

**Laporan Hasil Praktikum**  
**Algoritma Dan Struktur Data**  
**Jobsheet 12**



Disusun Oleh :

Nama : Fadhil Taufiqurrachman  
NIM : 244107020090  
Kelas : Teknik Informatika 1E

**Program Studi Teknik Informatika**  
**Jurusan Teknologi Informasi**  
**Politeknik Negeri Malang 2025**

## 2.1 Percobaan 1 : Pembuatan Single Linked List

### 2.1.1 Kode Program

Kode program pada class Mahasiswa08 :

```
package Jobsheet12;

public class Mahasiswa08 {
    public String nim, nama, kelas;
    public double ipk;

    public Mahasiswa08() {

    }

    public Mahasiswa08(String nm, String name, String kls, double ip) {
        nim = nm;
        nama = name;
        kelas = kls;
        ipk = ip;
    }

    public void tampilInformasi () {
        System.out.printf("%-15s %-25s %-10s %.1f\n", nama, nim, kelas,
ipk);
    }
}
```

Kode program pada class NodeMahasiswa08 :

```
package Jobsheet12;

public class NodeMahasiswa08 {
    Mahasiswa08 data;
    NodeMahasiswa08 next;

    public NodeMahasiswa08(Mahasiswa08 data, NodeMahasiswa08 next) {
        this.data = data;
        this.next = next;
    }
}
```

Kode program pada class SingleLinkedList08 :

```
package Jobsheet12;

public class SingleLinkedList08 {
    NodeMahasiswa08 head;
    NodeMahasiswa08 tail;

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

    public void print() {
        if (!isEmpty()) {
            NodeMahasiswa08 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(Mahasiswa08 input) {
        NodeMahasiswa08 ndInput = new NodeMahasiswa08(input, null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            ndInput.next = head;
            head = ndInput;
        }
    }
}
```

```

public void addLast(Mahasiswa08 input) {
    NodeMahasiswa08 ndInput = new NodeMahasiswa08(input, null);
    if (isEmpty()) {
        head = ndInput;
        tail = ndInput;
    } else {
        tail.next = ndInput;
        tail = ndInput;
    }
}

public void insertAfter (String key, Mahasiswa08 input) {
    NodeMahasiswa08 ndInput = new NodeMahasiswa08(input, null);
    NodeMahasiswa08 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, Mahasiswa08 input) {
    if (index < 0) {
        System.out.println("Indeks Salah.");
    } else if (index == 0) {
        addFirst(input);
    } else {
        NodeMahasiswa08 temp = head;
        for (int i = 0; i < index - 1; i++) {
            temp = temp.next;
        }
        temp.next = new NodeMahasiswa08(input, temp.next);
        if (temp.next.next == null) {
            tail = temp.next;
        }
    }
}
}

```

### Kode program pada class SLLMain08 :

```
package Jobsheet12;

public class SLLMain08 {
    public static void main(String[] args) {
        SingleLinkedList08 sll = new SingleLinkedList08();

        Mahasiswa08 mhs1 = new Mahasiswa08("21212203", "Dirga", "4D",
3.6);
        Mahasiswa08 mhs2 = new Mahasiswa08("23212201", "Bimon", "2B",
3.8);
        Mahasiswa08 mhs3 = new Mahasiswa08("22212202", "Cintia", "3C",
3.5);
        Mahasiswa08 mhs4 = new Mahasiswa08("24212200", "Alvaro", "1A",
4.0);

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

    }
}
```

### 2.1.2 Verifikasi

Linked List Kosong.

