

**LAPORAN HASIL PRAKTIKUM ALGORITMA
DAN STRUKTUR DATA
JOBSHEET 10**



NAMA :MOHAMAT FAUZI ROHMAN

NIM: 244107020067

KELAS : TI_1E

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2024**

JOBSHEET X

QUEUE

10. Praktikum

10.1 Percobaan 1: Operasi Dasar Queue

10.1.1 Langkah-langkah Percobaan

1. Buat folder baru bernama P1Jobsheet10 di dalam repository Praktikum ASD, kemudian buat class baru dengan nama Queue.
2. Tambahkan atribut-atribut Queue sesuai diagram class, kemudian tambahkan pula konstruktornya seperti berikut ini :

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

3. Buat method isEmpty bertipe boolean yang digunakan untuk mengecek apakah queue kosong.

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

4. Buat method IsFull bertipe boolean yang digunakan untuk mengecek apakah queue sudah penuh.

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

5. Buat method peek bertipe void untuk menampilkan elemen queue pada posisi paling depan.

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

6. Buat method print bertipe void untuk menampilkan seluruh elemen pada queue mulai dari posisi front sampai dengan posisi rear.

```
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.println(data[i] + " ");
        System.out.println("Jumlah elemen = " + size);
    }
}
```

7. Buat method clear bertipe void untuk menghapus semua elemen pada queue

```
public void clear(){
    if (!IsEmpty()) {
        front = rear = -1;
        size = 0;
        System.out.println("Queue berhasil dikosongkan");
    } else{
        System.out.println("Queue masih kosong");
    }
}
```

8. Buat method Enqueue bertipe void untuk menambahkan isi queue dengan parameter dt yang bertipe integer

```
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++;
    }
}
```

9. Buat method Dequeue bertipe int untuk mengeluarkan data pada queue di posisi belakang

```
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;
}
```

10. Selanjutnya, buat class baru dengan nama QueueMain tetap pada package Praktikum1. Buat method menu bertipe void untuk memilih menu program pada saat dijalankan.

```
import java.util.Scanner;
public class QueueMain {

    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("-----");
    }
}
```

11. Buat fungsi main, kemudian deklarasikan Scanner dengan nama sc.

12. Buat variabel n untuk menampung masukan berupa jumlah maksimal elemen yang dapat disimpan pada queue.

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    System.out.print("Masukkan kapasitas queue: ");
    int n = sc.nextInt();
}
```

13. Lakukan instansiasi objek Queue dengan nama Q dengan mengirimkan parameter n sebagai kapasitas elemen queue

```
Queue Q = new Queue(n);
```

14. Deklarasikan variabel dengan nama pilih bertipe integer untuk menampung pilih menu dari pengguna.
15. Lakukan perulangan menggunakan do-while untuk menjalankan program secara terus menerus sesuai masukan yang diberikan. Di dalam perulangan tersebut, terdapat pemilihan kondisi menggunakan switch-case untuk menjalankan operasi queue sesuai dengan masukan pengguna.

```
int pilih;

do{
    menu();
    pilih = sc.nextInt();
    switch (pilih) {
        case 1:
            System.out.print("Masukkan data baru: ");
            int dataMasuk = sc.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;
    }
} while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5);
}
```

16. Compile dan jalankan class QueueMain, kemudian amati hasilnya.

10.1.2 Verifikasi Hasil Percobaan

<pre>Masukkan kapasitas queue: 4 Masukkan operasi yang diinginkan: 1. Enqueue 2. Dequeue 3. Print 4. Peek 5. Clear ----- 1 Masukkan data baru: 15 Masukkan operasi yang diinginkan: 1. Enqueue 2. Dequeue 3. Print 4. Peek 5. Clear ----- 1 Masukkan data baru: 31</pre>	<pre>Masukkan operasi yang diinginkan: 1. Enqueue 2. Dequeue 3. Print 4. Peek 5. Clear ----- 4 Elemen Terdepan: 15 Masukkan operasi yang diinginkan: 1. Enqueue 2. Dequeue 3. Print 4. Peek 5. Clear -----</pre>
--	--

10.1.3 Pertanyaan

1. Pada konstruktor, mengapa nilai awal atribut front dan rear bernilai -1, sementara atribut size bernilai 0?
: Karena -1 pada atribut front dan rear menandakan bahwa antrian masih kosong. Nilai -1 berfungsi sebagai nilai awal yang tidak valid pada indeks array sehingga nilai -1 menunjukkan bahwa antrian belum memiliki data sama sekali. Sedangkan size bernilai 0 karena menandakan bahwa saat dibuat belum ada data yang masuk ke antrian.

2. Pada method Enqueue, jelaskan maksud dan kegunaan dari potongan kode berikut!

