

**LAPORAN HASIL PRAKTIKUM ALGORITMA DAN STRUKTUR
DATA**

Jobsheet 10



Adi Luhung

244107020088

Kelas 1E

Program Studi Teknik Informatika

Jurusan Teknologi Informasi

Politeknik Negeri Malang

2025

1. Praktikum

1.1 Percobaan 1

1. Membuat class dengan nama Queue.java, yang memiliki beberapa method untuk algoritma Queue

```
package P1Jobsheet10;
public class Queue {
    int[] data;
    int front;
    int rear;
    int size;
    int max;

    public Queue (int n) {
        max = n;
        data = new int[max];
        size = 0;
        front = rear = -1;
    }
    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 clear() {
    if (!IsEmpty()) {
        front = rear = -1;
        size = 0;
        System.out.println("Queue berhasil dikosong");
    } 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;
}
}

```

2. Membuat class main untuk menampilkan pilhan menu dan memanggil fungsi yang sudah dibuat di class Queue.java

```
package P1Jobsheet10;
import java.util.Scanner;
public class QueueMain {
    public static void menu() {
        System.out.println("Masukkan operasi yang diinginkan: ");
        System.out.println("1. Enqueue");
        System.out.println("2. Dequeue");
        System.out.println("3. Print");
        System.out.println("4. Peek");
        System.out.println("5. Clear");
        System.out.println("-----");
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Masukkan kapasitas queue: ");
        int n = sc.nextInt();
        Queue Q = new Queue(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);
    }
}
```

3. Hasil running kode program

```
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
-----
█
```

1.1.1 Pertanyaan

1. Karena, front dan rear menunjuk pada indeks. Sehingga jika Queue masih belum terisi, maka front dan rear bernilai -1 karena indek dimulai dari angka 0. Sedangkan size pada awalnya bernilai 0 karena menunjukan ukuran atau panjang dari Queue yang dimulai dari angka 1.

2. Pada method Enqueue

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

Maksud dari kode tersebut adalah saat ingin menambahkan data baru yang belum penuh, tetapi rear berada di indeks terakhir, maka rear akan kembali ke indeks 0 agar data dapat ditambahkan.

3. Pada method Dequeue

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

Kode tersebut bertujuan agar saat ingin menghapus data dan front berada pada indeks terakhir, maka front akan dipindah ke indeks 0. Sehingga front tidak berhenti di indeks terakhir.

4. Pada method print

```
int i = front;
```

Variabel i bernilai front karena front tidak selalu berada di indkes ke-0, jika program pernah menjalankan menu Dequeue, maka front akan pindah ke indeks setelahnya.

5. Pada method print

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

Kode tersebut berfungsi agar nilai pada variabel i kembali ke 0 saat berada di indeks terakhir. Sehingga semua data dapat ditampilkan.

6. Pada method Enqueue

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

Kode tersebut menunjukkan Queue overflow.

7. Memodifikasi agar kode program berhenti saat terjadi Queue Overflow dan Queue Underflow.

a. Pada method Enqueue

```
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++;  
    }  
}
```

b. Pada method Dequeue

```
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;
}
```

c. Hasil modifikasi kode program

```
Masukkan kapasitas queue: 2
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
-----
1
Masukkan data baru: 21
Queue sudah penuh
PS C:\Users\luhun\Kuliah1 Semester2\Praktikum_ASD>
```

```
Masukkan kapasitas queue: 1
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
1
Masukkan data baru: 32
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
2
Data yang dikeluarkan: 32
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
2
Queue masih kosong
PS C:\Users\luhun\Kuliah1 Semester2\Praktikum_ASD>
```

1.2 Percobaan 2

1. Membuat class baru dengan nama Mahasiswa yang berisi fungsi dan atributnya.

```
package P2Jobsheet10;

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

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

2. Membuat class AntrianLayanan yang berisi method-method untuk algoritma Queue

```
package P2Jobsheet10;
public class AntrianLayanan {
    Mahasiswa[] data;
    int front;
    int rear;
    int size;
    int max;

    public AntrianLayanan (int max) {
        this.max = max;
        this.data = new Mahasiswa[max];
        this.front = 0;
        this.rear = -1;
        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 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 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 clear() {
    if (!IsEmpty()) {
        front = rear = -1;
        size = 0;
        System.out.println("Queue berhasil dikosong");
    } 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 Mahasiswa layaniMahasiswa() {
    if (IsEmpty()) {
        System.out.println("Antrian kosong");
        return null;
    }
    Mahasiswa mhs = data[front];
    front = (front + 1) % max;
    size--;
    return mhs;
}

```

3. Membuat class main dengan nama LayananAkademikSiakad, yang berfungsi untuk menampilkan pilihan menu dan memanggil fungsi fungsinya.

