Menambahkan Data Mahasiswa pada Indeks Tertentu
5. Menampilkan Daftar Mahasiswa
6. Keluar
=====
Pilih Menu: 6
Terima kasih

```

## 1.2. Percobaan 2: Modifikasi Elemen pada Linked List

- Pada percobaan ini melanjutkan untuk mengimplementasikan method pada konsep single linked list pada class `SingleLinkedList11` method-method tersebut antara lain:
  - `getData(int index)`, method untuk mendapatkan data pada indeks tertentu pada class `SingleLinkedList11`
  - `indexOf(String key)`, method untuk mendapatkan posisi indeks dari data tertentu pada class `SingleLinkedList11`
  - `removeFirst()`, method untuk menghapus data terdepan (data head) pada class `SingleLinkedList11`.
  - `removeLast()`, method untuk menghapus data terakhir (data tail) pada class `SingleLinkedList11`.
  - `remove(String key)`, method untuk menghapus data pada class `SingleLinkedList11` yang sesuai dengan kata kunci yang diberikan.
  - `removeAt(int index)`, method untuk menghapus data pada indeks tertentu pada class `SingleLinkedList11`.

2. Mengimplementasikan method-method pada konsep single linked list di atas pada class SingeLinkedList11.

```
public void getData(int index){
    Node11 temp = head;
    for (int i = 0; i < index; i++) {
        temp = temp.next;
    }
    temp.data.tampilInformasi();
}

public int indexOf(String key){
    Node11 temp = head;
    int index = 0;
    while(temp!=null && !temp.data.nama.equalsIgnoreCase(key)){
        temp = temp.next;
        index++;
    }
    if (temp == null) return -1;
    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{
        Node11 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{
        Node11 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{
        Node11 temp = head;
        for (int i = 0; i < index; i++) {
            temp = temp.next;
        }
        temp.next = temp.next.next;
        if (temp.next == null) tail = temp;
    }
}

```

### 3. Menggunakan method untuk mengakses data dan menghapus data pada class

SingleLinkedList11 di dalam class SLLMain11

```

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

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

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

```

### 4. Hasil Percobaan

```

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

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

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

data index 1 :
22212202      Cintia     3C      3.5
data mahasiswa Bimon pada index : 2

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

Isi Linked List:
23212201      Bimon      2B      3.8

PS D:\Java\Praktikum-ASD>

```

## 5. Pertanyaan

1. Pada method remove diperlukan break karena pada method tersebut dilakukan perulangan untuk mencocokkan data dengan kata kunci (variabel key) yang diinputkan. Break digunakan untuk menghentikan perulangan ketika data yang sesuai dengan kata kunci yang diinputkan (variabel key) telah ditemukan.
2. Kode tersebut pada method remove digunakan untuk menghapus data dengan cara mengganti alamat yang menyambungkan data yang ingin dihapus ke alamat data setelahnya. Kemudian bagian kode if berfungsi untuk mengecek setelah data dihapus, apakah data tersebut tidak memiliki data selanjutnya atau data tersebut menjadi data terakhir.

## 2. Tugas

Membuat sistem untuk pengelolaan mahasiswa dengan konsep queue pada linked list

Class AntrianSLL.java

```
package Jobsheet11;

public class AntrianSLL {
    Node11 head, tail;
    int size, max;

    public AntrianSLL(int max){
        head = tail = null;
        size = 0;
        this.max = max;
    }

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

    public boolean isFull(){
        return size == max;
    }

    public void print(){
        if (isEmpty()){
            System.out.println("Antrian kosong\n");
            return;
        }
        Node11 tmp = head;
        System.out.println("Daftar Antrian:\t");
        while (tmp.next != null) {
            tmp.data.tampilInformasi();
            tmp = tmp.next;
        }
        tmp.data.tampilInformasi();
        System.out.println();
    }
}
```

