LAPORAN HASIL PRAKTIKUM

ALGORITMA DAN STRUKTUR DATA

JOBSHEET 10 QUEUE

RANI MIFTAHUS SAÁDAH

244107020057

TI_1E

PROGRAM STUDI D_IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

PRAKTIKUM 2025

2.1 Percobaan 1 : Operasi Dasar Queue

Queue25.java

```
public class Queue25 {
    int[] data;
    int front;
    int rear;
    int size;
    int max;

    public Queue25(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 kodsong: ");
        }
    }

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

```
        } else {
            System.out.println("Queue masih kosong");
        }
    }

    public void Enqueue(int dt) {
        if (IsFull()) {
            System.out.println("Queue sudah penuh");
            System.exit(0);
        } 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");
            System.exit(0);
        } else {
            dt = data[front];
            size--;
            if (IsEmpty()) {
                front = rear = -1;
            } else {
                if (front == max - 1) {
                    front = 0;
                } else {
                    front++;
                }
            }
        }
        return dt;
    }

}
```

QueueMain25.java

```
import java.util.Scanner;
public class QueueMain25 {
    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();

        Queue25 Q = new Queue25(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);
    }
}
```

2.2 Verifikasi

```
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
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
--------------------------------
```

2.3 Pertanyaan

1. Pada konstruktor, mengapa nilai awal atribut front dan rear bernilai -1, sementara atribut size bernilai 0?
    - *Front dan rear bernilai -1 menunjukkan queue kososng secara posisi*
    - *Size = 0 menandakan tidak ada elemen dalam queue*

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

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

    - *Saat queue kosong front dan rear diinisialisasikan ke indeks 0 untuk mulai menyimpan data*
    - *Jika sudah diisi hanya rear yangn naik (elemen barun sudah ditambahkan ke belakang antrian)*

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

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

- *Temp menyimpan data yang akan dikeluarkan dari depan*
- *Elemen selanjutnya digeser ke depan agar front tetap di indeks 0*
- *Rear – karena ada satu elemen yang hilang di akhir*
- *Size – karena jumlah total elemen berkurang 1*

4. Pada method print, mengapa pada proses perulangan variabel i tidak dimulai Dari 0 (int i=0), melainkan int i=front?
   Karena elemen yang valid dalam queue dimulai dari front hingga rear dan tidak semua elemen di array data[] berisi data aktif.

5. Perhatikan kembali method print, jelaskan maksud dari potongan kode berikut!
   ```
   i = (i + 1) % max;
   ```
   Mencetak seluruh elemen queue dari front ke rear sesuai urutan masuk, data[i] adalah elemen pada indeks I di queue.

6. Tunjukkan potongan kode program yang merupakan queue overflow!
   ```
   if (IsFull()) {
       System.out.println(x:"Queue sudah penuh");
       System.exit(status:0);
   ```
   Kondisi saat queue sudah dan tidak bisa enqueue data baru.

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 overflolw dan queue underflow, program dihentikan!
   ```
   if (IsFull()) {
       System.out.println(x:"Queue sudah penuh");
       System.exit(status:0);
   ```
   ```
   if (IsEmpty()) {
       System.out.println(x:"Queue masih kosong");
       System.exit(status:0);
   ```
   System.exit(); digunakan untuk menghentikan eksekusi program secara paksa saat error

3.1 Percobaan 2 : Antrian Layanan Akademik

Mahasiswa25.java

```
public class Mahasiswa25 {
    String nim, nama, prodi, kelas;

    public Mahasiswa25(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);
    }
}
```

AntrianLayanan25.java

```
public class AntrianLayanan25 {
    Mahasiswa25[] data;
    int front, rear, size, max;

    public AntrianLayanan25(int max) {
        this.max = max;
        this.data = new Mahasiswa25[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(Mahasiswa25 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 Mahasiswa25 layaniMahasiswa25() {
            if (isEmpty()) {
                System.out.println("Antrian kosong!");
                return null;
            }
            Mahasiswa25 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;
        }
    }
```

LayananAkademikSIAKAD25.java

```
    import java.util.Scanner;

    public class LayananAkademikSIAKAD25 {
        public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
            AntrianLayanan25 antrian = new AntrianLayanan25(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();
                        Mahasiswa25 mhs = new Mahasiswa25(nim,
    nama, prodi, kelas);
                        antrian.tambahAntrian(mhs);
                        break;
                    case 2:
                        Mahasiswa25 keluar =
    antrian.layaniMahasiswa25();
                        System.out.println("Melayani
    Mahasiswa:");
                        keluar.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);
        }
    }
```

3.2 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: 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
0. Keluar
Pilih Menu: 1
NIM: 124
Nama: angga
Prodi: TI
Kelas: 1E
angga 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 - agus - TI 1E
2. 124 - angga - TI 1E

=== 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 - agus - TI 1E

=== 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 - angga - TI 1E
```

```
=== 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.
```

3.3 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!