```
package 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 Mahasiswa ke Antrian");
            System.out.println("2. Layani Mahasiswa");
            System.out.println("3. Lihat Mahasiswa Terdepan");
            System.out.println("4. Lihat Semua Antrian");
            System.out.println("5. Jumlah Mahasiswa dalam Antrian");
            System.out.println("0. Keluar");
            System.out.print("Pilih menu: ");
            pilihan = sc.nextInt();
            sc.nextLine();
            switch (pilihan) {
                case 1:
                    System.out.print("NIM      : ");
                    String nim = sc.nextLine();
                    System.out.print("Nama      : ");
                    String nama = sc.nextLine();
                    System.out.print("Prodi     : ");
                    String prodi = sc.nextLine();
                    System.out.print("Kelas    : ");
                    String kelas = sc.nextLine();
                    Mahasiswa mhs = new Mahasiswa(nim, nama, prodi, kelas);
                    antrian.tambahAntrian(mhs);
                    break;

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

                case 3:
                    antrian.lihatTerdepan();
                    break;
                case 4:
                    antrian.tampilkanSemua();
                    break;
                case 5:
                    System.out.println("Jumlah data antrian: " + antrian.getJumlahAntrian());
                    break;
                case 0:
                    System.out.println("Terima kasih.");
                    break;
                default:
                    System.out.println("Pilihan tidak valid.");
            }
        } while (pilihan != 0);
        sc.close();
    }
}
```

4. Hasil running kode program

```
=== 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 data 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.
```

1.2.1 Pertanyaan

1. Memodifikasi kode program agar dapat menampilkan antrian paling belakang
 - a. Menambahkan method lihatAkhir pada class AntrianLayanan

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

- b. Menambahkan pilihan menu baru pada classs main

```
package 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 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 Mahasiswa Paling 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();
                    Mahasiswa mhs = new Mahasiswa(nim, nama, prodi, kelas);
                    antrian.tambahAntrian(mhs);
                    break;

                case 2:
                    Mahasiswa dilayani = antrian.layaniMahasiswa();
                    if (dilayani != null) {
                        System.out.print("Melayani mahasiswa: ");
                        dilayani.tampilkanData();
                    }
                    break;
            }
        } while (pilihan != 0);
    }
}
```

```

        case 3:
            antrian.lihatTerdepan();
            break;
        case 4:
            antrian.tampilkanSemua();
            break;
        case 5:
            System.out.println("Jumlah data antrian: " + antrian.getJumlahAntrian());
            break;
        case 6:
            antrian.lihatAkhir();
        case 0:
            System.out.println("Terima kasih.");
            break;
        default:
            System.out.println("Pilihan tidak valid.");
    }
} while (pilihan != 0);
sc.close();
}
}

```

2. Hasil running kode program

```

=== 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 Mahasiswa Paling Akhir
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
6. Lihat Mahasiswa Paling Akhir
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
6. Lihat Mahasiswa Paling Akhir
0. Keluar
Pilih menu: 6
Mahasiswa paling akhir:
NIM - NAMA - PRODI - KELAS
124 - Bobi - TI - 1G
Terima kasih.

=== 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 Mahasiswa Paling Akhir
0. Keluar
Pilih menu: █

```

2. Tugas

1. Membuat class Mahasiswa yang berisi atribut, konstruktor, dan method untuk menampilkan data.

```
package TugasJobsheet10;
public class DataMahasiswa {
    String nim;
    String nama;
    String prodi;
    String kelas;

    public DataMahasiswa(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);
    }
}
```

2. Membuat class KrsQueue yang berisi atribut, konstruktor, dan method-method untuk algoritma Queue.

```
package TugasJobsheet10;
public class KrsQueue {
    DataMahasiswa[] data;
    int front;
    int rear;
    int size;
    int max;
    int jumlahMahasiswa;
    int sudahDiProses = 0;

    public KrsQueue (int max) {
        this.max = max;
        this.data = new DataMahasiswa[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 lihatTerdepan() {
    if (IsEmpty()) {
        System.out.println("Antrian kosong");
    } else {
        System.out.println("Mahasiswa terdepan: ");
        System.out.println("No. NIM - NAMA - PRODI - KELAS");
        System.out.print("1. ");
        data[front].tampilkanData();
        System.out.print("2. ");
        data[front + 1].tampilkanData();
    }
}

public void tampilkanSemua() {
    if (IsEmpty()) {
        System.out.println("Antrian kosong");
        return;
    }
    System.out.println("Daftar Mahasiswa dalam antrian: ");
    System.out.println("No. 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 clear() {
    if (!IsEmpty()) {
        front = rear = -1;
        size = 0;
        System.out.println("Queue berhasil dikosong");
    } else {
        System.out.println("Queue masih kosong");
    }
}

public void tambahAntrian(DataMahasiswa mhs) {
    if (IsFull() && jumlahMahasiswa <= 30 ) {
        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.");
    jumlahMahasiswa++;
}

public void lihatAkhir() {
    if (IsEmpty()) {
        System.out.println("Antrian kosong");
    } else {
        System.out.println("Mahasiswa paling akhir: ");
        System.out.println("No. NIM - NAMA - PRODI - KELAS");
        System.out.print("1. ");
        data[rear].tampilkanData();
    }
}
}

```