Isi Linked List :

Alvaro	24212200	1A	4.0
--------	----------	----	-----

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

### 2.1.3 Pertanyaan

1. Karena saat program dijalankan pertama kali, belum ada data yang dimasukkan ke dalam linked list. Maka dari itu ketika method print dijalankan, hasilnya adalah berupa pesan “Linked List Kosong.” karena memang belum ada data yang ditambahkan.
2. Variabel temp pada setiap method dalam implementasi Linked List digunakan sebagai pointer sementara untuk melakukan traversal linked list tanpa mengubah pointer utama (head atau tail). Penggunaan variabel temp sangat penting karena jika kita langsung menggunakan pointer head untuk traversal, maka kita akan kehilangan struktur asli dari linked list (Data setelah atau sebelum dari node bisa hilang). Variabel temp juga dapat digunakan untuk melakukan operasi-operasi yang memerlukan akses ke node tertentu dan berurusan dengan head/tail, seperti penambahan node di tengah linked list, penghapusan node, pencarian node berdasarkan nilai tertentu, dan operasi lainnya.
3. Modifikasi kode program pada class SLLMain08.

```
package Jobsheet12;
import java.util.Scanner;
public class SLLMain08 {
    static Scanner input = new Scanner(System.in);
    static SingleLinkedList08 sll = new SingleLinkedList08();
    static int pilihan;
    static String nim, nama, kelas;
    static double ipk;

    public static Mahasiswa08 tambahkanData() {
        while (true) {
            System.out.print("NIM      : ");
            nim = input.nextLine();
            System.out.print("Nama      : ");
            nama = input.nextLine();
            System.out.print("Kelas   : ");
            kelas = input.nextLine();
            System.out.print("IPK      : ");
            ipk = input.nextDouble();
            input.nextLine();
            if (ipk >= 0 && ipk <= 4) {
                break;
            } else {
                System.out.println("IPK Tidak Valid, Silahkan Coba Lagi.");
            }
        }
        return new Mahasiswa08(nim, nama, kelas, ipk);
    }
}
```

```

static void tampilkanMenu() {
    System.out.println("===== Menu Operasi Single Linked List
=====");
    System.out.println("1. Tambah Data Mahasiswa Di Awal");
    System.out.println("2. Tambah Data Mahasiswa Di Akhir");
    System.out.println("3. Tambah Data Mahasiswa Pada Indeks
Tertentu");
    System.out.println("4. Tambah Data Mahasiswa Setelah Data
Tertentu");
    System.out.println("5. Cetak Semua Data Mahasiswa");
    System.out.println("0. Keluar");
    System.out.println("=====
=====");
    System.out.print("Pilih Menu : ");
    pilihan = input.nextInt();
    input.nextLine();
    System.out.println();
}

public static void main(String[] args) {
    do {
        tampilkanMenu();
        switch (pilihan) {
            case 1:
                sll.addFirst(tambahkanData());
                System.out.println("Data Berhasil Ditambahkan Di
Awal.\n");
                break;
            case 2:
                sll.addLast(tambahkanData());
                System.out.println("Data Berhasil Ditambahkan Di
Akhir.\n");
                break;
            case 3:
                System.out.print("Masukkan Indeks : ");
                int index = input.nextInt();
                input.nextLine();
                sll.insertAt(index, tambahkanData());
                System.out.println("Data Berhasil Ditambahkan Di
Indeks " + index + ".\n");
                break;
            case 4:
                System.out.print("Masukkan Nama Mahasiswa Yang
Dicari : ");
                String key = input.nextLine();
                sll.insertAfter(key, tambahkanData());
                System.out.println("Data Berhasil Ditambahkan
Setelah " + key + ".\n");
                break;
            case 5:
                sll.print();
                break;
            case 0:
                System.out.println("Terima Kasih Telah Menggunakan
Program Ini.");
                break;
            default:
                break;
        }
    } while (pilihan != 0);
}
}

```

## Hasil Verifikasi :

```
===== Menu Operasi Single Linked List =====
1. Tambah Data Mahasiswa Di Awal
2. Tambah Data Mahasiswa Di Akhir
3. Tambah Data Mahasiswa Pada Indeks Tertentu
4. Tambah Data Mahasiswa Setelah Data Tertentu
5. Cetak Semua Data Mahasiswa
0. Keluar
=====
Pilih Menu : 1

NIM      : 21212203
Nama     : Dirga
Kelas   : 4D
IPK      : 3.6
Data Berhasil Ditambahkan Di Awal.

===== Menu Operasi Single Linked List =====
1. Tambah Data Mahasiswa Di Awal
2. Tambah Data Mahasiswa Di Akhir
3. Tambah Data Mahasiswa Pada Indeks Tertentu
4. Tambah Data Mahasiswa Setelah Data Tertentu
5. Cetak Semua Data Mahasiswa
0. Keluar
=====
Pilih Menu : 5

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