```

public void addLast(Mahasiswa input){
    Node11 nodeInput = new Node11(input, null);
    if (isFull()) {
        System.out.println("Antrian sudah penuh\n");
        return;
    }
    else if(isEmpty()){
        head = tail = nodeInput;
        size++;
    }else{
        tail.next = nodeInput;
        tail = nodeInput;
        size++;
    }
    System.out.println("Mahasiswa berhasil ditambahkan\n");
}

```

```

public void removeFirst(){
    if (isEmpty()) {
        System.out.println("Antrian kosong\n");
        return;
    }
    Mahasiswa mhs = head.data;
    head = head.next;
    size--;
    if (isEmpty()) head = tail = null;
    mhs.tampilInformasi();
}

public void getHead(){
    if (isEmpty()) System.out.println("Antrian kosong\n");
    else {
        System.out.println("Antrian Terdepan: ");
        head.data.tampilInformasi();
    }
}

public void getTail(){
    if (isEmpty()) System.out.println("Antrian kosong\n");
    else {
        System.out.println("Data Paling Belakang: ");
        tail.data.tampilInformasi();
    }
}

public void getSize(){
    System.out.printf("Jumlah Antrian: %s\n\n",size);
}

public void clear(){
    if (isEmpty()) {
        System.out.println("Antrian Kosong\n");
        return;
    }
    head = tail = null;
    size = 0;
    System.out.println("Antrian Berhasil Dikosongkan\n");
}
}

```

## Class AntrianSLLMain

```
package Jobsheet11;
import java.util.Scanner;

public class AntrianSLLMain {
    static Scanner sc = new Scanner(System.in);

    public static Mahasiswa inputMhs(){
        String nim,nama,kelas;
        double ipk;
        do {
            System.out.println("Input Data Mahasiswa");
            System.out.print("NIM \t : ");
            nim = sc.nextLine();
            System.out.print("Nama \t : ");
            nama = sc.nextLine();
            System.out.print("Kelas \t : ");
            kelas = sc.nextLine();
            System.out.print("IPK \t : ");
            ipk = sc.nextDouble();
            if(ipk > 4.0 || ipk < 0.0) System.out.println("ipk tidak valid");
        } while (ipk > 4.0 || ipk < 0.0);
        return new Mahasiswa(nim,nama,kelas,ipk);
    }

    public static void main(String[] args) {
        System.out.print("Masukkan Jumlah Maksimal Antrian: ");
        int max = sc.nextInt();
        AntrianSLL antrian = new AntrianSLL(max);
        int menu;
        do {
            System.out.println("=== Menu Antrian Mahasiswa ===");
            System.out.println("1. Tambah Antrian");
            System.out.println("2. Melayani Antrian");
            System.out.println("3. Tampilkan Antrian");
            System.out.println("4. Tampilkan Antrian Terdepan");
            System.out.println("5. Tampilkan Antrian Terbelakang");
            System.out.println("6. Tampilkan Jumlah Antrian");
            System.out.println("7. Cek Antrian Penuh");
            System.out.println("8. Cek Antrian Kosong");
            System.out.println("9. Kosongkan Antrian");
            System.out.println("10. Keluar");
            System.out.println("=====");
            System.out.print("Pilih Menu: ");
            menu = sc.nextInt();sc.nextLine();
```

```
            switch (menu) {
                case 1:
                    antrian.addLast(inputMhs());
                    break;
                case 2:
                    System.out.println("Melayani Mahasiswa:");
                    antrian.removeFirst();
                    break;
                case 3:
                    antrian.print();
                    break;
                case 4:
                    antrian.getHead();
                    break;
```

```

        case 5:
            antrian.getTail();
            break;
        case 6:
            antrian.getSize();
            break;
        case 7:
            if (antrian.isFull()) System.out.println("Antrian Penuh");
            else System.out.println("Antrian Belum Penuh");
            break;
        case 8:
            if (antrian.isEmpty()) System.out.println("Antrian Kosong");
            else System.out.println("Antrian Tidak Kosong");
            break;
        case 9:
            antrian.clear();
            break;
        case 10:
            System.out.println("Terima Kasih");
            break;
        default:
            System.out.println("Menu tidak valid");
    }
    } while (menu != 10);
}

```

## Class Node11

