

**Laporan Hasil Praktikum**  
**Algoritma Dan Struktur Data**  
**Jobsheet 11**



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 : Operasi Dasar Queue

### 2.1.1 Kode Program

Kode program pada class Queue08 :

```
package Jobsheet11;

public class Queue08 {
    int[] data;
    int front, rear, size, max;

    public Queue08(int n) {
        max = n;
        data = new int[max];
        size = 0;
        front = rear = -1;
    }

    public boolean isEmpty() {
        if (size == 0) {
            return true;
        } else {
            return false;
        }
    }

    public boolean isFull() {
        if (size == max) {
            return true;
        } else {
            return false;
        }
    }

    public void peek() {
        if (!isEmpty()) {
            System.out.println("Elemen Terdepan : " + data[front]);
        } else {
            System.out.println("Queue Masih Kosong");
        }
    }

    public void print() {
        if (isEmpty()) {
            System.out.println("Queue Masih Kosong");
        } else {
            int i = front;
            while (i != rear) {
                System.out.print(data[i] + " ");
                i = (i + 1) % max;
            }
            System.out.print(data[i] + " ");
            System.out.println("Jumlah Elemen = " + size);
        }
    }
}
```

```

public void clear() {
    if (!isEmpty()) {
        front = rear = -1;
        size = 0;
        System.out.println("Queue Berhasil Dikосongkan");
    } else {
        System.out.println("Queue Masih Kosong");
    }
}

public void enqueue(int dt) {
    if (isFull()) {
        System.out.println("Queue Sudah Penuh");
    } else {
        if (isEmpty()) {
            front = rear = 0;
        } else {
            if (rear == max - 1) {
                rear = 0;
            } else {
                rear++;
            }
        }
        data[rear] = dt;
        size++;
    }
}

public int dequeue() {
    int dt = 0;
    if (isEmpty()) {
        System.out.println("Queue Masih Kosong");
    } else {
        dt = data[front];
        size--;
        if (isEmpty()) {
            front = rear = -1;
        } else {
            if (front == max - 1) {
                front = 0;
            } else {
                front++;
            }
        }
    }
    return dt;
}
}

```

### Kode program pada class QueueMain08 :

```
package Jobsheet11;

import java.util.Scanner;
public class QueueMain08 {
    public static void menu() {
        System.out.println("Masukkan Operasi Yang Diinginkan");
        System.out.println("1. Enqueue");
        System.out.println("2. Dequeue");
        System.out.println("3. Print");
        System.out.println("4. Peek");
        System.out.println("5. Clear");
        System.out.println("-----");
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Masukkan Kapasitas Queue : ");
        int n = input.nextInt();

        Queue08 Q = new Queue08(n);
        int pilih;

        do {
            menu();
            System.out.print("Pilih Menu : ");
            pilih = input.nextInt();
            switch (pilih) {
                case 1:
                    System.out.print("Masukkan Data Baru : ");
                    int dataMasuk = input.nextInt();
                    Q.enqueue(dataMasuk);
                    break;
                case 2:
                    int dataKeluar = Q.dequeue();
                    if (dataKeluar != 0) {
                        System.out.println("Data Yang Dikeluarkan : " +
dataKeluar);
                    }
                    break;
                case 3:
                    Q.print();
                    break;
                case 4:
                    Q.peek();
                    break;
                case 5:
                    Q.clear();
                    break;
                default:
                    System.out.println("Pilihan Tidak Tersedia");
            }
        } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 ||
pilih == 5);
    }
}
```

### 2.1.2 Verifikasi

```
Masukkan Kapasitas Queue : 4
Masukkan Operasi Yang Diinginkan
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
Pilih Menu : 1
Masukkan Data Baru : 15
Masukkan Operasi Yang Diinginkan
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
Pilih Menu : 1
Masukkan Data Baru : 31
Masukkan Operasi Yang Diinginkan
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
Pilih Menu : 4
Elemen Terdepan : 15
```

```
Masukkan Operasi Yang Diinginkan
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
Pilih Menu : 3
15 31 Jumlah Elemen = 2
Masukkan Operasi Yang Diinginkan
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
Pilih Menu : 0
Pilihan Tidak Tersedia
```