```
===== Menu Operasi Single Linked List =====
1. Tambah Data Mahasiswa Di Awal
2. Tambah Data Mahasiswa Di Akhir
3. Tambah Data Mahasiswa Pada Indeks Tertentu
4. Tambah Data Mahasiswa Setelah Data Tertentu
5. Cetak Semua Data Mahasiswa
0. Keluar
=====
Pilih Menu : 2

NIM      : 24212200
Nama     : Alvaro
Kelas   : 1A
IPK      : 4.0
Data Berhasil Ditambahkan Di Akhir.

===== Menu Operasi Single Linked List =====
1. Tambah Data Mahasiswa Di Awal
2. Tambah Data Mahasiswa Di Akhir
3. Tambah Data Mahasiswa Pada Indeks Tertentu
4. Tambah Data Mahasiswa Setelah Data Tertentu
5. Cetak Semua Data Mahasiswa
0. Keluar
=====
Pilih Menu : 5

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

```
===== Menu Operasi Single Linked List =====
1. Tambah Data Mahasiswa Di Awal
2. Tambah Data Mahasiswa Di Akhir
3. Tambah Data Mahasiswa Pada Indeks Tertentu
4. Tambah Data Mahasiswa Setelah Data Tertentu
5. Cetak Semua Data Mahasiswa
0. Keluar
=====
Pilih Menu : 4

Masukkan Nama Mahasiswa Yang Dicari : Dirga
NIM      : 22212202
Nama     : Cintia
Kelas   : 3C
IPK      : 3.5
Data Berhasil Ditambahkan Setelah Dirga.

===== Menu Operasi Single Linked List =====
1. Tambah Data Mahasiswa Di Awal
2. Tambah Data Mahasiswa Di Akhir
3. Tambah Data Mahasiswa Pada Indeks Tertentu
4. Tambah Data Mahasiswa Setelah Data Tertentu
5. Cetak Semua Data Mahasiswa
0. Keluar
=====
Pilih Menu : 3

Masukkan Indeks : 2
NIM      : 23212201
Nama     : Bimon
Kelas   : 2B
IPK      : 3.8
Data Berhasil Ditambahkan Di Indeks 2.
```

```
===== Menu Operasi Single Linked List =====
1. Tambah Data Mahasiswa Di Awal
2. Tambah Data Mahasiswa Di Akhir
3. Tambah Data Mahasiswa Pada Indeks Tertentu
4. Tambah Data Mahasiswa Setelah Data Tertentu
5. Cetak Semua Data Mahasiswa
0. Keluar
=====
Pilih Menu : 5

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

===== Menu Operasi Single Linked List =====
1. Tambah Data Mahasiswa Di Awal
2. Tambah Data Mahasiswa Di Akhir
3. Tambah Data Mahasiswa Pada Indeks Tertentu
4. Tambah Data Mahasiswa Setelah Data Tertentu
5. Cetak Semua Data Mahasiswa
0. Keluar
=====
Pilih Menu : 0

Terima Kasih Telah Menggunakan Program Ini.
```



## 2.2 Percobaan 2 : Modifikasi Elemen Pada Single Linked List

### 2.2.1 Kode Program

Menambahkan kode program pada class SingleLinkedList08 :

```
public void getData(int index) {
    NodeMahasiswa08 tmp = head;
    for (int i = 0; i < index; i++) {
        tmp = tmp.next;
    }
    tmp.data.tampilInformasi();
}

public int indexOf(String key) {
    NodeMahasiswa08 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 {
        NodeMahasiswa08 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 {
            NodeMahasiswa08 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 {
            NodeMahasiswa08 temp = head;
            for (int i = 0; i < index - 1; i++) {
                temp = temp.next;
            }
            temp.next = temp.next.next;
            if (temp.next == null) {
                tail = temp;
            }
        }
    }
}

```

**Menambahkan kode program pada class SLLMain08 :**

```

        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();

```

### 2.2.2 Verifikasi

```
Linked List Kosong.

Isi Linked List :
Alvaro      24212200      1A      4.0

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
```

