Laporan Hasil Praktikum Algoritma Dan Struktur Data

Jobsheet 10

Nama : Arifah Zhafirah Wikananda

NIM : 244107020188

Kelas : 1E

Program Studi Teknologi

Informasi Jurusan Teknik

Informatika Politeknik Negeri

Malang

2025

2.1 Percobaan 1 : Operasi Dasar Queue

```
package jobsheet10;

public class Queue04 {
    int[] data;
    int front;
    int rear;
    int size;
    int max;

    public Queue04(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.println(data[i] + " ");
            System.out.println("Jumlah elemen = " + size);
        }
    }
```

```
    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;
    }
}
```

```
package jobsheet10;
import java.util.Scanner;
public class QueueMain04 {
    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 sc = new Scanner(System.in);

        System.out.print("Masukkan kapasitas queue: ");
        int n = sc.nextInt();
        Queue04 Q = new Queue04(n);
        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);
    }
}
```

Hasil dari langkah percobaan diatas:

```
c:\Praktikum ASD> c: && cd "c:\Praktikum ASD" && cmd
ssages -cp "C:\Users\WINDOWS 11\AppData\Roaming\Code
um ASD_c51dcdc\bin" jobsheet10.QueueMain04 "
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
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
--------------------------------
4
Elemen terdepan: 15
```

2.1.3. Pertanyaan

1. Pada konstruktor, mengapa nilai awal atribut front dan rear bernilai -1, sementara atribut size bernilai 0?

Jawab:

1. Front dan rear = -1
   Jika nilai -1 digunakan untuk menunjukkan nilai status awal queue saat nilai masih kosong dan belum masuk
   - Maka, Front = -1
     Front adalah indeks elemen pertama atau yang paling depan, dengan nilai awal -1 jadi bermakna tidak ada elemen di depan atau queue kosong.
   - Dan Rear = -1
     Rear adalah indeks elemen terakhir atau yang paling terakhir atau yang paling belakang dengan nilai awal -1 jadi bermakna tidak ada elemen dibelakang atau queue kosong.
2. Size = 0
   Nilai ini menunjukkan bahwa elemen yang ada di dalam queue adalah nol, karena size yang digunakan untuk memeriksa kondisi kosong atau penuh. Berbeda dengan front dan rear yang bekerja pada indeks array atau bisa juga bemakna belum ada data yang masuk ke dalam queue.

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

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

Jawab:
1. Maksud kode tersebut digunakan untuk mengimplementasikan antrian melingkar atau bisa disebut circular queue. Dalam antrian tersebut, ketika sebuah elemen ditambahkan ke antrian dan posisi rear mencapai indeks terakhir dari array yaitu max – 1, maka akan diseret ke 0. Ini memungkinkan antrian untuk menggunakan kembali ruang yang telah dibebaskan ketika elemen dihapus dari sebuah bagian depan dari antrian tersebut.

2. Kegunaannya yaitu:

- Pada antrian melingkar memanfaatkan ruang yang telah dibebaskan setelah elemen dihapus, sehingga menghindari pemborosan ruang.
- Dengan mengatur pada indeks rear dan front kembali ke indeks 0 saat mencapai batas maksimum.
- Operasi berkelanjutan pada antrian melingkar mendukung operasi penambahan dan penghapusan elem secara efisien tanpa perlu memindahkan elem dalam array.

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

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

Jawan:
1. Maksud kode tersebut digunakan untuk mengimplementasikan antrian melingkar atau bisa disebut circular queue saat menghapus elemen adan antrian. Ketika sebuah elemen dihapus dari sebuah antrian dan posisi front mencapai indeks terakhir dari array max – 1, maka front akan diseret ke 0.

2. Kegunaannya yaitu:

- Antrian biasa tidak dapat memanfaatkan ruang yang telah dibebaskan setelah elemen dihapus.
- Memastikan indeks valid, pada antrian melingkar mengantri indeks front dan rear untuk tetap berada dalam rentang yang valid 0 hingga max – 1.
- Mengurangi overhead dan meningkatkan kinerja, terutama dalam aplikasi yang memerlukan operasi antrian yang cepat.

4. Pada method print, mengapa pada proses perulangan variabel i tidak dimulai dari 0 (int i=0), melainkan int i=front?
Jawab: jika i = front dalam antrian/queue maka data masuk dari belakang atau rear dan keluar dari depan front. Ketika mencetak isi antrian, hanya ingin mencetak elemen yang masih ada dalam antrian. Jika int 1 = 0 maka dalam perulangan untuk mencetak isi dari sebuah antrian tersebut, maka akan mencetak elemen yang sudah keluar dari antrian dan data tidak valid lagi.