### 2.1.3 Pertanyaan

1. Front dan rear merupakan representasi dari indeks, sedangkan size adalah representasi dari jumlah elemen atau data yang ada pada Queue. Itulah alasan front dan rear bernilai awal -1, karena indeks dimulai dari 0, maka untuk inisialisasi nilai awal (Ketika Queue masih kosong) bernilai -1. Sedangkan untuk size merupakan jumlah elemen atau data pada suatu Queue, ketika Queue kosong maka datanya juga belum ada, maka dari itu inisialisasi awalnya adalah 0.
2. Potongan tersebut adalah sebuah kondisi pemilihan di mana ingin menambahkan data jika rear sudah mencapai akhir indeks namun di bagian depan dari Queue masih ada slot kosong (Karena data yang telah ditambahkan di Dequeue). Maka dari itu rear akan kembali ke indeks 0 dan akan menambahkan data secara berlanjut selama Queue belum penuh.

3. Potongan tersebut adalah sebuah kondisi pemilihan di mana ingin menghapus atau mengeluarkan data dari Queue jika front sudah ada pada indeks akhir dan masih ada data yang tersisa. Maka dari itu front akan kembali ke indeks ke 0.
4. Karena bisa terjadi sebuah kondisi dimana data pada indeks ke 0 sudah di Dequeue. Hal ini menyebabkan front akan bergeser ke indeks selanjutnya. Jika print dari indeks ke 0 maka bisa jadi akan mengeprint elemen kosong karena tidak selalu ada datanya. Oleh karena itu print dimulai dari data ke front.
5. Kode tersebut adalah kondisi jika front berada pada di belakang rear. Maka untuk mereset indeks kembali ke 0 menggunakan rumus tersebut dan mencetak data sampai rear.
6. Berikut adalah potongan kode program untuk kondisi jika Queue Overflow, maka akan langsung mereturn kembali ke menu.

```
if (isFull()) {  
    System.out.println("Antrian Penuh, Tidak Dapat Menambah  
Mahasiswa.");  
    return;  
}
```

7. Berikut adalah modifikasi kode program terdapat pada class QueueMain08.

```
switch (pilih) {  
    case 1:  
        if (Q.isFull()) {  
            System.out.println("Queue Overflow, Program  
Berhenti.");  
            return;  
        }  
        System.out.print("Masukkan Data Baru : ");  
        int dataMasuk = input.nextInt();  
        Q.enqueue(dataMasuk);  
        break;  
    case 2:  
        if (Q.isEmpty()) {  
            System.out.println("Queue Underflow, Program  
Berhenti.");  
            return;  
        }  
        int dataKeluar = Q.dequeue();  
        if (dataKeluar != 0) {  
            System.out.println("Data Yang Dikeluarkan : " +  
dataKeluar);  
        }  
        break;  
}
```

## 2.2 Percobaan 2 : Antrian Layanan Akademik

### 2.2.1 Kode Program

Kode program pada class Mahasiswa08 :

```
package Jobsheet11;

public class Mahasiswa08 {
    String nim, nama, prodi, kelas;

    public Mahasiswa08(String nim, String nama, String prodi, String
kelas) {
        this.nim = nim;
        this.nama = nama;
        this.prodi = prodi;
        this.kelas = kelas;
    }

    public void tampilkanData() {
        System.out.println(nim + " - " + nama + " - " + prodi + " - " +
kelas);
    }
}
```

Kode program pada class AntrianLayanan08 :

```
package Jobsheet11;

public class AntrianLayanan08 {
    Mahasiswa08[] data;
    int front, rear, size, max;

    public AntrianLayanan08(int max) {
        this.max = max;
        this.data = new Mahasiswa08[max];
        this.front = 0;
        this.rear = -1;
        this.size = 0;
    }

    public boolean isEmpty() {
        if (size == 0) {
            return true;
        } else {
            return false;
        }
    }

    public boolean isFull() {
        if (size == max) {
            return true;
        } else {
            return false;
        }
    }
}
```

```

public void lihatTerdepan() {
    if (isEmpty()) {
        System.out.println("Antrian Kosong.");
    } else {
        System.out.print("Mahasiswa Terdepan : " );
        System.out.println("NIM - Nama - Prodi - Kelas");
        data[front].tampilkanData();
    }
}

public void tampilkanSemua() {
    if (isEmpty()) {
        System.out.println("Antrian Kosong.");
        return;
    }
    System.out.println("Daftar Mahasiswa Dalam Antrian : ");
    System.out.println("NIM - Nama - Prodi - Kelas");
    for (int i = 0; i < size; i++) {
        int index = (front + i) % max;
        System.out.print((i + 1) + ". ");
        data[index].tampilkanData();
    }
}

public void clear() {
    if (!isEmpty()) {
        front = rear = -1;
        size = 0;
        System.out.println("Queue Berhasil Dikosongkan");
    } else {
        System.out.println("Queue Masih Kosong");
    }
}

public void tambahAntrian(Mahasiswa08 mhs) {
    if (isFull()) {
        System.out.println("Antrian Penuh, Tidak Dapat Menambah Mahasiswa.");
        return;
    }
    rear = (rear + 1) % max;
    data[rear] = mhs;
    size++;
    System.out.println(mhs.nama + " Berhasil Masuk Ke Antrian.");
}

public Mahasiswa08 layaniMahasiswa() {
    if (isEmpty()) {
        System.out.println("Antrian Kosong.");
        return null;
    }
    Mahasiswa08 mhs = data[front];
    front = (front + 1) % max;
    size--;
    return mhs;
}

public int getJumlahAntrian() {
    return size;
}
}

