Laporan Hasil Praktikum
Algoritma dan Struktur Data
Jobsheet 10 Queue

Muhammad Firman Aditiasmara

244107020094

TI-1E

Program Studi Teknik Informatika

Jurusan Teknologi Informasi

Politeknik Negeri Malang

2025

1. Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Membuat dan mendeklarasikan struktur data Queue
2. Membuat dan mendeklarasikan struktur data Queue
3. Menerapkan algoritma Queue dengan menggunakan array

2. Praktikum

2.1 Percobaan 1: Operasi Dasar Queue

2.1.1 Langkah-langkah Percobaan

1. Membuat class Queue18 dan menambahkan atribut-atribut Queue sesuai diagram class, kemudian menambahkan pula konstruktornya

```
public class Queue18 {

    int[] data;
    int front;
    int rear;
    int size;
    int max;

    public Queue18(int n) {
        max = n;
        data = new int[max];
        size = 0;
        front = rear = -1;
    }
```

2. Membuat method IsEmpty bertipe boolean yang digunakan untuk mengecek apakah queue kosong

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

3. MemBuat method IsFull bertipe boolean yang digunakan untuk mengecek apakah queue sudah penuh.

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

4. Membuat 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");
        }
    }
```

5. Membuat 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);
        }
    }
```

6. Membuat 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");
        }
    }
```

7. Membuat 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++;
        }
    }
```

8. 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;
    }
```

9. Membuat class baru dengan nama QueueMain tetap pada package Praktikum1. Class ini untuk menampilkan menu dan juga menu yang dipilih

```
import java.util.Scanner;

public class QueueMain18 {
    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();

        Queue18 Q = new Queue18(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  );
    }

}
```

2.1.2 Verifikasi Hasil Percobaan

```
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 :
   Nilai front dan rear diatur ke -1 untuk menandai bahwa antrian masih kosong. Sedangkan size = 0 karena belum ada data yang dimasukkan ke dalam antrian.

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

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

   Jawab :
   Kode tersebut digunakan saat akan memasukkan data jika rear sudah berada di ujung array maka rear akan berpindah di awal pada indeks 0 yang kosong,yang membuat posisi rear berada didepan front

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

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

Jawab :
Kode tersebut jika kondisi front berada pada indeks terakhir pada arrau, maka setelah data diambil, front akan berpindah ke indeks awal yaitu 0

4. Pada method print, mengapa pada proses perulangan variabel i tidak dimulai dari 0 (int i=0), melainkan int i=front?
Jawab :
Perulangan variabel i dimulai dari front karena letak fornt tidak selalu berada di indeks 0, sehingga i harus mengikuti dimana front itu berada

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

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

Jawab :
Kode ini berfungsi agar saat telah mencapai indek terakhir maka akan langsung kembali ke awal

6. Tunjukkan potongan kode program yang merupakan queue overflow!
Jawab :

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

Ini merupakan kode program untuk batas queue, jika sudah penuh maka tidak bisa diisi lagi

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!
Jawab :

- Modifikasi pada Queue

```
if (isFull()) {
            System.out.println("Queue sudah
penuh");
            return;
```

- Modifikasi pada dequeue

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

2.2 Percobaan 2: : Antrian Layanan Akademik

2.2.1 Langkah-langkah Percobaan

1. Menambahkan atribut-atribut Mahasiswa seperti pada Class Diagram, kemudian tambahkan pula konstruktor serta method tampilkanData

```
public class Mahasiswa18 {
    String nim;
    String nama;
    String prodi;
    String kelas;

    public Mahasiswa18(String nim, String nama,
String prodi, String kelas) {
        this.nim = nim;
        this.nama = nama;
        this.prodi = prodi;
        this.kelas = kelas;
    }
    public void tampiilkanData(){
        System.out.println(nim + " - " + nama + " -
" + prodi + " - " + kelas );
    }
}
```

2. 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 class AntrianLayanan18 {

    Mahasiswa18[] data;
    int front;
    int rear;
    int size;
    int max;

    public AntrianLayanan18 (int max) {
        this.max = max;
        this.data = new Mahasiswa18[max];
        this.front = 0;
        this.rear = -1;
        this.size = 0;
    }
 public void tambahAntrian(Mahasiswa18 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 Mahasiswa18 layaniMahasiswa() {
        if (IsEmpty()) {
            System.out.println("Antrian kosong.");
            return null;
        }
        Mahasiswa18 mhs = data[front];
        front = (front + 1) % max;
        size--;
        return mhs;
    }
```

3. Merubah 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.println("Mahasiswa terdepan: ");
            System.out.println("NIM - NAMA - PRODI -
KELAS");
            data[front].tampiilkanData();
        }
    }

    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].tampiilkanData();
        }
    }
```