### 2.2.3 Pertanyaan

1. Break pada fungsi remove tersebut digunakan untuk menghentikan proses perulangan (loop) setelah data node yang dicari berhasil ditemukan dan dihapus. Jadi dalam method ini terdapat perulangan yang digunakan untuk mencari node dengan data tertentu yang akan dihapus. Setelah node tersebut ditemukan dan dihapus, tidak ada lagi keperluan untuk melanjutkan perulangan dikarenakan tujuan dari method tersebut sudah tercapai. Maka dari itu di break untuk menghentikan perulangan.
2. Kode pada method remove tersebut berfungsi untuk kasus khusus di mana node yang akan dihapus berada di tengah-tengah linked list. Jadi dengan mengganti node pada temp.next menjadi temp.next.next. Kemudian dilakukan pengecekan apakah temp.next berupa null, dalam artian menjadi list paling akhir. Jika benar maka temp akan langsung ditunjuk menjadi tail.

## 2.3 Latihan Praktikum

### 2.3.1 Kode Program

Kode program pada class MahasiswaAntrian08 :

```
package Jobsheet12;

public class MahasiswaAntrian08 {
    String nim, nama, jurusan, prodi, keperluan;

    public MahasiswaAntrian08(String nim, String nama, String jurusan,
String prodi, String keperluan) {
        this.nim = nim;
        this.nama = nama;
        this.jurusan = jurusan;
        this.prodi = prodi;
        this.keperluan = keperluan;
    }

    public void tampilInformasi() {
        System.out.printf("%-20s %-20s %-30s %-30s %-20s\n", nim, nama,
jurusan, prodi, keperluan);
    }
}
```

Kode program pada class NodeAntrian08 :

```
package Jobsheet12;

public class NodeAntrian08 {
    MahasiswaAntrian08 data;
    NodeAntrian08 next;

    public NodeAntrian08(MahasiswaAntrian08 data, NodeAntrian08 next) {
        this.data = data;
        this.next = next;
    }
}
```

### Kode program pada class LayananAntrian08 :

```
package Jobsheet12;

public class LayananAntrian08 {
    NodeAntrian08 head, tail;
    int size, max;

    public LayananAntrian08(int max) {
        this.max = max;
        head = null;
        tail = null;
        size = 0;
    }

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

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

    public void mengosongkanAntrian() {
        if (!isEmpty()) {
            System.out.println("Seluruh Antrian Mahasiswa Yang Dihapus :
");
            System.out.printf("%-20s %-20s %-30s %-30s %-20s\n", "NIM",
"Nama", "Jurusan", "Prodi", "Keperluan");
            NodeAntrian08 tmp = head;
            while (tmp != null) {
                tmp.data.tampilInformasi();
                tmp = tmp.next;
            }
            head = null;
            tail = null;
            size = 0;
            System.out.println("Antrian Berhasil Dikosongkan.");
        } else {
            System.out.println("Antrian Memang Kosong.");
        }
    }

    public void addLast(MahasiswaAntrian08 input) {
        NodeAntrian08 ndInput = new NodeAntrian08(input, null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            tail.next = ndInput;
            tail = ndInput;
        }
        System.out.println("Data Berhasil Ditambahkan Ke Antrian.");
        size++;
    }
}
```

```

    public void removeFirst() {
        if (!isEmpty()) {
            System.out.println("Antrian Mahasiswa Yang Dipanggil : ");
            System.out.printf("%-20s %-20s %-30s %-30s %-20s\n", "NIM",
"Nama", "Jurusan", "Prodi", "Keperluan");
            head.data.tampilInformasi();
            if (head == tail) {
                head = tail = null;
            } else {
                head = head.next;
            }
            size--;
        } else {
            System.out.println("Antrian Kosong.");
        }
    }

    public void print() {
        if (!isEmpty()) {
            NodeAntrian08 tmp = head;
            System.out.println("Seluruh Antrian Mahasiswa : ");
            System.out.printf("%-20s %-20s %-30s %-30s %-20s\n", "NIM",
"Nama", "Jurusan", "Prodi", "Keperluan");
            while (tmp != null) {
                tmp.data.tampilInformasi();
                tmp = tmp.next;
            }
        } else {
            System.out.println("Antrian Kosong.");
        }
    }

    public void printFirst() {
        if (!isEmpty()) {
            System.out.println("Antrian Mahasiswa Pertama : ");
            System.out.printf("%-20s %-20s %-30s %-30s %-20s\n", "NIM",
"Nama", "Jurusan", "Prodi", "Keperluan");
            head.data.tampilInformasi();
        } else {
            System.out.println("Antrian Kosong.");
        }
    }

    public void printLast() {
        if (!isEmpty()) {
            System.out.println("Antrian Mahasiswa Terakhir : ");
            System.out.printf("%-20s %-20s %-30s %-30s %-20s\n", "NIM",
"Nama", "Jurusan", "Prodi", "Keperluan");
            tail.data.tampilInformasi();
        } else {
            System.out.println("Antrian Kosong.");
        }
    }

    public void printSize() {
        System.out.println("Jumlah Mahasiswa Dalam Antrian : " + size);
    }
}