AntrianLayanan25.java

```
public class AntrianLayanan25 {
    Mahasiswa25[] data;
    int front, rear, size, max;

    public AntrianLayanan25(int max) {
        this.max = max;
        this.data = new Mahasiswa25[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(Mahasiswa25 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 Mahasiswa25 layaniMahasiswa25() {
        if (isEmpty()) {
            System.out.println("Antrian kosong!");
            return null;
        }
        Mahasiswa25 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;
        }

        public void lihatAkhir() {
            if (!isEmpty()) {
                System.out.println("Antrian paling belakang:");
                data[rear].tampilkanData();
            } else {
                System.out.println("Antrian kosong!");
            }
        }
    }
```

LayananAkademikSIAKAD25.java

```
    import java.util.Scanner;

    public class LayananAkademikSIAKAD25 {
        public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
            AntrianLayanan25 antrian = new AntrianLayanan25(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. Lihat Antrian Akhir");
                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();
                        Mahasiswa25 mhs = new Mahasiswa25(nim,
    nama, prodi, kelas);
                        antrian.tambahAntrian(mhs);
                        break;
                    case 2:
                        Mahasiswa25 keluar =
    antrian.layaniMahasiswa25();
                        System.out.println("Melayani
    Mahasiswa:");
                        keluar.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);
        }
    }
```

```
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
6. Lihat Antrian Akhir
0. Keluar
Pilih Menu: 1
NIM: 1124
Nama: yogi
Prodi: TI
Kelas: 1E
yogi 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. Lihat Antrian Akhir
0. Keluar
Pilih Menu: 1
NIM: 1125
Nama: yuda
Prodi: TI
Kelas: 1E
yuda 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. Lihat Antrian Akhir
0. Keluar
Pilih Menu: 6
Antrian paling belakang:
1125 - yuda - TI 1E
```

4.1 TUGAS

Mahasiswa25.java

```
public class Mahasiswa25 {
    String nim, nama, prodi, kelas;

    public Mahasiswa25(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);
        }
    }
```

AntrianKRS25.java

```
    public class AntrianKRS25 {
        Mahasiswa25[] data;
        int front, rear, size, max;
        int totalDiproses;

        public AntrianKRS25(int max) {
            this.max = max;
            this.data = new Mahasiswa25[max];
            this.front = rear = -1;
            this.size = 0;
            this.totalDiproses = 0;
        }

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

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

        public void tambahAntrian25(Mahasiswa25 mhs) {
            if (isFull()) {
                System.out.println("Antrian penuh!");
                return;
            }
            if (isEmpty()) {
                front = 0;
            }
            rear = (rear + 1) % max;
            data[rear] = mhs;
            size++;
            System.out.println(mhs.nama + " berhasil masuk ke
    antrian.");
        }

        public Mahasiswa25[] layaniMahasiswa() {
            System.out.println("Memproses 2 Mahasiswa: ");
            if (isEmpty()) {
                System.out.println("Antrian Kosong.");
                return null;
            }

            Mahasiswa25[] mhsDipanggil = new Mahasiswa25[2];
            for (int i = 0; i < 2; i++) {
                if (size > 0) {
                    mhsDipanggil[i] = data[front];
                    front = (front + 1) % max;
                    size--;
                    totalDiproses++;
                } else {
                    mhsDipanggil[i] = null;
                }
            }
```

```
            return mhsDipanggil;
        }

        public void tampilkanSemua25() {
            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 lihatTerdepan25() {
            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 lihatAkhir25() {
            if (isEmpty()) {
                System.out.println("Antrian kosong.");
            } else {
                System.out.println("Mahasiswa terakhir dalam
    antrian:");
                System.out.println("NIM - NAMA - PRODI - KELAS");
                data[rear].tampilkanData();
            }
        }

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

        public void jumlahAntrian() {
            System.out.println("Jumlah antrian saat ini: " +
    size);
        }

        public void jumlahProses() {
            System.out.println("Jumlah mahasiswa sudah proses
    KRS: " + totalDiproses);
        }
    }
```