4. Menambahkan dengan method getJumlahAntrian yaitu menampilkan nilai size

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

5. MEmbuat class LayananAkademikSIAKAD sebagai class main untuk menampilkan pilihan menu

```
import java.util.Scanner;

public class LayananAkademikSIAKAD18 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        AntrianLayanan18 antrian = new AntrianLayanan18 (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();
                    Mahasiswa18 mhs = new Mahasiswa18(nim, nama, prodi, kelas);
                    antrian.tambahAntrian(mhs);
                    break;
                case 2:
                    Mahasiswa18 dilayani = antrian.layaniMahasiswa();
                    if (dilayani != null) {
                        System.out.print("Melayani mahasiswa: ");
                        dilayani.tampiilkanData();
                    }
                    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();
    }
}
```

2.2.2 Verifikasi Hasil Percobaan

2.2.3

```
=== 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.
```

2.2.4 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!

- Modifikasi Penambahan method LihatAkhir

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

- Modifikasi penambahan menu 6

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

- Modifikasi pada switch-case

```
                case 4:
                    antrian.tampilkanSemua();
                    break;
                case 5:
                    System.out.println("Jumlah dalam
 antrian: " + antrian.getJumlahAntrian());
                    break;
                // Pertanyaan
                case 6:
                    antrian.LihatAkhir();
                    break;
```

Hasil Compile

```
=== 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: 3
Mahasiswa terdepan:
NIM - NAMA - PRODI - KELAS
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
6. Cek Antrian Paling Belakang
0. Keluar
Pilih Menu: 6
Mahasiswa terdepan:
NIM - NAMA - PRODI - KELAS
124 - Bobi - TI - 1G
```

2.3 Latihan Praktikum

Kode Program 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 tampil () {
        System.out.println(nim + " - " + nama + " - " + prodi + "
- " + kelas );
    }
}
```

- Kode Program AntrianKRS

```
public class AntrianKRS18 {

    Mahasiswa[] data;
    int front;
    int rear;
    int size;
    int max = 10;
    int proses;
    int ditangani = 30;
    public AntrianKRS18(int max) {
        this.max = max;
        this.data = new Mahasiswa[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 clear() {
        if (!IsEmpty()) {
            front = rear = -1;
            size = 0;
            System.out.println("Queue berhasil dikosongkan");
        } else {
            System.out.println("Queue masih kosong");
        }
    }

    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 void memanggilAtrian() {
        if (size < 2) {
            System.out.println("Antrian kurang dari 2. Minimal 2
mahasiswa untuk proses KRS.");
            return;
        }

        System.out.println("Memproses KRS untuk:");
        for (int i = 0; i < 2; i++) {
            Mahasiswa mhs = data[front];
            mhs.tampil();
            front = (front + 1) % max;
            size--;
            proses++;
        }
    }
    public void tampilSemua() {
        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].tampil();
            }
        }

    public void tampilDuaTerdepan() {
        if (size < 2) {
            System.out.println("Belum ada 2 antrian terdepan.");
            return;
        }

        System.out.println("2 antrian terdepan:");
        int i = front;
        for (int j = 0; j < 2; j++) {
            data[i].tampil();
            i = (i + 1) % max;
        }
    }

    public void tampilAkhir() {
        if (IsEmpty()) {
            System.out.println("Antrian kosong.");
            return;
        }

        System.out.print("Antrian paling akhir: ");
        data[rear].tampil();
    }

    public void cetakJumlahAntrian() {
        System.out.println("Jumlah mahasiswa dalam antrian: " + size +
" mahasiswa");
    }

    public void cetakJumlahDiproses() {
        System.out.println("Jumlah mahasiswa yang sudah proses KRS: "
+ proses + " mahasiswa");
    }

    public void cetakBelumDiproses() {
        int sisa = ditangani - proses;
        System.out.println("Mahasiswa yang belum melakukan proses KRS:
" + sisa + " mahasiswa");
    }