```

package Jobsheet11;

public class Node11 {
    Mahasiswall data;
    Node11 next;

    public Node11(Mahasiswall data, Node11 next){
        this.data = data;
        this.next = next;
    }
}

```

## Class Mahasiswa1

```
package Jobsheet11;

public class Mahasiswa1 {
    String nim,nama,kelas;
    double ipk;

    public Mahasiswa1(){}
    public Mahasiswa1(String nim, String nama, String kelas, double ipk) {
        this.nama = nama;
        this.nim = nim;
        this.kelas = kelas;
        this.ipk = ipk;
    }

    void tampilInformasi(){
        System.out.printf("%s \t %-18s %-8s \t %.1f\n", nim, nama, kelas,
ipk);
    }
}
```



## Hasil output

```
Masukkan Jumlah Maksimal Antrian: 2
=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 2
Melayani Mahasiswa:
Antrian kosong
```

```
=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 3
Antrian kosong
```

```
=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 4
Antrian kosong
```

```
=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 5
Antrian kosong
```

```
=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 6
Jumlah Antrian: 0
```

```
=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 7
Antrian Belum Penuh
```

```
=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 8
Antrian Kosong
```

```
=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 9
Antrian Kosong
```

```

=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 1
Input Data Mahasiswa
NIM      : 24212200
Nama     : Alvaro
Kelas   : 1A
IPK      : 4.0
Mahasiswa berhasil ditambahkan

```

```

=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 3
Daftar Antrian:
24212200      Alvaro      1A      4.0
23212201      Bimon       2B      3.8

```

```

=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 4
Antrian Terdepan:
24212200      Alvaro      1A      4.0
=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 5
Data Paling Belakang:
23212201      Bimon       2B      3.8

```

```

=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 6
Jumlah Antrian: 2
=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 7
Antrian Penuh

```

```

=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 8
Antrian Tidak Kosong

```

```

=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 1
Input Data Mahasiswa
NIM      : 22212202
Nama     : Cintia
Kelas   : 3C
IPK      : 3.5
Antrian sudah penuh

```

```

=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 9
Antrian Berhasil Dikосongkan

```

```

=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 3
Daftar Antrian:
22212202      Cintia      3C      3.5
23212201      Bimon      2B      3.8

```

```

=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 3
Antrian kosong

```

```

=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 2
Melayani Mahasiswa:
22212202      Cintia      3C      3.5

```

```

=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 3
Daftar Antrian:
23212201      Bimon      2B      3.8

```

```

=== Menu Antrian Mahasiswa ===
1. Tambah Antrian
2. Melayani Antrian
3. Tampilkan Antrian
4. Tampilkan Antrian Terdepan
5. Tampilkan Antrian Terbelakang
6. Tampilkan Jumlah Antrian
7. Cek Antrian Penuh
8. Cek Antrian Kosong
9. Kosongkan Antrian
10. Keluar
=====
Pilih Menu: 10
Terima Kasih

```

Commit dan Push kode program ke github

```
PS D:\Java\Praktikum-ASD> git add .
PS D:\Java\Praktikum-ASD> git commit -m "Jobsheet 11 Single Linked List Tugas Implementasi Queue pada Linked List"
[main a3dc016] Jobsheet 11 Single Linked List Tugas Implementasi Queue pada Linked List
 2 files changed, 178 insertions(+)
 create mode 100644 Jobsheet11/AntrianSLL.java
 create mode 100644 Jobsheet11/AntrianSLLMain.java
PS D:\Java\Praktikum-ASD> git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.71 KiB | 291.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/ghazwanz/Praktikum-ASD.git
 3551255..a3dc016  main -> main
branch 'main' set up to track 'origin/main'.
```