```



### Kode program pada class LayananAkademikSiakad08 :

```
package Jobsheet11;

import java.util.Scanner;
public class LayananAkademikSiakad08 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        AntrianLayanan08 antrian = new AntrianLayanan08(5);
        int pilihan;

        do {
            System.out.println("\n===== Menu Antrian Layanan Akademik
=====");
            System.out.println("1. Tambah Mahasiswa ke Antrian");
            System.out.println("2. Layani Mahasiswa");
            System.out.println("3. Lihat Mahasiswa Terdepan");
            System.out.println("4. Lihat Semua Antrian");
            System.out.println("5. Jumlah Mahasiswa dalam Antrian");
            System.out.println("0. Keluar");
            System.out.print("Pilih menu: ");
            pilihan = input.nextInt();
            input.nextLine();
            switch (pilihan) {
                case 1:
                    System.out.print("NIM      : ");
                    String nim = input.nextLine();
                    System.out.print("Nama      : ");
                    String nama = input.nextLine();
                    System.out.print("Prodi     : ");
                    String prodi = input.nextLine();
                    System.out.print("Kelas    : ");
                    String kelas = input.nextLine();
                    Mahasiswa08 mhs = new Mahasiswa08(nim, nama, prodi,
kelas);

                    antrian.tambahAntrian(mhs);
                    break;
                case 2:
                    Mahasiswa08 dilayani = antrian.layaniMahasiswa();
                    if (dilayani != null) {
                        System.out.println("Melayani Mahasiswa : ");
                        dilayani.tampilkanData();
                    }
                    break;
                case 3:
                    antrian.lihatTerdepan();
                    break;
                case 4:
                    antrian.tampilkanSemua();
                    break;
                case 5:
                    System.out.println("Jumlah Dalam Antrian : " +
antrian.getJumlahAntrian());
                    break;
                case 0:
                    System.out.println("Terima Kasih.");
                    break;
                default:
                    System.out.println("Pilihan Tidak Valid.");
                    break;
            }
        } while (pilihan != 0);
        input.close();
    }
}
```

### 2.2.2 Verifikasi

```
===== Menu Antrian Layanan Akademik =====
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
0. Keluar
Pilih menu: 1
NIM      : 123
Nama     : Aldi
Prodi    : TI
Kelas   : 1A
Aldi Berhasil Masuk Ke Antrian.

===== Menu Antrian Layanan Akademik =====
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
0. Keluar
Pilih menu: 1
NIM      : 124
Nama     : Bobi
Prodi    : TI
Kelas   : 1G
Bobi Berhasil Masuk Ke Antrian.
```

```
===== Menu Antrian Layanan Akademik =====
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
0. Keluar
Pilih menu: 4
Daftar Mahasiswa Dalam Antrian :
NIM - Nama - Prodi - Kelas
1. 123 - Aldi - TI - 1A
2. 124 - Bobi - TI - 1G

===== Menu Antrian Layanan Akademik =====
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
0. Keluar
Pilih menu: 2
Melayani Mahasiswa :
123 - Aldi - TI - 1A
```

```

===== Menu Antrian Layanan Akademik =====
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
0. Keluar
Pilih menu: 4
Daftar Mahasiswa Dalam Antrian :
NIM - Nama - Prodi - Kelas
1. 124 - Bobi - TI - 1G

===== Menu Antrian Layanan Akademik =====
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
0. Keluar
Pilih menu: 5
Jumlah Dalam Antrian : 1

===== Menu Antrian Layanan Akademik =====
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
0. Keluar
Pilih menu: 0
Terima Kasih.