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

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

Jawab: Dalam potongan kode tesebut, dapat digunakan untuk mengimplementasikan antrian melingkar saat menginterasi elemen – elemen dalam antrian. Dengan penjelasan kode i + 1 berfungsi untuk menaikkan indeks ke elemen berikutnya, sedangkan % max berfungsi untuk emmastikan indeks tersebut tetap berada dalam batas ukuran array. Dengan demikian, jika i sudah mencapai indeks terakhir atau mas – 1, maka (i+1) % akan kembali ke-0.

6. Tunjukkan potongan kode program yang merupakan queue overflow!

```
public void Enqueue(int dt) {
    if (IsFull()) {
        System.out.println(x:"Queue sudah penuh");
    }else {
        if (IsEmpty()) {
```

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!

```
public void Enqueue(int dt) {
    if (IsFull()) {
        System.out.println(x:"Queue sudah penuh");
        System.exit(status:0);
```

```
public int Dequeue() {
    int dt = 0;
    if(IsEmpty()) {
        System.out.println(x:"Queue masih kosong");
        System.exit(status:0);
```

2.2. Percobaan 2 : Antrian Layanan Akademik

```
package jobsheet10;

public class Mahasiswa04 {
    String nim;
    String nama;
    String prodi;
    String kelas;

    public Mahasiswa04(String nim, String nama, String prodi, String kelas) {
        this.nama = nama;
        this.nim = nim;
        this.prodi = prodi;
        thi
} s.kelas = kelas;
    }

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

```
package jobsheet10;

public class AntrianLayanan04 {
    Mahasiswa04[] data;
    int front;
    int rear;
    int size;
    int max;

    public AntrianLayanan04(int max) {
        this.max = max;
        this.data = new Mahasiswa04[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 tambahAntrian(Mahasiswa04 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 Mahasiswa04 layaniMahasiswa04() {
        if(IsEmpty()) {
            System.out.println("Antrian kosong.");
            return null;
        }
        Mahasiswa04 mhs = data[front];
        front = (front + 1) % max;
        size--;
        return mhs;
    }

    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 int getJumlahAntrian(){
        return size;
    }
}
```

```
package jobsheet10;
import java.util.Scanner;
public class LayananAkademikSIAKAD04 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        AntrianLayanan04 antrian = new AntrianLayanan04(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 = 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();
                    Mahasiswa04 mhs = new Mahasiswa04(nim, nama, prodi, kelas);
                    antrian.tambahAntrian(mhs);
                    break;
                case 2:
                    Mahasiswa04 dilayani = antrian.layaniMahasiswa04();
                    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();
    }
}
```

Outputnya:

```
C:\Praktikum ASD> cmd /C ""C:\Program Files\J
ser\workspaceStorage\9ab0f952d31733a7c7e6cca3

=== 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 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 LayananAkademikSIAKAD sehingga method LihatAkhir dapat dipanggil!

```
public class AntrianLayanan04 {
    public void tampilkanSemua(){
            data[index].tampilkanData();
        }
    }

    public int getJumlahAntrian(){
        return size;
    }

    public void lihatAkhir(){
        if(IsEmpty()) {
            System.out.println(x:"Antrian kosong.");
        }else {
            System.out.println(x:"Mahasiswa dalam Antrian Terakhir");
            System.out.println(x:"NIM - NAMA - PRODI - KELAS");
            data[rear].tampilkanData();
        }
    }
}
```

```
package jobsheet10;
import java.util.Scanner;
public class LayananAkademikSIAKAD04 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        AntrianLayanan04 antrian = new AntrianLayanan04(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("6. Antrian Paling Belakang");
            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();
                    Mahasiswa04 mhs = new Mahasiswa04(nim, nama, prodi, kelas);
                    antrian.tambahAntrian(mhs);
                    break;
                case 2:
                    Mahasiswa04 dilayani = antrian.layaniMahasiswa04();
                    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 6:
                    antrian.lihatAkhir();
                    break;
                case 0:
                    System.out.println("Terima Kasih.");
                    break;
                default:
                    System.out.println("Pilihan tidak valid.");
            }
        }while (pilihan != 0);
        sc.close();
    }
}
```