```



```

        case 3:
            sllAntrian.mengosongkanAntrian();
            break;
        case 4:
            if (sllAntrian.isFull()) {
                System.out.println("Maaf, Antrian Sudah
Penuh.");
                break;
            }
            System.out.println("----- Tambah Mahasiswa Ke
Antrian -----");
            System.out.print("Masukkan NIM          : ");
            String nim = input.nextLine();
            System.out.print("Masukkan Nama          : ");
            String nama = input.nextLine();
            System.out.print("Masukkan Jurusan       : ");
            String jurusan = input.nextLine();
            System.out.print("Masukkan Prodi        : ");
            String prodi = input.nextLine();
            System.out.print("Masukkan Keperluan   : ");
            String keperluan = input.nextLine();
            MahasiswaAntrian08 mahasiswa = new
MahasiswaAntrian08(nim, nama, jurusan, prodi, keperluan);
            sllAntrian.addLast(mahasiswa);
            break;
        case 5:
            sllAntrian.removeFirst();
            break;
        case 6:
            sllAntrian.print();
            break;
        case 7:
            sllAntrian.printFirst();
            break;
        case 8:
            sllAntrian.printLast();
            break;
        case 9:
            sllAntrian.printSize();
            break;
        case 0:
            System.out.println("Terima Kasih Telah Menggunakan
Sistem Layanan Kemahasiswaan.\n");
            break;
        default:
            System.out.println("Pilihan Tidak Valid, Silakan
Coba Lagi!");
    }
    } while (pilihan != 0);
}

```



### 2.3.2 Hasil Output

```
=====
                        Sistem Layanan Kemahasiswaan
=====
1. Cek Antrian Kosong.
2. Cek Antrian Penuh.
3. Menghapus Seluruh Antrian.
4. Tambah Mahasiswa Ke Antrian.
5. Memanggil Antrian Mahasiswa.
6. Tampilkan Semua Antrian.
7. Tampilkan Antrian Terdepan.
8. Tampilkan Antrian Terakhir
9. Tampilkan Jumlah Mahasiswa Dalam Antrian.
0. Keluar.
=====
Pilih Menu : 4

----- Tambah Mahasiswa Ke Antrian -----
Masukkan NIM       : 2024001
Masukkan Nama      : Arion Sky
Masukkan Jurusan   : Ekonomi Dan Bisnis
Masukkan Prodi     : Manajemen
Masukkan Keperluan : Legalisir Transkrip Nilai
Data Berhasil Ditambahkan Ke Antrian.
```

```
=====
                        Sistem Layanan Kemahasiswaan
=====
1. Cek Antrian Kosong.
2. Cek Antrian Penuh.
3. Menghapus Seluruh Antrian.
4. Tambah Mahasiswa Ke Antrian.
5. Memanggil Antrian Mahasiswa.
6. Tampilkan Semua Antrian.
7. Tampilkan Antrian Terdepan.
8. Tampilkan Antrian Terakhir
9. Tampilkan Jumlah Mahasiswa Dalam Antrian.
0. Keluar.
=====
Pilih Menu : 2

Antrian Tidak Penuh.

=====
                        Sistem Layanan Kemahasiswaan
=====
1. Cek Antrian Kosong.
2. Cek Antrian Penuh.
3. Menghapus Seluruh Antrian.
4. Tambah Mahasiswa Ke Antrian.
5. Memanggil Antrian Mahasiswa.
6. Tampilkan Semua Antrian.
7. Tampilkan Antrian Terdepan.
8. Tampilkan Antrian Terakhir
9. Tampilkan Jumlah Mahasiswa Dalam Antrian.
0. Keluar.
=====
Pilih Menu : 1

Antrian Tidak Kosong.
```