```

### 2.2.3 Pertanyaan

1. Berikut adalah kode modifikasi kode programnya.

Menambahkan method pada class AntrianLayanan08.

```

public void lihatAkhir() {
    if (isEmpty()) {
        System.out.println("Antrian Kosong.");
    } else {
        System.out.print("Mahasiswa Terakhir : " );
        System.out.println("NIM - Nama - Prodi - Kelas");
        data[rear].tampilkanData();
    }
}
}

```

Menambahkan kode program pada class LayananAkademikSiakad08.

```
do {
    System.out.println("\n===== Menu Antrian Layanan Akademik
=====");
    System.out.println("1. Tambah Mahasiswa ke Antrian");
    System.out.println("2. Layani Mahasiswa");
    System.out.println("3. Lihat Mahasiswa Terdepan");
    System.out.println("4. Lihat Semua Antrian");
    System.out.println("5. Jumlah Mahasiswa dalam Antrian");
    System.out.println("6. Cek Antrian Paling Belakang");
    System.out.println("0. Keluar");
    System.out.print("Pilih menu: ");
    pilihan = input.nextInt();
    input.nextLine();

    .....

        case 6:
            antrian.lihatAkhir();
            break;
        case 0:
            System.out.println("Terima Kasih.");
            break;
        default:
            System.out.println("Pilihan Tidak Valid.");
            break;
    }
} while (pilihan != 0);
input.close();
}
```

## 2.3 Latihan Praktikum

### 2.3.1 Kode Program

Kode program pada class MahasiswaKrs08 :

```
package Jobsheet11;

public class MahasiswaKrs08 {
    String nim, nama, kelas, prodi;
    int semester;

    public MahasiswaKrs08(String nim, String nama, String prodi, String
kelas, int semester) {
        this.nim = nim;
        this.nama = nama;
        this.prodi = prodi;
        this.kelas = kelas;
        this.semester = semester;
    }

    public void tampilkanData() {
        System.out.printf("%-10s %-20s %-30s %-10s %-10s\n", nim, nama,
prodi, kelas, semester);
    }
}
```

### Kode program pada class AntrianLayananKrs08 :

```
package Jobsheet11;

public class AntrianLayananKrs08 {
    MahasiswaKrs08[] queueKrs;
    int front, rear, max, totalMahasiswa, size, sudahDilayani,
    belumDilayani;

    public AntrianLayananKrs08(int max) {
        this.max = max;
        queueKrs = new MahasiswaKrs08[max];
        front = rear = -1;
        size = sudahDilayani = 0;
        totalMahasiswa = belumDilayani = 30;
    }

    public boolean antrianKosong() {
        return size == 0;
    }

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

    public void menghapusAntrian() {
        if (antrianKosong()) {
            System.out.println("Antrian Sudah Kosong.\n");
        } else {
            front = rear = -1;
            size = 0;
            System.out.println("Antrian Berhasil Dikosongkan.\n");
        }
    }

    public void menambahkanAntrian(MahasiswaKrs08 mhs) {
        if (sudahDilayani >= totalMahasiswa) {
            System.out.println("Jumlah Mahasiswa Yang Dapat Ditangani  
Mencapai Batas Maksimal (30 Mahasiswa).\n");
            return;
        }
        if (antrianPenuh()) {
            System.out.println("Antrian Sudah Penuh.\n");
        } else {
            if (antrianKosong()) {
                front = rear = 0;
            } else {
                rear = (rear + 1) % max;
            }
            queueKrs[rear] = mhs;
            size++;
            System.out.println("Mahasiswa " + mhs.nama + " Berhasil  
Ditambahkan Ke Antrian.\n");
        }
    }
}
```

```

        public MahasiswaKrs08[] memprosesAntrian() {
            int maxProses = 2;
            if (antrianKosong()) {
                System.out.println("Antrian Kosong, Tidak Ada Mahasiswa Yang
Dapat Diproses.\n");
                return null;
            } else if (size < maxProses) {
                System.out.println("Antrian Kurang Dari 2 Mahasiswa, Tidak
Dapat Memproses KRS.\n");
                return null;
            } else if (sudahDilayani >= totalMahasiswa) {
                System.out.println("Jumlah Mahasiswa Yang Dapat Ditangani
Mencapai Batas Maksimal (30 Mahasiswa).\n");
                return null;
            }
            MahasiswaKrs08[] mahasiswaDiproses = new
MahasiswaKrs08[maxProses];
            System.out.println("===== Memproses 2 Mahasiswa Untuk KRS
=====");
            System.out.printf("%-10s %-20s %-30s %-10s %-10s\n", "NIM",
"Nama", "Prodi", "Kelas", "Semester");
            for (int i = 0; i < maxProses; i++) {
                mahasiswaDiproses[i] = queueKrs[front];
                front = (front + 1) % max;
                size--;
                sudahDilayani++;
                mahasiswaDiproses[i].tampilkanData();
            }
            belumDilayani = totalMahasiswa - sudahDilayani;
            if (size == 0) {
                front = -1;
                rear = -1;
            }
            System.out.println("Proses KRS Berhasil Dilakukan.\n");
            return mahasiswaDiproses;
        }

        public void tampilkanSemuaAntrian() {
            if (antrianKosong()) {
                System.out.println("Antrian Masih kosong.");
                return;
            }
            System.out.println("===== Daftar Mahasiswa Dalam Antrian
=====");
            System.out.printf("%-10s %-20s %-30s %-10s %-10s\n", "NIM",
"Nama", "Prodi", "Kelas", "Semester");
            int temp = front;
            for (int i = 0; i < size; i++) {
                queueKrs[temp].tampilkanData();
                temp = (temp + 1) % max;
            }
            System.out.println();
        }
    }

