LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
JOBSHEET 10

Disusun Oleh :

Nama : Nawaf Azril Annaufal

Nim : 244107020047

Kelas : TI 1E

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLINEMA
2025

Percobaan 1:

a. Hasil penulisan kode program untuk class Queue21.java

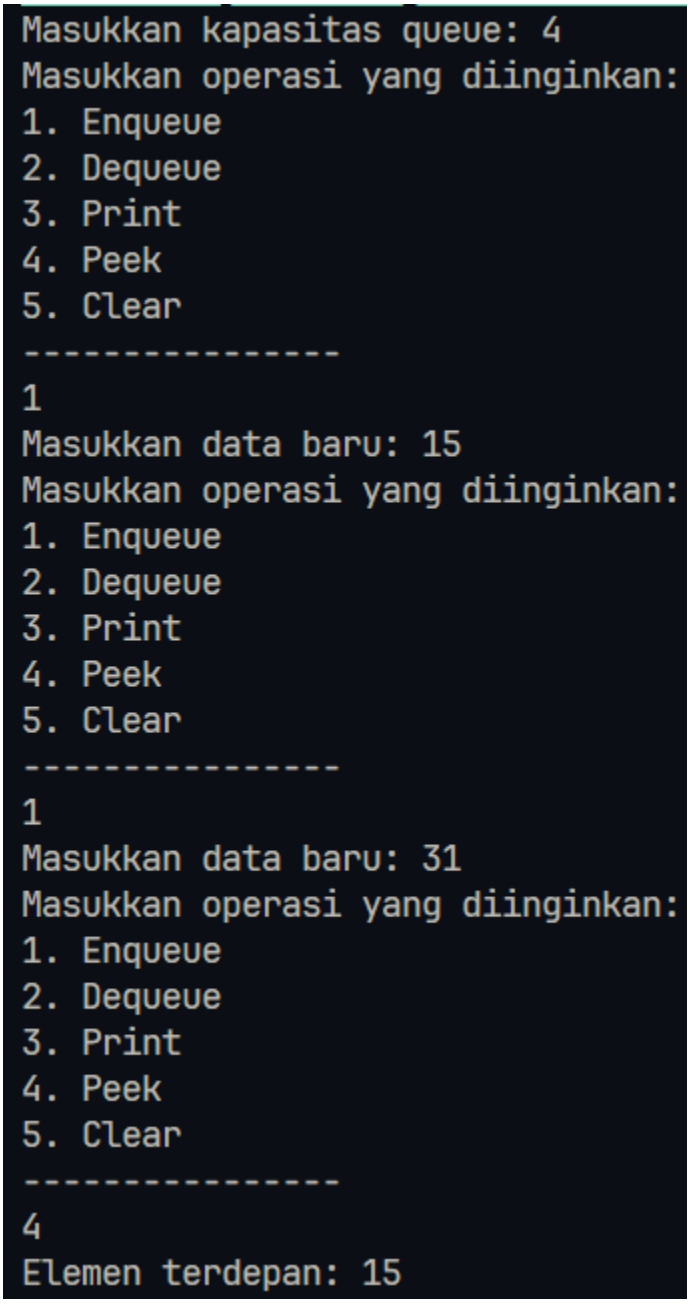
```
package Praktikum_ASD.P1Jobsheet10;

public class Queue21 {

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

    public Queue21(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.println(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");
        } 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;
    }
}
```

b. Hasil penulisan kode program untuk class QueueMain21.java

```
package Praktikum_ASD.P1Jobsheet10;

import java.util.Scanner;

public class QueueMain21 {

    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. Peek");
        System.out.println("4. Print");
        System.out.println("5. Clear");
        System.out.println("----------------");
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Masukkan kapasitas queue: ");
        int n = sc.nextInt();
        Queue21 Q = new Queue21(n);
        int pilih;

        do {
            menu();
            pilih = sc.nextInt();
            switch (pilih) {
                case 1:
                    System.out.println("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;

                default:
                    break;
            }

        } while(pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5);

        sc.close();
    }
}
```

c. Hasil run kode program



Pertanyaan:

1. Pada konstruktor, mengapa nilai awal atribut front dan rear bernilai -1, sementara atribut size bernilai 0?
2. Pada method **Enqueue**, jelaskan maksud dan kegunaan dari potongan kode berikut!