LATIHAN

Buatlah program antrian untuk mengilustasikan antrian persetujuan Kartu Rencana Studi (KRS) Mahasiswa oleh Dosen Pembina Akademik (DPA). Ketika seorang mahasiswa akan mengantri, maka dia harus mendaftarkan datanya (data mahasiswa seperti pada praktikum 2). Gunakan class untuk antrian seperti pada Praktikum 1 dan 2, dengan method-method yang berfungsi :

- Cek antrian kosong, Cek antrian penuh, Mengosongkan antrian.
- Menambahkan antrian, Memanggil antrian untuk proses KRS – setiap 1x panggilan terdiri dari 2 mahasiswa (pada antrian no 1 dan 2)
- Menampilkan semua antrian, Menampilkan 2 antrian terdepan, Menampilkan antrian paling akhir.
- Cetak jumlah antrian, Cetak jumlah yang sudah melakukan proses KRS
- Jumlah antrian maximal 10, jumlah yang ditangani masing-masing DPA 30 mahasiswa, cetak jumlah mahasiswa yang belum melakukan proses KRS.

Gambarkan Diagram Class untuk antriannya. Implementasikan semua method menggunakan menu pilihan pada fungsi main.

Mahasiswa
nim : String nama : String prodi : String kelas : String
Mahasiswa(nim : String, nama : String, prodi : String, kelas : String) void tampilkanData()

AntrianKRS
data : Mahasiswa[] front : int rear : int size : int max : int
AntrianKRS(int) isFull() : boolean isEmpty() : boolean tambahanAntrian(Mahasiswa) layaniMahasiswa() : Mahasiswa[] tampilkanSemua() lihatTerdepan() lihatAkhir() getJumlahAntrian() : int getJumlahProsesKRS() : int clear()

```
package jobsheet10;

public class Mahasiswa04 {
    String nim;
    String nama;
    String prodi;
    String kelas;

    public Mahasiswa04(String nim, String nama, String prodi, String kelas) {
        this.nama = nama;
        this.nim = nim;
        this.prodi = prodi;
        this.kelas = kelas;
    }

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

}
```

```
package jobsheet10;

public class AntrianKRS04 {
    Mahasiswa04[]  data;
    int front;
    int rear;
    int size;
    int max;

    public AntrianKRS04(int max) {
        this.max = max;
        this.data = new Mahasiswa04[max];
        this.size = 0;
        this.front = 0;
        this.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 tambahAntrian(Mahasiswa04 mhs) {
    if (IsFull()) {
        System.out.println("Antrian penuh, tidak dapat menambahkan mahasiswa.");
        return;
    }
    rear = (rear + 1) % max;
    data [rear] = mhs;
    size++;
    System.out.println(mhs.nama + " berhasil masuk ke antrian.");
}

public Mahasiswa04[] layaniMahasiswa04() {
    System.out.println("Daftar Mahasiswa dalam Antrian: ");
    System.out.println("NIM - NAMA - PRODI - KELAS");

    if (IsEmpty()) {
        System.out.println("Antrian Kosong.");
        return null;
    }
    Mahasiswa04[] mhsDipanggil = new Mahasiswa04[2];
    for (int i = 0; i < 2; i++){
        if (size > 0) {
            mhsDipanggil[i] = data[front];
            front = (front + 1) % max;
            size--;
        }else {
            mhsDipanggil[i] = null;
        }
    }
    return mhsDipanggil;
}

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.println((i+1) + ". ");
        data[index].tampilkanData();
    }
}

public void lihatTerdepan() {
    if(IsEmpty()) {
        System.out.println("Antrian kosong.");
    }else {
        System.out.println("Mahasiswa terdepan: ");
        System.out.println("NIM - NAMA - PRODI - KELAS");
        data[front].tampilkanData();
    }
}

public void lihatAkhir() {
    if (IsEmpty()) {
        System.out.println("Antrian kosong.");
    }else {
        System.out.println("Mahasiswa dalam Antrian Terakhir");
        System.out.println("NIM - NAMA - PRODI - KELAS");
        data[rear].tampilkanData();
    }
}
```

```
    public int getJumlahAntrian() {
        return size;
    }

    public int getJumlahProsesKRD() {
        return 30 - size;
    }

    public void clear() {
        front = rear = -1;
        size = 0;
        System.out.println("Antrian berhasil dikosongkan.");
    }
}
```

```
package jobsheet10;
import java.util.Scanner;
public class LayananKRS04 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        AntrianKRS04 antrian = new AntrianKRS04(10);
        int pilihan;

        do{
            System.out.println("\n=== Menu Antrian KRS ===");
            System.out.println("1. Tambah Mahasiswa ke Antrian");
            System.out.println("2. Proses KRS (2 Mahasiswa)");
            System.out.println("3. Tampilkan Semua Antrian");
            System.out.println("4. Tampilkan 2 Antrian Terdepan");
            System.out.println("5. Tampilkan Antrian Paling Akhir");
            System.out.println("6. Jumlah Antrian");
            System.out.println("7. Jumlah Mahasiswa yang Belum Proses KRS");
            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();
                    Mahasiswa04 mhs = new Mahasiswa04(nim, nama, prodi, kelas);
                    antrian.tambahAntrian(mhs);
                    break;
                case 2:
                    System.out.println("Mahasiswa yang dipanggil untuk proses KRS:");
                    Mahasiswa04[] dipanggil = antrian.layaniMahasiswa04();
                    for (Mahasiswa04 mahasiswa : dipanggil) {
                    if(mahasiswa != null) {
                        mahasiswa.tampilkanData();
                        }
                    }
                    break;
                case 3:
                    antrian.tampilkanSemua();
                    break;
```

```
                case 4:
                    antrian.lihatAkhir();
                    break;
                case 5:
                    antrian.lihatAkhir();
                    break;
                case 6:
                    System.out.println("Jumlah antrian: " + antrian.getJumlahAntrian());
                    break;
                case 7:
                    System.out.println("Jumlah mahasiswa yang belum melakukan proses KRS: " +
antrian.getJumlahAntrian());
                    break;
                case 0:
                    System.out.println("Terima kasih");
                    break;
                default:
                    System.out.println("Pilihan tidak valid.");
            }
        }while (pilihan != 0);
        sc.close();
    }
}
```