LayananAkademikKRS25.java

```
import java.util.Scanner;

public class LayananAkademikKRS25 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        AntrianKRS25 antrian = new AntrianKRS25(10);
        int pilih;

        do {
            System.out.println("\n===== MENU ANTRIAN KRS
=====");
            System.out.println("1. Tambah Antrian
Mahasiswa");
            System.out.println("2. Proses (2 Mahasiswa)");
            System.out.println("3. Lihat Semua Antrian");
            System.out.println("4. Lihat 1 Antrian
Terdepan");
            System.out.println("5. Lihat Antrian Terakhir");
            System.out.println("6. Jumlah Antrian Saat Ini");
            System.out.println("7. Jumlah Mahasiswa yang
Sudah KRS");
            System.out.println("8. Jumlah Mahasiswa Belum
KRS");
            System.out.println("9. Kosongkan Antrian");
            System.out.println("0. Keluar");
            System.out.print("Pilih menu: ");
            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();
                    Mahasiswa25 mhs = new Mahasiswa25(nim,
nama, prodi, kelas);
                    antrian.tambahAntrian25(mhs);
                    break;

                case 2:
                    Mahasiswa25[] dilayani =
antrian.layaniMahasiswa();
                    if (dilayani != null) {
                        for (Mahasiswa25 m : dilayani) {
                            if (m != null) {
                                System.out.print("Melayani
mahasiswa: ");
                                m.tampilkanData();
                            }
                        }
                    }
                    break;

                case 3:
                    antrian.tampilkanSemua25();
                    break;
```

```
                    case 4:
                        antrian.lihatTerdepan25();
                        break;

                    case 5:
                        antrian.lihatAkhir25();
                        break;

                    case 6:
                        antrian.jumlahAntrian();
                        break;

                    case 7:
                        antrian.jumlahProses();
                        break;

                    case 8:
                        System.out.println("Jumlah mahasiswa
belum KRS: " + (10 - antrian.size));
                        break;

                    case 9:
                        antrian.clear();
                        break;

                    case 0:
                        System.out.println("Program selesai.");
                        break;

                    default:
                        System.out.println("Pilihan tidak
valid.");
                }
            } while (pilih != 0);
        }
}
```

```
===== MENU ANTRIAN KRS =====
1. Tambah Antrian Mahasiswa
2. Proses (2 Mahasiswa)
3. Lihat Semua Antrian
4. Lihat 1 Antrian Terdepan
5. Lihat Antrian Terakhir
6. Jumlah Antrian Saat Ini
7. Jumlah Mahasiswa yang Sudah KRS
8. Jumlah Mahasiswa Belum KRS
9. Kosongkan Antrian
0. Keluar
Pilih menu: 1
NIM: 112
Nama: rani
Prodi: TI
Kelas: 1E
rani berhasil masuk ke antrian.

===== MENU ANTRIAN KRS =====
1. Tambah Antrian Mahasiswa
2. Proses (2 Mahasiswa)
3. Lihat Semua Antrian