```
if (rear == max - 1) {  
    rear = 0;
```

: Maksud dari kode tersebut ialah untuk menandakan bahwa, jika posisi rear berada di posisi terakhir data maka rear akan di set kembali ke indeks awal (0).

3. Pada method Dequeue, jelaskan maksud dan kegunaan dari potongan kode berikut!

```
if (front == max - 1) {  
    front = 0;
```

: Maksud dari kode tersebut ialah untuk menandakan bahwa, jika posisi front berada di posisi terakhir data maka front akan di set kembali ke indeks awal (0).

4. Pada method print, mengapa pada proses perulangan variabel i tidak dimulai dari 0 (int i=0), melainkan int i=front?
: Perulangan tidak selalu mulai dari indeks ke-0 karena front tidak selalu berada di indeks ke-0, karena data di queue tidak selalu dimulai dari indeks 0, tapi dari front, yaitu posisi terdepan tempat data pertama disimpan (yang akan dikeluarkan terlebih dahulu).

5. Perhatikan kembali method print, jelaskan maksud dari potongan kode berikut!

```
i = (i + 1) % max;
```

: i adalah indeks saat ini dan max adalah kapasitas maksimum data. Jika i ditambahkan 1 dan hasilnya sama dengan max, maka indeks i akan kembali ke awal karena sisa dari modulus max adalah 0.

6. Tunjukkan potongan kode program yang merupakan queue overflow!

```
if (IsFull()) {  
    System.out.println("Queue sudah penuh");  
}
```

7. Pada saat terjadi queue overflow dan queue underflow, program tersebut tetap dapat berjalan dan hanya menampilkan teks informasi. Lakukan modifikasi program sehingga pada saat terjadi queue overflow dan queue underflow, program dihentikan!

: Untuk memberhentikan program, tambahkan System.exit(1); setelah print teks informasi

```
if (IsFull()) {  
    System.out.println("Queue sudah penuh");  
    System.exit(1);
```

```
if (IsEmpty()) {  
    System.out.println("Queue masih kosong");  
    System.exit(1);
```

10.2 Percobaan 2: Antrian Layanan Akademik

10.2.1 Langkah-langkah Percobaan

1. Buat folder baru bernama P2Jobsheet10 di dalam repository Praktikum ASD, kemudian buat class baru dengan nama Mahasiswa.
2. Tambahkan atribut-atribut Nasabah seperti pada Class Diagram, kemudian tambahkan pula konstruktornya seperti gambar berikut ini.

```
public class Mahasiswa {  
  
    String nim;  
    String nama;  
    String prodi;  
    String kelas;  
  
    public Mahasiswa(String nim, String nama, String prodi, String kelas){  
        this.nim = nim;  
        this.nama = nama;  
        this.prodi = prodi;  
        this.kelas = kelas;  
    }  
}
```

Dan tambahkan method tampilkanData berikut :

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

3. Salin kode program class Queue pada Praktikum 1 untuk digunakan kembali pada Praktikum 2 ini, ganti nama class-nya dengan AntrianLayanan. Karena pada Praktikum 1, data yang disimpan pada queue hanya berupa array bertipe integer, sedangkan pada Praktikum 2 data yang digunakan adalah object, maka perlu dilakukan modifikasi pada class AntrianLayanan tersebut.

```
public class AntrianLayanan{  
    Mahasiswa [] data;  
    int front;  
    int rear;  
    int size;  
    int max;  
  
    public AntrianLayanan(int max){  
        this.max = max;  
        this.data = new Mahasiswa[max];  
        this.front = 0;  
        this.rear = -1;  
        size = 0;  
    }  
}
```

4. Lakukan modifikasi pada class AntrianLayanan dengan mengubah tipe int[] data menjadi Mahasiswa[] data karena pada kasus ini data yang akan disimpan berupa object Mahasiswa. Modifikasi perlu dilakukan pada atribut, method Enqueue, dan method Dequeue.