4. Tambah Mahasiswa Ke Antrian.
5. Memanggil Antrian Mahasiswa.
6. Tampilkan Semua Antrian.
7. Tampilkan Antrian Terdepan.
8. Tampilkan Antrian Terakhir
9. Tampilkan Jumlah Mahasiswa Dalam Antrian.
0. Keluar.

Pilih Menu : 6

Seluruh Antrian Mahasiswa :

NIM	Nama	Jurusan	Prodi	Keperluan
2024001	Arion Sky	Ekonomi Dan Bisnis	Manajemen	Legalisir Transkrip Nilai
2024002	Elysia Kirana	Teknologi Informasi	Sistem Informasi Bisnis	Konsultasi KRS
2024003	Blade Xepher	Ilmu Komunikasi	Ilmu Komunikasi	Pengajuan Cuti Akademik
2024004	Kaelen Ember	Teknologi Informasi	Teknik Informatika	Pengurusan KIP
2024005	Nova Aruna	Kedokteran	Pendidikan Kedokteran	Pengambilan KTM Baru

#### Sistem Layanan Kemahasiswaan

1. Cek Antrian Kosong.
2. Cek Antrian Penuh.
3. Menghapus Seluruh Antrian.
4. Tambah Mahasiswa Ke Antrian.
5. Memanggil Antrian Mahasiswa.
6. Tampilkan Semua Antrian.
7. Tampilkan Antrian Terdepan.
8. Tampilkan Antrian Terakhir
9. Tampilkan Jumlah Mahasiswa Dalam Antrian.
0. Keluar.

Pilih Menu : 7

Antrian Mahasiswa Pertama :

NIM	Nama	Jurusan	Prodi	Keperluan
2024001	Arion Sky	Ekonomi Dan Bisnis	Manajemen	Legalisir Transkrip Nilai

#### Sistem Layanan Kemahasiswaan

1. Cek Antrian Kosong.
2. Cek Antrian Penuh.
3. Menghapus Seluruh Antrian.
4. Tambah Mahasiswa Ke Antrian.
5. Memanggil Antrian Mahasiswa.
6. Tampilkan Semua Antrian.
7. Tampilkan Antrian Terdepan.
8. Tampilkan Antrian Terakhir
9. Tampilkan Jumlah Mahasiswa Dalam Antrian.
0. Keluar.

Pilih Menu : 8

Antrian Mahasiswa Terakhir :

NIM	Nama	Jurusan	Prodi	Keperluan
2024005	Nova Aruna	Kedokteran	Pendidikan Kedokteran	Pengambilan KTM Baru

#### Sistem Layanan Kemahasiswaan

1. Cek Antrian Kosong.
2. Cek Antrian Penuh.
3. Menghapus Seluruh Antrian.
4. Tambah Mahasiswa Ke Antrian.
5. Memanggil Antrian Mahasiswa.
6. Tampilkan Semua Antrian.
7. Tampilkan Antrian Terdepan.
8. Tampilkan Antrian Terakhir
9. Tampilkan Jumlah Mahasiswa Dalam Antrian.
0. Keluar.