Outputnya:

```
C:\Praktikum ASD> cmd /C ""C:\Program Files\Java\jd
ser\workspaceStorage\9ab0f952d31733a7c7e6cca3460806

=== Menu Antrian KRS ===
1. Tambah Mahasiswa ke Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Jumlah Antrian
7. Jumlah Mahasiswa yang Belum Proses KRS
0. Keluar
Pilih Menu: 1
NIM : 123
Nama : Arifah
Prodi : TI
Kelas : 1E
Arifah berhasil masuk ke antrian.
```

```
=== Menu Antrian KRS ===
1. Tambah Mahasiswa ke Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Jumlah Antrian
7. Jumlah Mahasiswa yang Belum Proses KRS
0. Keluar
Pilih Menu: 1
NIM : 124
Nama : Delan
Prodi : TI
Kelas : 1E
Delan berhasil masuk ke antrian.

=== Menu Antrian KRS ===
1. Tambah Mahasiswa ke Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Jumlah Antrian
7. Jumlah Mahasiswa yang Belum Proses KRS
0. Keluar
Pilih Menu: 2
Mahasiswa yang dipanggil untuk proses KRS:
Daftar Mahasiswa dalam Antrian:
NIM - NAMA - PRODI - KELAS
123 - Arifah - TI - 1E
124 - Delan - TI - 1E
```

```
=== Menu Antrian KRS ===
1. Tambah Mahasiswa ke Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Jumlah Antrian
7. Jumlah Mahasiswa yang Belum Proses KRS
0. Keluar
Pilih Menu: 3
Antrian kosong.

=== Menu Antrian KRS ===
1. Tambah Mahasiswa ke Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Jumlah Antrian
7. Jumlah Mahasiswa yang Belum Proses KRS
0. Keluar
Pilih Menu: 4
Antrian kosong.
```

```
=== Menu Antrian KRS ===
1. Tambah Mahasiswa ke Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Jumlah Antrian
7. Jumlah Mahasiswa yang Belum Proses KRS
0. Keluar
Pilih Menu: 5
Antrian kosong.

=== Menu Antrian KRS ===
1. Tambah Mahasiswa ke Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Jumlah Antrian
7. Jumlah Mahasiswa yang Belum Proses KRS
0. Keluar
Pilih Menu: 6
Jumlah antrian: 0
```

```
=== Menu Antrian KRS ===
1. Tambah Mahasiswa ke Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Jumlah Antrian
7. Jumlah Mahasiswa yang Belum Proses KRS
0. Keluar
Pilih Menu: 7
Jumlah mahasiswa yang belum melakukan proses KRS: 0

=== Menu Antrian KRS ===
1. Tambah Mahasiswa ke Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Jumlah Antrian
7. Jumlah Mahasiswa yang Belum Proses KRS
0. Keluar
Pilih Menu: 0
Terima kasih
```