```

public void tambahAntrian(Mahasiswa 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 Mahasiswa layaniMahasiswa() {
    if (IsEmpty()) {
        System.out.println("Antrian kosong");
        return null;
    }
    Mahasiswa mhs = data[front];
    front = (front + 1) % max;
    size--;
    return mhs;
}

```

5. Berikutnya method peek dan print yaitu untuk menampilkan data antrian layanan paling depan dan menampilkan semua data antrian layanan.

```

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();
    }
}

```

Ditambahkan dengan method getJumlahAntrian yaitu menampilkan nilai size

```

public int getJumlahAntrian() {
    return size;
}

```

6. Selanjutnya, buat class baru dengan nama LayananAkademikSIKAD tetap pada package yang sama. Buat fungsi main, deklarasi Scanner dengan nama sc.
7. Kemudian lakukan instansiasi objek AntrianLayanan dengan nama antrian dan nilai parameternya adalah nilai maksimal antrian yang ditentukan (misal sama dengan 5).

8. Deklarasikan variabel dengan nama pilihan bertipe integer untuk menampung pilih menu dari pengguna.

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

        AntrianLayanan antrian = new
        AntrianLayanan(5);
        int pilihan;
```

9. Tambahkan kode berikut untuk melakukan perulangan menu sesuai dengan masukan yang diberikan oleh pengguna

```
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 = sc.nextInt();
    sc.nextLine();

    switch (pilihan) {
        case 1:
            System.out.print("NIM   : ");
            String nim = sc.nextLine();
            System.out.print("Nama   : ");
            String nama = sc.nextLine();
            System.out.print("Prodi  : ");
            String prodi = sc.nextLine();
            System.out.print("Kelas : ");
            String kelas = sc.nextLine();
            Mahasiswa mhs = new Mahasiswa(nim, nama, prodi, kelas);
            antrian.tambahAntrian(mhs);
            break;
        case 2:
            Mahasiswa dilayani = antrian.layaniMahasiswa();
            if (dilayani != null) {
                System.out.print("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");
    }
} while (pilihan != 0);
sc.close();
}
```

10.2.2 Verifikasi Hasil Percobaan

```
=== 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.
```

10.2.3 Pertanyaan

1. Lakukan modifikasi program dengan menambahkan method baru bernama LihatAkhir pada class AntrianLayanan yang digunakan untuk mengecek antrian yang berada di posisi belakang. Tambahkan pula daftar menu 6. Cek Antrian paling belakang pada class LayananAkademikSIKAD sehingga method LihatAkhir dapat dipanggil!

Jawab:

- Tambahkan method LihatAkhir pada class AntrianLayanan

```
public Mahasiswa LihatAkhir(){
    if (IsEmpty()) {
        System.out.println("Antrian kosong, tidak ada data di posisi belakang.");
    } else{
        System.out.println("Mahasiswa paling belakang di antrian: ");
        System.out.println("NIM - NAMA - PRODI - KELAS");
        data[rear].tampilkanData();
    }
    return data[rear];
}
```

- Tambahkan juga case 6 pada class main

```
case 6:
    antrian.LihatAkhir();
    break;
```

Hasil:

```
=== 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
6. Cek Antrian Paling Belakang
0. Keluar
Pilih Menu: 1
NIM : 123
Nama : Agus
Prodi : TI
Kelas : 1E
Agus 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
6. Cek Antrian Paling Belakang
0. Keluar
Pilih Menu: 1
NIM : 124
Nama : Bella
Prodi : SIB
Kelas : 1A
Bella 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
6. Cek Antrian Paling Belakang
0. Keluar
Pilih Menu: 6
Mahasiswa paling belakang di antrian:
NIM - NAMA - PRODI - KELAS
124 - Bella - SIB - 1A
```

10.3 Tugas

Class Mahasiswa

```
public class Mahasiswa {

    String nim;
    String nama;
    String prodi;
    String kelas;

    public Mahasiswa(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);
    }
}
```