```

public DataMahasiswa[] layaniMahasiswa() {
    if(IsEmpty()) {
        System.out.println("Antrian kosong");
        return null;
    } else {
        if (size >= 2) {
            DataMahasiswa[] duaMhs = new DataMahasiswa[2];
            duaMhs[0] = data[front];
            duaMhs[1] = data[(front + 1) % max];
            front = (front + 2) % max;
            size -= 2;
            sudahDiProses += 2;
            return duaMhs;
        } else {
            DataMahasiswa[] satuMhs = new DataMahasiswa[1];
            satuMhs[0] = data[front];
            front = (front + 1) % max;
            size--;
            sudahDiProses++;
            return satuMhs;
        }
    }
}

public int SudahDiProses() {
    return sudahDiProses;
}
}

```

3. Membuat class KrsMain berfungsi untuk menampilkan menu dan memanggil method yang sudah dibuat di class KrsQueue.

```

package TugasJobsheet10;
import java.util.Scanner;
public class KrsMain {
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        KrsQueue antrian = new KrsQueue(10);
        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 Dua Mahasiswa Terdepan");
            System.out.println("4. Lihat Semua Antrian");
            System.out.println("5. Jumlah Mahasiswa dalam Antrian");
            System.out.println("6. Lihat Mahasiswa Paling Akhir");
            System.out.println("7. Jumlah Mahasiswa Sudah Diproses");
            System.out.println("8. Cek status 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();
        DataMahasiswa mhs = new DataMahasiswa(nim, nama, prodi, kelas);
        antrian.tambahAntrian(mhs);
        break;

    case 2:
        DataMahasiswa dilayani[] = antrian.layaniMahasiswa();
        System.out.println("Melayani mahasiswa: ");
        System.out.println("No. NIM - NAMA - PRODI - KELAS");
        if (dilayani != null) {
            for (int i = 0; i < dilayani.length; i++) {
                System.out.print((i + 1) + ". ");
                dilayani[i].tampilkanData();
            }
        }
        break;

    case 3:
        antrian.lihatTerdepan();
        break;

    case 4:
        antrian.tampilkanSemua();
        break;

    case 5:
        System.out.println("Jumlah data antrian: " + antrian.getJumlahAntrian());
        break;

    case 6:
        antrian.lihatAkhir();
        break;

    case 7:
        System.out.println("Jumlah data sudah diproses: " +
antrian.SudahDiProses());
        break;

    case 8:
        System.out.print("Status antrian: ");
        if (antrian.isEmpty()) {
            System.out.println("Antrian kosong.");
        } else if (antrian.IsFull()) {
            System.out.println("Antrian penuh.");
        } else {
            System.out.println("Terdapat antrian");
        }
        }

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

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

```

4. Hasil running kode program
a. Menu 1

```
=== Menu Antrian layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Dua Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
6. Lihat Mahasiswa Paling Akhir
7. Jumlah Mahasiswa Sudah Diproses
8. Cek status antrian
0. Keluar
Pilih menu: 1
NIM      : 123
Nama     : Aldi
Prodi    : TI
Kelas   : 1E
Aldi berhasil masuk ke antrian.
```

```
=== Menu Antrian layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Dua Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
6. Lihat Mahasiswa Paling Akhir
7. Jumlah Mahasiswa Sudah Diproses
8. Cek status antrian
0. Keluar
Pilih menu: 1
NIM      : 124
Nama     : Edi
Prodi    : TI
Kelas   : 1A
Edi berhasil masuk ke antrian.
```

```
=== Menu Antrian layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Dua Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
6. Lihat Mahasiswa Paling Akhir
7. Jumlah Mahasiswa Sudah Diproses
8. Cek status antrian
0. Keluar
Pilih menu: 1
NIM      : 125
Nama     : Budi
Prodi    : TI
Kelas   : 1B
Budi berhasil masuk ke antrian.
```

```
=== Menu Antrian layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Dua Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
6. Lihat Mahasiswa Paling Akhir
7. Jumlah Mahasiswa Sudah Diproses
8. Cek status antrian
0. Keluar
Pilih menu: 1
NIM      : 126
Nama     : Rosa
Prodi    : TI
Kelas   : 1C
Rosa berhasil masuk ke antrian.
```