```

```

    public void tampilkanDuaTerdepan() {
        if (antrianKosong()) {
            System.out.println("Antrian Masih Kosong.");
            return;
        }
        System.out.println("===== Daftar Mahasiswa 2 Terdepan
=====");
        System.out.printf("%-10s %-20s %-30s %-10s %-10s\n", "NIM",
"Nama", "Prodi", "Kelas", "Semester");
        int temp = front;
        for (int i = 0; i < 2; i++) {
            queueKrs[temp].tampilkanData();
            temp = (temp + 1) % max;
        }
        System.out.println();
    }

    public void tampilkanPalingAkhir() {
        if (antrianKosong()) {
            System.out.println("Antrian kosong.");
            return;
        }
        System.out.println("===== Mahasiswa Paling Akhir
=====");
        System.out.printf("%-10s %-20s %-30s %-10s %-10s\n", "NIM",
"Nama", "Prodi", "Kelas", "Semester");
        queueKrs[rear].tampilkanData();
        System.out.println();
    }

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

    public void cetakJumlahSudahDilayani() {
        System.out.println("Jumlah Mahasiswa Yang sudah Melakukan Proses
KRS : " + sudahDilayani);
        System.out.println();
    }

    public void cetakJumlahBelumDilayani() {
        System.out.println("Jumlah Mahasiswa Yang Belum Melakukan Proses
KRS : " + belumDilayani);
        System.out.println();
    }
}

```





```

        case 3:
            antrianKrs.menghapusAntrian();
            break;
        case 4:
            System.out.println("===== Tambah Mahasiswa Ke
Antrian KRS =====");
            System.out.print("NIM      : ");
            String nim = input.nextLine();
            System.out.print("Nama      : ");
            String nama = input.nextLine();
            System.out.print("Prodi     : ");
            String prodi = input.nextLine();
            System.out.print("Kelas   : ");
            String kelas = input.nextLine();
            System.out.print("Semester : ");
            int semester = input.nextInt();
            input.nextLine();
            MahasiswaKrs08 inputMahasiswa = new
MahasiswaKrs08(nim, nama, prodi, kelas, semester);
            antrianKrs.menambahAntrian(inputMahasiswa);
            break;
        case 5:
            antrianKrs.memprosesAntrian();
            break;
        case 6:
            antrianKrs.tampilkanSemuaAntrian();
            break;
        case 7:
            antrianKrs.tampilkanDuaTerdepan();
            break;
        case 8:
            antrianKrs.tampilkanPalingAkhir();
            break;
        case 9:
            antrianKrs.cetakJumlahAntrian();
            break;
        case 10:
            antrianKrs.cetakJumlahSudahDilayani();
            break;
        case 11:
            antrianKrs.cetakJumlahBelumDilayani();
            break;
        case 0:
            System.out.println("Terima Kasih Telah Menggunakan
Layanan Akademik KRS.\n");
            break;
        default:
            System.out.println("Pilihan Tidak Valid.");
            break;
    }
    } while (pilihan != 0);
    input.close();
}
}