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

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

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

4. Pada method **print**, mengapa pada proses perulangan variabel i tidak dimulai dari 0 (**`int i=0`**), melainkan **`int i=front`**?
5. Perhatikan kembali method **print**, jelaskan maksud dari potongan kode berikut!

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

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

1. Karena dalam kondisi kosong atau awal run program, front tidak sedang menunjuk ke index manapun yang artinya tidak ada data didalam Queue saat ini, sedangkan size mengindikasikan jumlah data di dalam Queue yang berdasarkan max yang sudah ditentukan, misalkan max di set 5 maka data kosongnya adalah 0.

2. Pada statement tersebut menunjukkan bahwa jika rear sama dengan max – 1 yang artinya rear berada di posisi paling belakang Queue, maka rear akan di set menjadi 0 untuk memasukkan data yang baru.

3. Pada statement tersebut menunjukkan bahwa jika front sama dengan max -1 yang artinya front berada di posisi paling akhir Queue, maka front akan di set menjadi 0 atau front akan menunjuk index ke 0 sebagai front yang baru, maka index max – 1 tidak ditunjuk lagi (terhapus).

4. Karena dalam konsep Queue Menggunakan FIFO jadi mau itu terhapus atau dicetak perlakuan pertama akan tetap pada front.

5. Statement itu digunakan untuk mengiterasi indeks array secara melingkar, yang artinya jika sudah berada di posisi (max – 1) maka indeks berikutnya akan kembali ke 0 karena max % max = 0, agar tidak melampaui batas array yang sudah ditentukan.

6. Ini merupakan potongan kode yang mengarah pada peringatan Queueoverflow

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

7. Hasil modifikasi kode program

```
case 1:
      System.out.print("Masukkan data baru: ");
      int dataMasuk = sc.nextInt();
      if(Q.isFull()) {
            System.out.println("Queue sudah penuh, Program dihentikan");
            stop = true;
      } else {
            Q.enqueue(dataMasuk);
      }
      break;
```

```
} while(!stop);
```

Set Pada fungsi enque dimana terdapat pengecekan jika isfull = true maka akan langsung return ke main, setelah itu pada main sistem menambahkan varible baru yaitu boolean stop = false, kemudian pada case 1 dilakukan pengecekan dimana jika Q.isFull = true maka set stop = true yang akan menyebabkan statemen while menjadi false dan akan menghentikan program.

```
case 2:
      int dataKeluar = Q.dequeue();
      if(Q.isEmpty()) {
            System.out.println("Queue kosong, Program dihentikan");
            stop = true;
      } else if (dataKeluar != 0) {
            System.out.println("Data yang dikeluarkan: " + dataKeluar);
            break;
      }
```

Sama halnya dengan case 2 mengecek apakah isEmpty = true jika iya maka akan set nilai stop = true yang akan mengehntikan program.

Percobaan 2:

a. Hasil penulisan kode program untuk class Mahasiswa21.java

```
package Praktikum_ASD.Jobsheet10.P2Jobsheet10;

public class Mahasiswa21 {

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

    public Mahasiswa21(String nim, String nama, String prodi, String kelas) {
        this.nim = nim;
        this.nama = nama;
        this.prodi = prodi;
        this.kelas = kelas;
    }

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

    }
}
```

b. Hasil penulisan kode program untuk class AntrianLayanan.java

```
package Praktikum_ASD.Jobsheet10.P2Jobsheet10;

public class AntrianLayanan {

    Mahasiswa21[] data;
    int front;
    int rear;
    int size;
    int max;

    public AntrianLayanan(int max) {
        this.max = max;
        data = new Mahasiswa21[max];
        front = 0;
        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 lihatTerdepan() {
        if (isEmpty()) {
            System.out.println("Antrian kosong");
        } else {
            System.out.println("Mahasiswa terdepan: ");
            System.out.println("NIM - NAMA - PRODI - KELAS");
            data[front].tampilData();
        }
    }
```