```
=== Menu Antrian layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Dua Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
6. Lihat Mahasiswa Paling Akhir
7. Jumlah Mahasiswa Sudah Diproses
8. Cek status antrian
0. Keluar
Pilih menu: 1
NIM      : 127
Nama     : Ocad
Prodi    : TI
Kelas   : 1E
Ocad berhasil masuk ke antrian.
```

b. Menu 2

```
=== Menu Antrian layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Dua Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
6. Lihat Mahasiswa Paling Akhir
7. Jumlah Mahasiswa Sudah Diproses
8. Cek status antrian
0. Keluar
Pilih menu: 2
Melayani mahasiswa:
No. NIM - NAMA - PRODI - KELAS
1. 123 - Aldi - TI - 1E
2. 124 - Edi - TI - 1A
```

c. Menu 3

```
=== Menu Antrian layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Dua Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
6. Lihat Mahasiswa Paling Akhir
7. Jumlah Mahasiswa Sudah Diproses
8. Cek status antrian
0. Keluar
Pilih menu: 3
Mahasiswa terdepan:
No. NIM - NAMA - PRODI - KELAS
1. 125 - Budi - TI - 1B
2. 126 - Rosa - TI - 1C
```

d. Menu 4

```
=== Menu Antrian layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Dua Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
6. Lihat Mahasiswa Paling Akhir
7. Jumlah Mahasiswa Sudah Diproses
8. Cek status antrian
0. Keluar
Pilih menu: 4
Daftar Mahasiswa dalam antrian:
No. NIM - NAMA - PRODI - KELAS
1. 125 - Budi - TI - 1B
2. 126 - Rosa - TI - 1C
3. 127 - Ocad - TI - 1E
```

e. Menu 5

```
=== Menu Antrian layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Dua Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
6. Lihat Mahasiswa Paling Akhir
7. Jumlah Mahasiswa Sudah Diproses
8. Cek status antrian
0. Keluar
Pilih menu: 5
Jumlah data antrian: 3
```

f. Menu 6

```
=== Menu Antrian layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Dua Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
6. Lihat Mahasiswa Paling Akhir
7. Jumlah Mahasiswa Sudah Diproses
8. Cek status antrian
0. Keluar
Pilih menu: 6
Mahasiswa paling akhir:
No. NIM - NAMA - PRODI - KELAS
1. 127 - Ocad - TI - 1E
```

g. Menu 7

```
=== Menu Antrian layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Dua Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
6. Lihat Mahasiswa Paling Akhir
7. Jumlah Mahasiswa Sudah Diproses
8. Cek status antrian
0. Keluar
Pilih menu: 7
Jumlah data sudah diproses: 2
```

h. Menu 8

```
=== Menu Antrian layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Dua Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
6. Lihat Mahasiswa Paling Akhir
7. Jumlah Mahasiswa Sudah Diproses
8. Cek status antrian
0. Keluar
Pilih menu: 8
Status antrian: Terdapat antrian
```

i. Menu 0

```
=== Menu Antrian layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Dua Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
6. Lihat Mahasiswa Paling Akhir
7. Jumlah Mahasiswa Sudah Diproses
8. Cek status antrian
0. Keluar
Pilih menu: 0
Terima kasih.
```

j. Menu default

```
=== Menu Antrian layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Dua Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
6. Lihat Mahasiswa Paling Akhir
7. Jumlah Mahasiswa Sudah Diproses
8. Cek status antrian
0. Keluar
Pilih menu: 9
Pilihan tidak valid.
```

5. Push kode program ke github

```
PS C:\Users\luhun\Kuliah1 Semester2\Praktikum_ASD> git add .
PS C:\Users\luhun\Kuliah1 Semester2\Praktikum_ASD> git commit -m "Jobsheet 10"
[main 4880e22] Jobsheet 10
8 files changed, 541 insertions(+)
create mode 100644 P1Jobsheet10/Queue.java
create mode 100644 P1Jobsheet10/QueueMain.java
create mode 100644 P2Jobsheet10/AntrianLayanan.java
create mode 100644 P2Jobsheet10/LayananAkademikSiakad.java
create mode 100644 P2Jobsheet10/Mahasiswa.java
create mode 100644 TugasJobsheet10/DataMahasiswa.java
create mode 100644 TugasJobsheet10/KrsMain.java
create mode 100644 TugasJobsheet10/KrsQueue.java
PS C:\Users\luhun\Kuliah1 Semester2\Praktikum_ASD> git push origin main
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 8 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (13/13), 4.02 KiB | 2.01 MiB/s, done.
Total 13 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (6/6), completed with 1 local object.
remote: This repository moved. Please use the new location:
remote:   https://github.com/Kumaaan/Praktikum_ASD.git
To https://github.com/Kumaaan/Praktikum-ASD.git
    1945654..4880e22  main -> main
PS C:\Users\luhun\Kuliah1 Semester2\Praktikum_ASD>
```

6. Class Diagram