Pilih Menu : 9

Jumlah Mahasiswa Dalam Antrian : 5

```

=====
Sistem Layanan Kemahasiswaan
=====
1. Cek Antrian Kosong.
2. Cek Antrian Penuh.
3. Menghapus Seluruh Antrian.
4. Tambah Mahasiswa Ke Antrian.
5. Memanggil Antrian Mahasiswa.
6. Tampilkan Semua Antrian.
7. Tampilkan Antrian Terdepan.
8. Tampilkan Antrian Terakhir
9. Tampilkan Jumlah Mahasiswa Dalam Antrian.
0. Keluar.
=====
Pilih Menu : 4

Maaf, Antrian Sudah Penuh.

=====
Sistem Layanan Kemahasiswaan
=====
1. Cek Antrian Kosong.
2. Cek Antrian Penuh.
3. Menghapus Seluruh Antrian.
4. Tambah Mahasiswa Ke Antrian.
5. Memanggil Antrian Mahasiswa.
6. Tampilkan Semua Antrian.
7. Tampilkan Antrian Terdepan.
8. Tampilkan Antrian Terakhir
9. Tampilkan Jumlah Mahasiswa Dalam Antrian.
0. Keluar.
=====
Pilih Menu : 5

Antrian Mahasiswa Yang Dipanggil :
NIM          Nama          Jurusan          Prodi          Keperluan
2024001      Arion Sky      Ekonomi Dan Bisnis  Manajemen      Legalisir Transkrip Nilai

```

```

=====
Pilih Menu : 6

Seluruh Antrian Mahasiswa :
NIM          Nama          Jurusan          Prodi          Keperluan
2024002      Elysia Kirana  Teknologi Informasi  Sistem Informasi Bisnis  Konsultasi KRS
2024003      Blade Xepher   Ilmu Komunikasi      Ilmu Komunikasi      Pengajuan Cuti Akademik
2024004      Kaelen Ember   Teknologi Informasi  Teknik Informatika      Pengurusan KIP
2024005      Nova Aruna     Kedokteran           Pendidikan Kedokteran   Pengambilan KTM Baru

=====
Sistem Layanan Kemahasiswaan
=====
1. Cek Antrian Kosong.
2. Cek Antrian Penuh.
3. Menghapus Seluruh Antrian.
4. Tambah Mahasiswa Ke Antrian.
5. Memanggil Antrian Mahasiswa.
6. Tampilkan Semua Antrian.
7. Tampilkan Antrian Terdepan.
8. Tampilkan Antrian Terakhir
9. Tampilkan Jumlah Mahasiswa Dalam Antrian.
0. Keluar.
=====
Pilih Menu : 3

Seluruh Antrian Mahasiswa Yang Dihapus :
NIM          Nama          Jurusan          Prodi          Keperluan
2024002      Elysia Kirana  Teknologi Informasi  Sistem Informasi Bisnis  Konsultasi KRS
2024003      Blade Xepher   Ilmu Komunikasi      Ilmu Komunikasi      Pengajuan Cuti Akademik
2024004      Kaelen Ember   Teknologi Informasi  Teknik Informatika      Pengurusan KIP
2024005      Nova Aruna     Kedokteran           Pendidikan Kedokteran   Pengambilan KTM Baru
Antrian Berhasil Dikосongkan.

```

```
=====
Sistem Layanan Kemahasiswaan
=====
```

1. Cek Antrian Kosong.
2. Cek Antrian Penuh.
3. Menghapus Seluruh Antrian.
4. Tambah Mahasiswa Ke Antrian.
5. Memanggil Antrian Mahasiswa.
6. Tampilkan Semua Antrian.
7. Tampilkan Antrian Terdepan.
8. Tampilkan Antrian Terakhir
9. Tampilkan Jumlah Mahasiswa Dalam Antrian.
0. Keluar.

```
=====
Pilih Menu : 9
```

Jumlah Mahasiswa Dalam Antrian : 0

```
=====
Sistem Layanan Kemahasiswaan
=====
```

1. Cek Antrian Kosong.
2. Cek Antrian Penuh.
3. Menghapus Seluruh Antrian.
4. Tambah Mahasiswa Ke Antrian.
5. Memanggil Antrian Mahasiswa.
6. Tampilkan Semua Antrian.
7. Tampilkan Antrian Terdepan.
8. Tampilkan Antrian Terakhir
9. Tampilkan Jumlah Mahasiswa Dalam Antrian.
0. Keluar.

```
=====
Pilih Menu : 0
```

Terima Kasih Telah Menggunakan Sistem Layanan Kemahasiswaan.