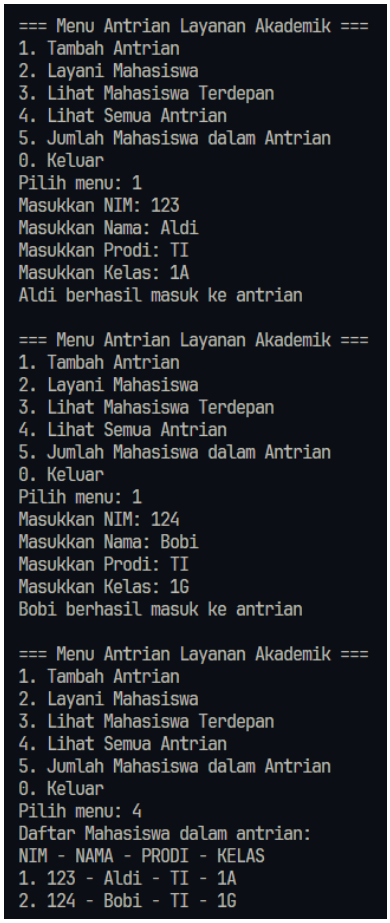
```
public void tampilkanSemua() {
        if(isEmpty()) {
            System.out.println("Antrian kosong");
            return;
        }
        System.out.println("Daftar Mahasiswa dalam antrian:");
        System.out.println("NIM - NAMA - PRODI - KELAS");
        for (int i = 0; i < size; i++) {
            int index = (front + i) % max;
            System.out.print((i + 1) + ". ");
            data[index].tampilData();

        }

    }

    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(Mahasiswa21 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 Mahasiswa21 LayaniMahasiswa() {
        if(isEmpty()) {
            System.out.println("Antrian kosong");
            return null;
        }
        Mahasiswa21 mhs = data[front];
        front = (front + 1) % max;
        size--;
```