Class AntrianKRS

```
public class AntrianKRS {
    Mahasiswa [] data;
    int front;
    int rear;
    int size;
    int max;
    int totalProses;

    public AntrianKRS(int max){
        this.max = max;
        this.data = new Mahasiswa[max];
        this.front = 0;
        this.rear = -1;
        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 tambahAntrian(Mahasiswa mhs){
        if (IsFull()) {
            System.out.println("Antrian penuh, tidak dapat menambah mahasiswa lagi");
            return;
        }
        rear = (rear + 1) % max;
        data[rear] = mhs;
        size++;
        System.out.println(mhs.nama + " berhasil masuk ke antrian.");
    }
}
```

```

public void layaniKRS(){
    if (size < 2) {
        System.out.println("Antrian kurang dari 2, tidak bisa diproses");
        return;
    }
    System.out.println("Memproses KRS untuk 2 Mahasiswa: ");
    for (int i = 0; i < 2; i++){
        Mahasiswa mhs = data[front];
        front = (front + 1) % max;
        size--;
        data[front].tampilkanData();
        totalProses++;
    }
}

public void lihatTerdepan(){
    if (size < 2) {
        System.out.println("Antrian kurang dari 2, tidak bisa diproses");
        return;
    }
    System.out.println("Daftar Antrian 2 Mahasiswa Terdepan: ");
    System.out.println("NIM - NAMA - PRODI - KELAS");
    for (int i = 0; i < 2; i++){
        int index = (front + i) % max;
        data[index].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 Mahasiswa LihatAkhir(){
    if (IsEmpty()) {
        System.out.println("Antrian kosong, tidak ada data di posisi belakang.");
    } else{
        System.out.println("Mahasiswa paling belakang di antrian: ");
        System.out.println("NIM - NAMA - PRODI - KELAS");
        data[rear].tampilkanData();
    }
    return data[rear];
}

public void clear(){
    if (!IsEmpty()) {
        front = rear = -1;
        size = 0;
        System.out.println("Antrian berhasil dikosongkan");
    } else{
        System.out.println("Antrian masih kosong");
    }
}

public int getJumlahAntrian(){
    return size;
}

```

```
public int getJmlProses(){
    return totalProses;
}
}
```

Class LayananKRS

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

        AntrianKRS antrian = new AntrianKRS(5);
        int pilihan;

        do {
            System.out.println("\n=== Menu Antrian Kartu Rencana Studi ===");
            System.out.println("1. Tambah Mahasiswa ke Antrian KRS.");
            System.out.println("2. Memproses KRS.");
            System.out.println("3. Tampilkan Semua Antrian KRS.");
            System.out.println("4. Tampilkan 2 Antrian Terdepan.");
            System.out.println("5. Tampilkan Antrian Paling Akhir.");
            System.out.println("6. Cek Jumlah Antrian KRS.");
            System.out.println("7. Cek Jumlah Proses KRS.");
            System.out.println("8. Kosongkan Antrian");
            System.out.println("0. Keluar");
            System.out.print("Pilih Menu: ");
            pilihan = sc.nextInt();
            sc.nextLine();

            switch (pilihan) {
                case 1:
                    System.out.print("NIM    : ");
                    String nim = sc.nextLine();
                    System.out.print("Nama    : ");
                    String nama = sc.nextLine();
                    System.out.print("Prodi   : ");
                    String prodi = sc.nextLine();
                    System.out.print("Kelas  : ");
                    String kelas = sc.nextLine();
                    Mahasiswa mhs = new Mahasiswa(nim, nama, prodi, kelas);
                    antrian.tambahAntrian(mhs);
                    break;

                case 2:
                    antrian.layaniKRS();
                    break;

                case 3:
                    antrian.tampilkanSemua();
                    break;

                case 4:
                    antrian.lihatTerdepan();
                    break;

                case 5:
                    antrian.LihatAkhir();
                    break;

                case 6:
                    System.out.println("Jumlah Mahasiswa pada Antrian KRS: " + antrian.getJumlahAntrian());
                    break;
            }
        } while (pilihan != 0);
    }
}
```

```

        case 7:
            System.out.println("Jumlah Mahasiswa yang sudah Melakukan Proses KRS: " +
                antrian.getJmlProses());
            break;

        case 8:
            antrian.clear();
            break;

        case 0:
            System.out.println("Terima Kasih.");
            break;

        default:
            System.out.println("Pilihan tidak valid");
    }
} while (pilihan != 0);
sc.close();
}
}

```

Output :