```

### 2.3.2 Hasil Output

```
=====
                        Sistem Layanan Akademik KRS
=====
1. Cek Antrian Kosong.
2. Cek Antrian Penuh.
3. Menghapus Seluruh Antrian KRS (Mengosongkan).
4. Tambah Mahasiswa Ke Antrian KRS.
5. Memproses Antrian KRS (2 Mahasiswa).
6. Tampilkan Semua Antrian KRS.
7. Tampilkan 2 Antrian Terdepan.
8. Tampilkan Antrian Paling Akhir
9. Cetak Jumlah Antrian Saat Ini
10. Cetak Jumlah Mahasiswa Yang Sudah Proses KRS.
11. Cetak Jumlah Mahasiswa Yang Belum Proses KRS.
0. Keluar.
=====
Pilih Menu : 4

===== Tambah Mahasiswa Ke Antrian KRS =====
NIM      : 2024001
Nama     : Arion Sky
Prodi    : Teknik Informatika
Kelas   : TI-1L
Semester : 2
Mahasiswa Arion Sky Berhasil Ditambahkan Ke Antrian.
```

```
=====
                        Sistem Layanan Akademik KRS
=====
1. Cek Antrian Kosong.
2. Cek Antrian Penuh.
3. Menghapus Seluruh Antrian KRS (Mengosongkan).
4. Tambah Mahasiswa Ke Antrian KRS.
5. Memproses Antrian KRS (2 Mahasiswa).
6. Tampilkan Semua Antrian KRS.
7. Tampilkan 2 Antrian Terdepan.
8. Tampilkan Antrian Paling Akhir
9. Cetak Jumlah Antrian Saat Ini
10. Cetak Jumlah Mahasiswa Yang Sudah Proses KRS.
11. Cetak Jumlah Mahasiswa Yang Belum Proses KRS.
0. Keluar.
=====
Pilih Menu : 5

Antrian Kurang Dari 2 Mahasiswa, Tidak Dapat Memproses KRS.
```

```
=====
                        Sistem Layanan Akademik KRS
=====
1. Cek Antrian Kosong.
2. Cek Antrian Penuh.
3. Menghapus Seluruh Antrian KRS (Mengosongkan).
4. Tambah Mahasiswa Ke Antrian KRS.
5. Memproses Antrian KRS (2 Mahasiswa).
6. Tampilkan Semua Antrian KRS.
7. Tampilkan 2 Antrian Terdepan.
8. Tampilkan Antrian Paling Akhir
9. Cetak Jumlah Antrian Saat Ini
10. Cetak Jumlah Mahasiswa Yang Sudah Proses KRS.
11. Cetak Jumlah Mahasiswa Yang Belum Proses KRS.
0. Keluar.
=====
Pilih Menu : 4

===== Tambah Mahasiswa Ke Antrian KRS =====
NIM      : 2024002
Nama     : Luna Nova
Prodi    : Sistem Informasi Bisnis
Kelas   : SIB-1K
Semester : 2
Mahasiswa Luna Nova Berhasil Ditambahkan Ke Antrian.

=====
                        Sistem Layanan Akademik KRS
=====
1. Cek Antrian Kosong.
2. Cek Antrian Penuh.
3. Menghapus Seluruh Antrian KRS (Mengosongkan).
4. Tambah Mahasiswa Ke Antrian KRS.
5. Memproses Antrian KRS (2 Mahasiswa).
6. Tampilkan Semua Antrian KRS.
7. Tampilkan 2 Antrian Terdepan.
8. Tampilkan Antrian Paling Akhir
9. Cetak Jumlah Antrian Saat Ini
10. Cetak Jumlah Mahasiswa Yang Sudah Proses KRS.
11. Cetak Jumlah Mahasiswa Yang Belum Proses KRS.
0. Keluar.
=====
Pilih Menu : 4

===== Tambah Mahasiswa Ke Antrian KRS =====
NIM      : 2024003
Nama     : Kaelen Ember
Prodi    : Teknik Informatika
Kelas   : TI-2J
Semester : 4
Mahasiswa Kaelen Ember Berhasil Ditambahkan Ke Antrian.
```