c. Hasil penulisan kode program untuk class LayananAkademikSIAKAD.java

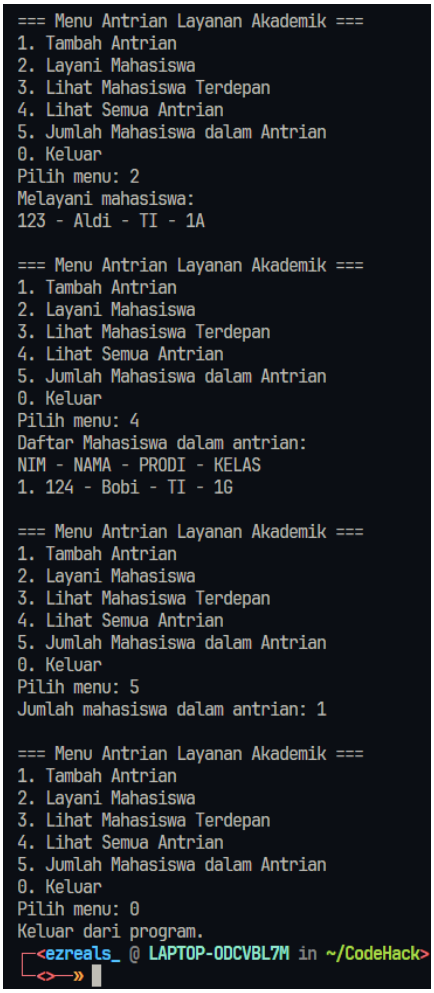
```
package Praktikum_ASD.Jobsheet10.P2Jobsheet10;

import java.util.Scanner;

public class LayananAkademikSIAKAD {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        AntrianLayanan antrian = new AntrianLayanan(5);
        int pilihan;

        do {
            System.out.println("\n=== Menu Antrian Layanan Akademik ===");
            System.out.println("1. Tambah 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.println("Pilih menu: ");
            pilihan = sc.nextInt();
            sc.nextLine();

            switch (pilihan) {
                case 1:
                    if (!antrian.isFull()) {
                        System.out.print("Masukkan NIM: ");
                        String nim = sc.nextLine();
                        System.out.print("Masukkan Nama: ");
                        String nama = sc.nextLine();
                        System.out.print("Masukkan Prodi: ");
                        String prodi = sc.nextLine();
                        System.out.print("Masukkan Kelas: ");
                        String kelas = sc.nextLine();
                        Mahasiswa21 mhs = new Mahasiswa21(nim, nama, prodi, kelas);
                        antrian.tambahAntrian(mhs);
                    } else {
                        System.out.println("Antrian sudah penuh!");
                    }
                    break;
```

```
 case 2:
                    Mahasiswa21 dilayani = antrian.LayaniMahasiswa();
                    if(dilayani != null) {
                        System.out.println("Melayani mahasiswa:");
                        dilayani.tampilData();
                    }
                    break;
                case 3:
                    antrian.lihatTerdepan();
                    break;
                case 4:
                    antrian.tampilkanSemua();
                    break;
                case 5:
                    System.out.println("Jumlah mahasiswa dalam antrian: " +
antrian.getJumlahAntrian());
                    break;
                case 0:
                    System.out.println("Keluar dari program.");
                    break;
                default:
                    System.out.println("Pilihan tidak valid!");
            }
        } while(pilihan != 0);
        sc.close();
    }
}
```

d. Hasil run kode program



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

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

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

=== Menu Antrian Layanan Akademik ===
1. Tambah Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
0. Keluar
Pilih menu: 0
Keluar dari program.
┌─<ezreals_ @ LAPTOP-0DCVBL7M in ~/CodeHack>
└─<>──» 
```

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!

Jawab:

Hasil modifikasi pada class AntrianLayanan.java

```
public void lihatAkhir() {
        if (isEmpty()) {
           System.out.println("Antrian kosong");
       } else {
           System.out.println("Mahasiswa paling belakang: ");
           System.out.println("NIM - NAMA - PRODI - KELAS");
           data[rear].tampilData();
       }
   }
```

Hasil modifikasi pada class LayananAkademikSIAKAD.java

```
System.out.println("\n=== Menu Antrian Layanan Akademik ===");
           System.out.println("1. Tambah 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: ");
```

```
case 6:
    antrian.lihatAkhir();
```

Tugas:

Hasil penulisan kode program untuk class Mahasiswa21.java

```
package Praktikum_ASD.Jobsheet10.TugasQueue;

public class Mahasiswa21 {

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

    public Mahasiswa21(String nim, String nama, String prodi, String kelas ) {
        this.nim = nim;
        this.nama = nama;
        this.prodi = prodi;
        this.kelas = kelas;
    }