```

=== Menu Antrian Kartu Rencana Studi ===
1. Tambah Mahasiswa ke Antrian KRS.
2. Memproses KRS.
3. Tampilkan Semua Antrian KRS.
4. Tampilkan 2 Antrian Terdepan.
5. Tampilkan Antrian Paling Akhir.
6. Cek Jumlah Antrian KRS.
7. Cek Jumlah Proses KRS.
8. Kosongkan Antrian
0. Keluar
Pilih Menu: 8
Antrian masih kosong

```

```

=== Menu Antrian Kartu Rencana Studi ===
1. Tambah Mahasiswa ke Antrian KRS.
2. Memproses KRS.
3. Tampilkan Semua Antrian KRS.
4. Tampilkan 2 Antrian Terdepan.
5. Tampilkan Antrian Paling Akhir.
6. Cek Jumlah Antrian KRS.
7. Cek Jumlah Proses KRS.
8. Kosongkan Antrian
0. Keluar
Pilih Menu: 1
NIM : 111
Nama : Fauzi
Prodi : TI
Kelas : 1E
Fauzi berhasil masuk ke antrian.

```

```

=== Menu Antrian Kartu Rencana Studi ===
1. Tambah Mahasiswa ke Antrian KRS.
2. Memproses KRS.
3. Tampilkan Semua Antrian KRS.
4. Tampilkan 2 Antrian Terdepan.
5. Tampilkan Antrian Paling Akhir.
6. Cek Jumlah Antrian KRS.
7. Cek Jumlah Proses KRS.
8. Kosongkan Antrian
0. Keluar
Pilih Menu: 2
Antrian kurang dari 2, tidak bisa diproses

```

```

=== Menu Antrian Kartu Rencana Studi ===
1. Tambah Mahasiswa ke Antrian KRS.
2. Memproses KRS.
3. Tampilkan Semua Antrian KRS.
4. Tampilkan 2 Antrian Terdepan.
5. Tampilkan Antrian Paling Akhir.
6. Cek Jumlah Antrian KRS.
7. Cek Jumlah Proses KRS.
8. Kosongkan Antrian
0. Keluar
Pilih Menu: 1
NIM : 222
Nama : Agus
Prodi : TI
Kelas : 1A
Agus berhasil masuk ke antrian.

```

```

=== Menu Antrian Kartu Rencana Studi ===
1. Tambah Mahasiswa ke Antrian KRS.
2. Memproses KRS.
3. Tampilkan Semua Antrian KRS.
4. Tampilkan 2 Antrian Terdepan.
5. Tampilkan Antrian Paling Akhir.
6. Cek Jumlah Antrian KRS.
7. Cek Jumlah Proses KRS.
8. Kosongkan Antrian
0. Keluar
Pilih Menu: 1
NIM : 333
Nama : Bella
Prodi : TI
Kelas : 1D
Bella berhasil masuk ke antrian.

```

```

=== Menu Antrian Kartu Rencana Studi ===
1. Tambah Mahasiswa ke Antrian KRS.
2. Memproses KRS.
3. Tampilkan Semua Antrian KRS.
4. Tampilkan 2 Antrian Terdepan.
5. Tampilkan Antrian Paling Akhir.
6. Cek Jumlah Antrian KRS.
7. Cek Jumlah Proses KRS.
8. Kosongkan Antrian
0. Keluar
Pilih Menu: 3
Daftar Mahasiswa dalam Antrian:
NIM - NAMA - PRODI - KELAS
1. 111 - Fauzi - TI - 1E
2. 222 - Agus - TI - 1A
3. 333 - Bella - TI - 1D

```

```

=== Menu Antrian Kartu Rencana Studi ===
1. Tambah Mahasiswa ke Antrian KRS.
2. Memproses KRS.
3. Tampilkan Semua Antrian KRS.
4. Tampilkan 2 Antrian Terdepan.
5. Tampilkan Antrian Paling Akhir.
6. Cek Jumlah Antrian KRS.
7. Cek Jumlah Proses KRS.
8. Kosongkan Antrian
0. Keluar
Pilih Menu: 4
Daftar Antrian 2 Mahasiswa Terdepan:
NIM - NAMA - PRODI - KELAS
111 - Fauzi - TI - 1E
222 - Agus - TI - 1A

```