=====

Pilih Menu : 6

===== Daftar Mahasiswa Dalam Antrian =====

NIM	Nama	Prodi	Kelas	Semester
2024001	Arion Sky	Teknik Informatika	TI-1L	2
2024002	Luna Nova	Sistem Informasi Bisnis	SIB-1K	2
2024003	Kaelen Ember	Teknik Informatika	TI-2J	4

=====

Sistem Layanan Akademik KRS

=====

1. Cek Antrian Kosong.
2. Cek Antrian Penuh.
3. Menghapus Seluruh Antrian KRS (Mengosongkan).
4. Tambah Mahasiswa Ke Antrian KRS.
5. Memproses Antrian KRS (2 Mahasiswa).
6. Tampilkan Semua Antrian KRS.
7. Tampilkan 2 Antrian Terdepan.
8. Tampilkan Antrian Paling Akhir
9. Cetak Jumlah Antrian Saat Ini
10. Cetak Jumlah Mahasiswa Yang Sudah Proses KRS.
11. Cetak Jumlah Mahasiswa Yang Belum Proses KRS.
0. Keluar.

=====

Pilih Menu : 7

===== Daftar Mahasiswa 2 Terdepan =====

NIM	Nama	Prodi	Kelas	Semester
2024001	Arion Sky	Teknik Informatika	TI-1L	2
2024002	Luna Nova	Sistem Informasi Bisnis	SIB-1K	2

=====

Pilih Menu : 8

===== Mahasiswa Paling Akhir =====

NIM	Nama	Prodi	Kelas	Semester
2024003	Kaelen Ember	Teknik Informatika	TI-2J	4

=====

Sistem Layanan Akademik KRS

=====

1. Cek Antrian Kosong.
2. Cek Antrian Penuh.
3. Menghapus Seluruh Antrian KRS (Mengosongkan).
4. Tambah Mahasiswa Ke Antrian KRS.
5. Memproses Antrian KRS (2 Mahasiswa).
6. Tampilkan Semua Antrian KRS.
7. Tampilkan 2 Antrian Terdepan.
8. Tampilkan Antrian Paling Akhir
9. Cetak Jumlah Antrian Saat Ini
10. Cetak Jumlah Mahasiswa Yang Sudah Proses KRS.
11. Cetak Jumlah Mahasiswa Yang Belum Proses KRS.
0. Keluar.

=====

Pilih Menu : 9

Jumlah Mahasiswa Dalam Antrian Saat Ini : 3

Pilih Menu : 5

===== Memproses 2 Mahasiswa Untuk KRS =====

NIM	Nama	Prodi	Kelas	Semester
2024001	Arion Sky	Teknik Informatika	TI-1L	2
2024002	Luna Nova	Sistem Informasi Bisnis	SIB-1K	2

Proses KRS Berhasil Dilakukan.

=====

#### Sistem Layanan Akademik KRS

=====

1. Cek Antrian Kosong.
2. Cek Antrian Penuh.
3. Menghapus Seluruh Antrian KRS (Mengosongkan).
4. Tambah Mahasiswa Ke Antrian KRS.
5. Memproses Antrian KRS (2 Mahasiswa).
6. Tampilkan Semua Antrian KRS.
7. Tampilkan 2 Antrian Terdepan.
8. Tampilkan Antrian Paling Akhir
9. Cetak Jumlah Antrian Saat Ini
10. Cetak Jumlah Mahasiswa Yang Sudah Proses KRS.
11. Cetak Jumlah Mahasiswa Yang Belum Proses KRS.
0. Keluar.

=====

Pilih Menu : 10

Jumlah Mahasiswa Yang sudah Melakukan Proses KRS : 2

=====

Pilih Menu : 11

Jumlah Mahasiswa Yang Belum Melakukan Proses KRS : 28

=====

#### Sistem Layanan Akademik KRS

=====

1. Cek Antrian Kosong.
2. Cek Antrian Penuh.
3. Menghapus Seluruh Antrian KRS (Mengosongkan).
4. Tambah Mahasiswa Ke Antrian KRS.
5. Memproses Antrian KRS (2 Mahasiswa).
6. Tampilkan Semua Antrian KRS.
7. Tampilkan 2 Antrian Terdepan.
8. Tampilkan Antrian Paling Akhir
9. Cetak Jumlah Antrian Saat Ini
10. Cetak Jumlah Mahasiswa Yang Sudah Proses KRS.
11. Cetak Jumlah Mahasiswa Yang Belum Proses KRS.
0. Keluar.

=====

Pilih Menu : 3

Antrian Berhasil Dikosongkan.