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

Hasil penulisan kode program AntrianKRS21.java

```
package Praktikum_ASD.Jobsheet10.TugasQueue;
public class AntrianKRS21 {
    Mahasiswa21[] data;
    int front, rear, size, max;
    int totalDiproses = 0;

    public AntrianKRS21(int max) {
        this.max = max;
        data = new Mahasiswa21[max];
        front = 0;
        rear = -1;
        size = 0;
    }

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

    public boolean isFull() {
        return size == max;
    }
    public void clear() {
        front = 0;
        rear = -1;
        size = 0;
        totalDiproses = 0;
        System.out.println("Antrian dikosongkan.");
    }
    public void tambahAntrian(Mahasiswa21 mhs) {
        if (isFull()) {
            System.out.println("Antrian penuh, tidak bisa menambah mahasiswa.");
            return;
        }
        rear = (rear + 1) % max;
        data[rear] = mhs;
        size++;
        System.out.println(mhs.nama + " berhasil ditambahkan ke antrian.");
    }
    public void prosesKRS() {
        if (size < 2) {
            System.out.println("Antrian kurang dari 2 mahasiswa. Tidak bisa proses.");
            return;
        }
        System.out.println("Memproses 2 mahasiswa:");
        for (int i = 0; i < 2; i++) {
            Mahasiswa21 mhs = data[front];
            mhs.tampilData();
            front = (front + 1) % max;
            size--;
            totalDiproses++;
        }
    }
```

```
    public void tampilkanSemua() {
        if (isEmpty()) {
            System.out.println("Antrian kosong.");
            return;
        }
        System.out.println("Daftar Mahasiswa dalam Antrian:");
        for (int i = 0; i < size; i++) {
            int index = (front + i) % max;
            System.out.print((i + 1) + ". ");
            data[index].tampilData();
        }
    }

    public void lihatDuaTerdepan() {
        if (size < 2) {
            System.out.println("Kurang dari dua mahasiswa dalam antrian.");
            return;
        }
        System.out.println("2 Mahasiswa Terdepan:");
        for (int i = 0; i < 2; i++) {
            int index = (front + i) % max;
            data[index].tampilData();
        }
    }

    public void lihatAkhir() {
        if (isEmpty()) {
            System.out.println("Antrian kosong.");
        } else {
            System.out.println("Mahasiswa terakhir dalam antrian:");
            data[rear].tampilData();
        }
    }

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

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

    public void jumlahBelumDiproses() {
        int sisa = 30 - totalDiproses;
        System.out.println("Sisa mahasiswa yang belum melakukan proses KRS oleh DPA: "
+ (sisa > 0 ? sisa : 0));
    }
}
```

Hasil penulisan kode program LayananKRS21.java

```
package Praktikum_ASD.Jobsheet10.TugasQueue;
import java.util.Scanner;

public class LayananKRS21 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        AntrianKRS21 antrian = new AntrianKRS21(10);
        int pilihan;

        do {
            System.out.println("\n=== Menu Antrian KRS Mahasiswa ===");
            System.out.println("1. Tambah Antrian");
            System.out.println("2. Proses 2 Mahasiswa");
            System.out.println("3. Lihat 2 Terdepan");
            System.out.println("4. Lihat Semua Antrian");
            System.out.println("5. Lihat Antrian Terakhir");
            System.out.println("6. Cetak Jumlah Antrian");
            System.out.println("7. Cetak Jumlah yang Sudah Proses KRS");
            System.out.println("8. Lihat Sisa Mahasiswa yang Belum Diproses");
            System.out.println("9. Kosongkan Antrian");
            System.out.println("0. Keluar");
            System.out.print("Pilih menu: ");
            pilihan = sc.nextInt();
            sc.nextLine();

            switch (pilihan) {
                case 1:
                    if (!antrian.isFull()) {
                        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();
                        Mahasiswa21 mhs = new Mahasiswa21(nim, nama, prodi, kelas);
                        antrian.tambahAntrian(mhs);
                    } else {
                        System.out.println("Antrian penuh!");
                    }
                    break;
                case 2:
                    antrian.prosesKRS();
                    break;
                case 3:
                    antrian.lihatDuaTerdepan();
                    break;
                case 4:
                    antrian.tampilkanSemua();
                    break;
```

```
 case 5:
                    antrian.lihatAkhir();
                    break;
                case 6:
                    antrian.jumlahAntrian();
                    break;
                case 7:
                    antrian.jumlahDiproses();
                    break;
                case 8:
                    antrian.jumlahBelumDiproses();
                    break;
                case 9:
                    antrian.clear();
                    break;
                case 0:
                    System.out.println("Terima kasih!");
                    break;
                default:
                    System.out.println("Pilihan tidak valid.");
            }
        } while (pilihan != 0);

        sc.close();
    }
}
```

Diagram class Antrian