```

=== Menu Antrian Kartu Rencana Studi ===
1. Tambah Mahasiswa ke Antrian KRS.
2. Memproses KRS.
3. Tampilkan Semua Antrian KRS.
4. Tampilkan 2 Antrian Terdepan.
5. Tampilkan Antrian Paling Akhir.
6. Cek Jumlah Antrian KRS.
7. Cek Jumlah Proses KRS.
8. Kosongkan Antrian
0. Keluar
Pilih Menu: 5
Mahasiswa paling belakang di antrian:
NIM - NAMA - PRODI - KELAS
333 - Bella - TI - 1D

```

```

=== Menu Antrian Kartu Rencana Studi ===
1. Tambah Mahasiswa ke Antrian KRS.
2. Memproses KRS.
3. Tampilkan Semua Antrian KRS.
4. Tampilkan 2 Antrian Terdepan.
5. Tampilkan Antrian Paling Akhir.
6. Cek Jumlah Antrian KRS.
7. Cek Jumlah Proses KRS.
8. Kosongkan Antrian
0. Keluar
Pilih Menu: 2
Memproses KRS untuk 2 Mahasiswa:
222 - Agus - TI - 1A
333 - Bella - TI - 1D

```

=== Menu Antrian Kartu Rencana Studi ===

1. Tambah Mahasiswa ke Antrian KRS.
2. Memproses KRS.
3. Tampilkan Semua Antrian KRS.
4. Tampilkan 2 Antrian Terdepan.
5. Tampilkan Antrian Paling Akhir.
6. Cek Jumlah Antrian KRS.
7. Cek Jumlah Proses KRS.
8. Kosongkan Antrian
0. Keluar

Pilih Menu: 3

Daftar Mahasiswa dalam Antrian:

NIM - NAMA - PRODI - KELAS

1. 333 - Bella - TI - 1D

=== Menu Antrian Kartu Rencana Studi ===

1. Tambah Mahasiswa ke Antrian KRS.
2. Memproses KRS.
3. Tampilkan Semua Antrian KRS.
4. Tampilkan 2 Antrian Terdepan.
5. Tampilkan Antrian Paling Akhir.
6. Cek Jumlah Antrian KRS.
7. Cek Jumlah Proses KRS.
8. Kosongkan Antrian
0. Keluar

Pilih Menu: 6

Jumlah Mahasiswa pada Antrian KRS: 1

=== Menu Antrian Kartu Rencana Studi ===

1. Tambah Mahasiswa ke Antrian KRS.
2. Memproses KRS.
3. Tampilkan Semua Antrian KRS.
4. Tampilkan 2 Antrian Terdepan.
5. Tampilkan Antrian Paling Akhir.
6. Cek Jumlah Antrian KRS.
7. Cek Jumlah Proses KRS.
8. Kosongkan Antrian
0. Keluar

Pilih Menu: 7

Jumlah Mahasiswa yang sudah Melakukan Proses KRS: 2

=== Menu Antrian Kartu Rencana Studi ===

1. Tambah Mahasiswa ke Antrian KRS.
2. Memproses KRS.
3. Tampilkan Semua Antrian KRS.
4. Tampilkan 2 Antrian Terdepan.
5. Tampilkan Antrian Paling Akhir.
6. Cek Jumlah Antrian KRS.
7. Cek Jumlah Proses KRS.
8. Kosongkan Antrian
0. Keluar

Pilih Menu: 8

Antrian berhasil dikosongkan

=== Menu Antrian Kartu Rencana Studi ===

1. Tambah Mahasiswa ke Antrian KRS.
2. Memproses KRS.
3. Tampilkan Semua Antrian KRS.
4. Tampilkan 2 Antrian Terdepan.
5. Tampilkan Antrian Paling Akhir.
6. Cek Jumlah Antrian KRS.
7. Cek Jumlah Proses KRS.
8. Kosongkan Antrian
0. Keluar

Pilih Menu: 3

Antrian kosong

=== Menu Antrian Kartu Rencana Studi ===

1. Tambah Mahasiswa ke Antrian KRS.
2. Memproses KRS.
3. Tampilkan Semua Antrian KRS.
4. Tampilkan 2 Antrian Terdepan.
5. Tampilkan Antrian Paling Akhir.
6. Cek Jumlah Antrian KRS.
7. Cek Jumlah Proses KRS.
8. Kosongkan Antrian
0. Keluar

Pilih Menu: 0

Terima Kasih.