}
```

- Kode Program class LayananKRS

```
mport java.util.Scanner;

public class LayananKRS {
     public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        AntrianKRS18 antrian = new AntrianKRS18 (10);
        int pilih;

        do {
            System.out.println("\nMENU ANTRIAN KRS:");
            System.out.println("1. Tambah Antrian");
            System.out.println("2. Proses KRS (2 Mahasiswa)");
            System.out.println("3. Tampilkan Semua Antrian");
            System.out.println("4. Tampilkan 2 Terdepan");
            System.out.println("5. Tampilkan Antrian Terakhir");
            System.out.println("6. Cetak Jumlah Antrian");
            System.out.println("7. Cetak Jumlah Diproses");
            System.out.println("8. Cetak Mahasiswa Belum Diproses");
            System.out.println("9. Kosongkan Antrian");
            System.out.println("0. Keluar");
            System.out.print("Pilih: ");
            pilih = sc.nextInt();
            sc.nextLine();

            switch (pilih) {
                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.memanggilAtrian();
                    break;
                case 3:
                    antrian.tampilSemua();
                    break;
                case 4:
                    antrian.tampilDuaTerdepan();
                    break;
                case 5:
                    antrian.tampilAkhir();
                    break;
                case 6:
                    antrian.cetakJumlahAntrian();
                    break;
```

```
case 7:
                    antrian.cetakJumlahDiproses();
                    break;
                case 8:
                    antrian.cetakBelumDiproses();
                    break;
                case 9:
                    antrian.clear();
                    break;
                case 0:
                    System.out.println("Terima Kasih.");
                    break;
                default:
                    System.out.println("Pilihan tidak valid.");
            }
        } while (pilih != 0);
    }
}
```

Hasil Compile kode Program

```
MENU ANTRIAN KRS:
1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Antrian Terakhir
6. Cetak Jumlah Antrian
7. Cetak Jumlah Diproses
8. Cetak Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih: 1
NIM: 1234
Nama: Adid
Prodi: TI
Kelas: 1E
Adid berhasil masuk ke antrian.

MENU ANTRIAN KRS:
1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Antrian Terakhir
6. Cetak Jumlah Antrian
7. Cetak Jumlah Diproses
8. Cetak Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih: 1
NIM: 123456
Nama: Fara
Prodi: TI
Kelas: 1E
Fara berhasil masuk ke antrian.

MENU ANTRIAN KRS:
1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Antrian Terakhir
6. Cetak Jumlah Antrian
7. Cetak Jumlah Diproses
8. Cetak Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih: 3
Daftar Mahasiswa dalam Antrian:
NIM - NAMA - PRODI - KELAS
1. 1234 - Adid - TI - 1E
2. 123456 - Fara - TI - 1E
```

```
MENU ANTRIAN KRS:
1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Antrian Terakhir
6. Cetak Jumlah Antrian
7. Cetak Jumlah Diproses
8. Cetak Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih: 1
NIM: 122
Nama: Ucup
Prodi: TI
Kelas: 1E
Ucup berhasil masuk ke antrian.

MENU ANTRIAN KRS:
1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Antrian Terakhir
6. Cetak Jumlah Antrian
7. Cetak Jumlah Diproses
8. Cetak Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih: 2
Memproses KRS untuk:
1234 - Adid - TI - 1E
123456 - Fara - TI - 1E

MENU ANTRIAN KRS:
1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Antrian Terakhir
6. Cetak Jumlah Antrian
7. Cetak Jumlah Diproses
8. Cetak Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih: 3
Daftar Mahasiswa dalam Antrian:
NIM - NAMA - PRODI - KELAS
1. 122 - Ucup - TI - 1E
```

```
MENU ANTRIAN KRS:
1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Antrian Terakhir
6. Cetak Jumlah Antrian
7. Cetak Jumlah Diproses
8. Cetak Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih: 2
Antrian kurang dari 2. Minimal 2 mahasiswa untuk proses KRS.

MENU ANTRIAN KRS:
1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Antrian Terakhir
6. Cetak Jumlah Antrian
7. Cetak Jumlah Diproses
8. Cetak Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih: 7
Jumlah mahasiswa yang sudah proses KRS: 2 mahasiswa

MENU ANTRIAN KRS:
1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Antrian Terakhir
6. Cetak Jumlah Antrian
7. Cetak Jumlah Diproses
8. Cetak Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih: 8
Mahasiswa yang belum melakukan proses KRS: 28 mahasiswa
```

```
MENU ANTRIAN KRS:
1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Antrian Terakhir
6. Cetak Jumlah Antrian
7. Cetak Jumlah Diproses
8. Cetak Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih: 9
Queue berhasil dikosongkan

MENU ANTRIAN KRS:
1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Antrian Terakhir
6. Cetak Jumlah Antrian
7. Cetak Jumlah Diproses
8. Cetak Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih: 3
Antrian kosong.
```