4. Lihat 1 Antrian Terdepan
5. Lihat Antrian Terakhir
6. Jumlah Antrian Saat Ini
7. Jumlah Mahasiswa yang Sudah KRS
8. Jumlah Mahasiswa Belum KRS
9. Kosongkan Antrian
0. Keluar
Pilih menu: 1
NIM: 113
Nama: cimeng
Prodi: TI
Kelas: 1E
cimeng berhasil masuk ke antrian.
```

```
===== MENU ANTRIAN KRS =====
1. Tambah Antrian Mahasiswa
2. Proses (2 Mahasiswa)
3. Lihat Semua Antrian
4. Lihat 1 Antrian Terdepan
5. Lihat Antrian Terakhir
6. Jumlah Antrian Saat Ini
7. Jumlah Mahasiswa yang Sudah KRS
8. Jumlah Mahasiswa Belum KRS
9. Kosongkan Antrian
0. Keluar
Pilih menu: 1
NIM: 114
Nama: angel
Prodi: TI
Kelas: 1E
angel berhasil masuk ke antrian.

===== MENU ANTRIAN KRS =====
1. Tambah Antrian Mahasiswa
2. Proses (2 Mahasiswa)
3. Lihat Semua Antrian
4. Lihat 1 Antrian Terdepan
5. Lihat Antrian Terakhir
6. Jumlah Antrian Saat Ini
7. Jumlah Mahasiswa yang Sudah KRS
8. Jumlah Mahasiswa Belum KRS
9. Kosongkan Antrian
0. Keluar
Pilih menu: 2
Memproses 2 Mahasiswa:
Melayani mahasiswa: 112 - rani - TI 1E
Melayani mahasiswa: 113 - cimeng - TI 1E
```

```
===== MENU ANTRIAN KRS =====
1. Tambah Antrian Mahasiswa
2. Proses (2 Mahasiswa)
3. Lihat Semua Antrian
4. Lihat 1 Antrian Terdepan
5. Lihat Antrian Terakhir
6. Jumlah Antrian Saat Ini
7. Jumlah Mahasiswa yang Sudah KRS
8. Jumlah Mahasiswa Belum KRS
9. Kosongkan Antrian
0. Keluar
Pilih menu: 3
Daftar Mahasiswa dalam Antrian:
NIM - NAMA - PRODI - KELAS
1. 114 - angel - TI 1E

===== MENU ANTRIAN KRS =====
1. Tambah Antrian Mahasiswa
2. Proses (2 Mahasiswa)
3. Lihat Semua Antrian
4. Lihat 1 Antrian Terdepan
5. Lihat Antrian Terakhir
6. Jumlah Antrian Saat Ini
7. Jumlah Mahasiswa yang Sudah KRS
8. Jumlah Mahasiswa Belum KRS
9. Kosongkan Antrian
0. Keluar
Pilih menu: 4
Mahasiswa terdepan:
NIM - NAMA - PRODI - KELAS
114 - angel - TI 1E
```

```
===== MENU ANTRIAN KRS =====
1. Tambah Antrian Mahasiswa
2. Proses (2 Mahasiswa)
3. Lihat Semua Antrian
4. Lihat 1 Antrian Terdepan
5. Lihat Antrian Terakhir
6. Jumlah Antrian Saat Ini
7. Jumlah Mahasiswa yang Sudah KRS
8. Jumlah Mahasiswa Belum KRS
9. Kosongkan Antrian
0. Keluar
Pilih menu: 5
Mahasiswa terakhir dalam antrian:
NIM - NAMA - PRODI - KELAS
114 - angel - TI 1E

===== MENU ANTRIAN KRS =====
1. Tambah Antrian Mahasiswa
2. Proses (2 Mahasiswa)
3. Lihat Semua Antrian
4. Lihat 1 Antrian Terdepan
5. Lihat Antrian Terakhir
6. Jumlah Antrian Saat Ini
7. Jumlah Mahasiswa yang Sudah KRS
8. Jumlah Mahasiswa Belum KRS
9. Kosongkan Antrian
0. Keluar
Pilih menu: 6
Jumlah antrian saat ini: 1
```

```
Pilih menu: 7
Jumlah mahasiswa sudah proses KRS: 2

===== MENU ANTRIAN KRS =====
1. Tambah Antrian Mahasiswa
2. Proses (2 Mahasiswa)
3. Lihat Semua Antrian
4. Lihat 1 Antrian Terdepan
5. Lihat Antrian Terakhir
6. Jumlah Antrian Saat Ini
7. Jumlah Mahasiswa yang Sudah KRS
8. Jumlah Mahasiswa Belum KRS
9. Kosongkan Antrian
0. Keluar
Pilih menu: 8
Jumlah mahasiswa belum KRS: 9

===== MENU ANTRIAN KRS =====
1. Tambah Antrian Mahasiswa
2. Proses (2 Mahasiswa)
3. Lihat Semua Antrian
4. Lihat 1 Antrian Terdepan
5. Lihat Antrian Terakhir
6. Jumlah Antrian Saat Ini
7. Jumlah Mahasiswa yang Sudah KRS
8. Jumlah Mahasiswa Belum KRS
9. Kosongkan Antrian
0. Keluar
Pilih menu: 9
Antrian dikosongkan.
```