

**LAPORAN HASIL PRAKTIKUM**  
**ALGORITMA SISTEM DATA**  
**JOBSHEET 10**



**IRFAN PANDU PRATAMA**

**244107020193**

**TI 1E**

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**  
**2025**

## PERCOBAAN 1

### 1. Kode program Queue

```
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() {  
        return size == 0;  
    }  
}
```

### 2. Kode program QueueMain

```
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);  
    }  
}
```

Hasil kode program Percobaan 1:

```
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
-----
```

Jawaban pertanyaan percobaan 1 :

1. Karena -1 menandakan queue kosong (belum ada data), dan `size = 0` berarti belum ada elemen yang masuk.
2. Mengatur posisi `rear`; jika kosong, mulai dari indeks 0, jika `rear` sudah di akhir array, kembali ke awal (0), lalu data disimpan di `rear`.
3. Mengambil data dari `front`, kurangi `size`; jika kosong setelah dequeue, reset ke -1, jika tidak, geser `front` ke indeks berikutnya (melingkar jika perlu).
4. Karena data pertama dalam antrian berada di `front`, bukan selalu di indeks 0 (karena antrian bisa bergeser).
5. Mencetak semua data dari `front` ke `rear` dalam antrian sirkular, dengan indeks `i` berputar kembali ke awal saat mencapai batas array.
6. Digunakan untuk menandai jika queue tidak dapat menerima data baru karena kapasitas penuh
7. Modifikasi kode program

```
public void Enqueue(int dt) {
    if (isFull()) {
        System.out.println(x:"Queue sudah penuh");
        System.exit(status:1);
    } else {
        if (isEmpty()) {
            front = rear = 0;
        } else {
            rear = (rear + 1) % max;
        }
        data[rear] = dt;
        size++;
    }
}

public int Dequeue() {
    int dt = 0;
    if (isEmpty()) {
        System.out.println(x:"Queue masih kosong");
        System.exit(status:1);
    } else {
```

## PERCOBAAN 2

### 1. Kode program Mahasiswa

```
public class Mahasiswa {
    String nim, nama, prodi, 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. Kode program AntrianLayanan

```
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) {
```

### 3. Kode program LayananAkademikSIKAD

```
import java.util.Scanner;
public class LayananAkademikSIKAD {
    public static void main(String[] args){
        Scanner input = 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 = input.nextInt();
        } while (pilihan != 0);
    }
}
```

Hasil kode program Percobaan 2:

=== 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 Terakhir
0. Keluar

Pilih menu: 1

Masukkan NIM: 1

Masukkan Nama: irfan

Masukkan Prodi: ti

Masukkan Kelas: 1e

irfan 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 Terakhir
0. Keluar

Pilih menu: 1

Masukkan NIM: 2

Masukkan Nama: pandu

Masukkan Prodi: mesin

Masukkan Kelas: 1a

pandu 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 Terakhir
0. Keluar

Pilih menu: 4

Daftar Mahasiswa dalam antrian:

NIM - NAMA - PRODI - KELAS

1. 1 - irfan - ti - 1e

2. 2 - pandu - mesin - 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. Lihat Mahasiswa Terakhir

0. Keluar

Pilih menu: 2

Melayani mahasiswa:

1 - irfan - 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

6. Lihat Mahasiswa Terakhir

0. Keluar

Pilih menu: 2

Melayani mahasiswa:

2 - pandu - mesin - 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. Lihat Mahasiswa Terakhir

0. Keluar

Pilih menu: 4

Antrian kosong.



```
=== 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 Terakhir
0. Keluar
Pilih menu: 0
Terima kasih!
```

Modifikasi kode program pada pertanyaan percobaan 2 :

```
case 6:
    antrian.LihatAkhir();
    break;
```

Windsurf: Refactor | Explain | Generate Javadoc | X

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

## TUGAS

### 1. Kode program Mahasiswa

```
public class Mahasiswa {
    String nim, nama, prodi, kelas;
    boolean sudahProses;

    public Mahasiswa(String nim, String nama, String prodi, String
kelas) {
        this.nim = nim;
        this.nama = nama;
        this.prodi = prodi;
        this.kelas = kelas;
        this.sudahProses = false;
    }

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

### 2. Kode program AntrianKRS

```
public class AntrianKRS {
    Mahasiswa[] data;
    int front;
    int rear;
    int size;
    int max;
    int jumlahProses;

    public AntrianKRS(int max) {
        this.max = max;
        this.data = new Mahasiswa[max];
        this.front = 0;
        this.rear = -1;
        this.size = 0;
        this.jumlahProses = 0;
    }
}
```

### 3. Kode program AntrianKRSmain

```
import java.util.Scanner;

public class AntrianKRSmain {
    public static void menu() {
        System.out.println("\n=== Menu Sistem Antrian KRS ===");
        System.out.println("1. Tambahkan Mahasiswa ke Antrian");
        System.out.println("2. Proses 2 Mahasiswa");
        System.out.println("3. Periksa Apakah Antrian Kosong");
        System.out.println("4. Periksa Apakah Antrian Penuh");
        System.out.println("5. Lihat Seluruh Antrian");
        System.out.println("6. Lihat 2 Mahasiswa Terdepan");
        System.out.println("7. Lihat Mahasiswa Terakhir");
        System.out.println("8. Jumlah Mahasiswa dalam Antrian");
        System.out.println("9. Jumlah Mahasiswa yang Sudah Diproses");
        System.out.println("10. Jumlah Mahasiswa yang Belum Diproses");
        System.out.println("11. Hapus Semua Antrian");
        System.out.println("0. Keluar");
    }
}
```

Hasil kode program Percobaan 2: