

**LAPORAN HASIL PRATIKUM**  
**ANALISIS DAN STRUKTUR DATA**  
**JOBSHEET 10**



**Raihan Akbar Putra Prasetyo/244107020087**

**Kelas: TI-1E**

**D-IV TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI**  
**PRAKTIKUM 25**

## Percobaan 1

### Kode program

#### Queue24.java

```
package jobsheet10;

public class Queue24 {
    int[] data;
    int front;
    int rear;
    int size;
    int max;

    public Queue24(int n) {
        max = n;
        data = new int[max];
        size = 0;
        front = rear = -1;
    }

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

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

    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 {
            System.out.print("Isi Queue: ");
            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");
    } 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("Queue masih kosong");
    } else {
        dt = data[front];
        size--;
        if (IsEmpty()) {
            front = rear = -1;
        } else {
            front = (front + 1) % max;
        }
    }
    return dt;
}
```

## queueMain24.java

```
package jobsheet10;
import java.util.Scanner;
public class queueMain24 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Masukkan kapasitas queue: ");
        int n = sc.nextInt();
        Queue24 q = new Queue24(n);

        int pilih;
        do {
            menu();
            System.out.print("Pilih 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;
                default:
                    System.out.println("Menu tidak tersedia.");
            }

        } while (pilih >= 1 && pilih <= 5);
    }

    public static void menu() {
        System.out.println("\n=====");
        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("=====");
    }
}
```

## Output

```
Masukkan kapasitas queue: 4

=====
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
=====
Pilih menu: 1
Masukkan data baru: 15

=====
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
=====
Pilih menu: 1
Masukkan data baru: 31

=====
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
=====
Pilih menu: 4
Elemen terdepan: 15
```

## Pertanyaan:

1. Front dan rear di-set ke -1 sebagai penanda queue masih kosong, dan size 0 karena belum ada elemen.
2. Cek apakah rear sudah di akhir array; jika ya, reset ke indeks 0 (awal).
3. Memeriksa apakah front sudah di indeks max - 1; jika iya, atur ulang ke 0.
4. Karena queue selalu dimulai dari front
5. Agar pergerakan tetap lanjut walau front atau rear melewati batas

```
6. public void enqueue(int dt) {
    if (IsFull()) {
        System.out.println(x:"Queue sudah penuh");
    } else {
        if (IsEmpty()) {
            front = rear = 0;
        } else {
            rear = (rear + 1) % max;
        }
        data[rear] = dt;
        size++;
    }
}
```

```

public void enqueue(int dt){
    if (IsFull()) {
        System.out.println(x:"Queue sudah penuh");
        return;
    }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(x:"Queue masih kosong");
        System.exit(status:1);
    }else{

```

7.

## Percobaan 2

### kode program

#### Mahasiswa24.java

```
package jobsheet10;

public class Mahasiswa24 {
    String nim, nama, prodi, kelas;

    public Mahasiswa24 (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 );
    }
}
```

#### Antrianlayanan24.java

```
package jobsheet10;

public class AntrianLayanan24 {
    Mahasiswa24[] data;
    int front;
    int rear;
    int size;
    int max;

    public AntrianLayanan24(int max) {
        this.max = max;
        this.data = new Mahasiswa24[max];
        this.front = 0;
        this.rear = -1;
        this.size = 0;
    }

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

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

    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 = 0;
        rear = -1;
        size = 0;
        System.out.println("Queue berhasil dikosongkan");
    } else {
        System.out.println("Queue masih kosong");
    }
}

public void tambahAntrian(Mahasiswa24 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 Mahasiswa24 layaniMahasiswa() {
    if (IsEmpty()) {
        System.out.println("Antrian kosong");
        return null;
    }
    Mahasiswa24 mhs = data[front];
    front = (front + 1) % max;
    size--;
    return mhs;
}

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;
}

```



## LayananAkademikSiakad24.java

```
package jobsheet10;

import java.util.Scanner;

public class LayananAkademikSiakad24 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        AntrianLayanan24 antrian = new AntrianLayanan24(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 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();

                    Mahasiswa24 mhs = new Mahasiswa24(nim, nama, prodi, kelas);
                    antrian.tambahAntrian(mhs);
                    break;

                case 2:
                    Mahasiswa24 dilayani = antrian.layaniMahasiswa();
                    if (dilayani != null) {
                        System.out.println("Melayani Mahasiswa:");
                        dilayani.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.");
                    break;
            }

        } while (pilihan != 0);
    }
}
```

## Output

```
=== Menu Antrian Layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat 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 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 Antrian
5. Jumlah Mahasiswa dalam Antrian
0. Keluar
Pilih menu: 4
Daftar Mahasiswa dalam Antrian:
```

```
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 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 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 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 Antrian
5. Jumlah Mahasiswa dalam Antrian
0. Keluar
Pilih menu: 0
Terima kasih!
PS D:\kuliah\PRAKTIKUM-ASD>
```

## Pertanyaan

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

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

## Tugas:

### MahasiswaTugas24.java

```
package jobsheet10;

public class MahasiswaTugas24 {
    public String nama;
    public String nim;
    public String prodi;
    public String kelas;

    public MahasiswaTugas24(String nama, String nim, String prodi, String kelas) {
        this.nama = nama;
        this.nim = nim;
        this.prodi = prodi;
        this.kelas = kelas;
    }
}
```

### queueKrs24.java

```
package jobsheet10;

public class QueueKrs24 {
    private int maxAntrian = 10;
    private MahasiswaTugas24[] queue = new MahasiswaTugas24[maxAntrian];
    private int front = 0;
    private int rear = -1;
    private int size = 0;
    private int jumlahProses = 0;

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

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

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

    public void tambahMahasiswa(MahasiswaTugas24 mhs) {
        if (isFull()) {
            System.out.println("Antrian penuh, tidak bisa menambahkan mahasiswa.");
        } else {
            rear = (rear + 1) % maxAntrian;
            queue[rear] = mhs;
            size++;
            System.out.println(mhs.nama + " ditambahkan ke antrian.");
        }
    }

    public void panggilProsesKRS() {
        if (size >= 2) {
            MahasiswaTugas24 m1 = dequeue();
            MahasiswaTugas24 m2 = dequeue();
            jumlahProses += 2;
            System.out.println("Memproses KRS untuk:");
            tampilkanMahasiswa(m1);
            tampilkanMahasiswa(m2);
        } else {
            System.out.println("Antrian tidak cukup untuk memproses 2");
        }
    }
}
```

```

public MahasiswaTugas24 dequeue() {
    if (isEmpty()) return null;
    MahasiswaTugas24 mhs = queue[front];
    front = (front + 1) % maxAntrian;
    size--;
    return mhs;
}

public void tampilkanSemua() {
    if (isEmpty()) {
        System.out.println("Antrian kosong.");
    } else {
        System.out.println("Daftar antrian:");
        for (int i = 0; i < size; i++) {
            int index = (front + i) % maxAntrian;
            System.out.print((i + 1) + ". ");
            tampilkanMahasiswa(queue[index]);
        }
    }
}

public void tampilkanDepan() {
    if (!isEmpty()) {
        System.out.println("Antrian terdepan:");
        tampilkanMahasiswa(queue[front]);
    } else {
        System.out.println("Antrian kosong.");
    }
}

public void tampilkanDuaTerdepan() {
    if (size >= 2) {
        System.out.println("1.");
        tampilkanMahasiswa(queue[front]);
        System.out.println("2.");
        tampilkanMahasiswa(queue[(front + 1) % maxAntrian]);
    } else if (size == 1) {
        System.out.println("1.");
        tampilkanMahasiswa(queue[front]);
        System.out.println("2. Tidak ada");
    } else {
        System.out.println("Antrian kosong.");
    }
}

public void tampilkanAkhir() {
    if (!isEmpty()) {
        System.out.println("Antrian terakhir:");
        tampilkanMahasiswa(queue[rear]);
    } else {
        System.out.println("Antrian kosong.");
    }
}

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

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

public void cetakBelumProses() {
    System.out.println("Jumlah mahasiswa yang belum melakukan KRS: " + size);
}

private void tampilkanMahasiswa(MahasiswaTugas24 m) {
    if (m != null) {
        System.out.println("Nama : " + m.nama);
        System.out.println("NIM : " + m.nim);
        System.out.println("Prodi: " + m.prodi);
        System.out.println("Kelas: " + m.kelas);
    }
}

```

## KrsMain24.java

```
package jobsheet10;

import java.util.Scanner;

public class KrsMain24 {
    public static void main(String[] args) {
        QueueKrs24 antrian = new QueueKrs24();
        Scanner scanner = new Scanner(System.in);
        int pilihan;

        do {
            System.out.println("\n=== MENU ANTRIAN KRS ===");
            System.out.println("1. Cek antrian kosong");
            System.out.println("2. Cek antrian penuh");
            System.out.println("3. Kosongkan antrian");
            System.out.println("4. Tambah mahasiswa ke antrian");
            System.out.println("5. Panggil 2 mahasiswa untuk proses KRS");
            System.out.println("6. Tampilkan semua antrian");
            System.out.println("7. Tampilkan 2 antrian terdepan");
            System.out.println("8. Tampilkan antrian terakhir");
            System.out.println("9. Cetak jumlah antrian");
            System.out.println("10. Cetak jumlah yang sudah proses KRS");
            System.out.println("11. Cetak jumlah yang belum proses KRS");
            System.out.println("0. Keluar");
            System.out.print("Pilih menu: ");
            pilihan = scanner.nextInt();
            scanner.nextLine();

            switch (pilihan) {
                case 1:
                    System.out.println(antrian.isEmpty() ? "Antrian kosong." : "Antrian tidak kosong.");
                    break;
                case 2:
                    System.out.println(antrian.isFull() ? "Antrian penuh." : "Antrian belum penuh.");
                    break;
                case 3:
                    antrian.kosongkan();
                    break;
                case 4:
                    System.out.print("Nama: ");
                    String nama = scanner.nextLine();
                    System.out.print("NIM: ");
                    String nim = scanner.nextLine();
                    System.out.print("Prodi: ");
                    String prodi = scanner.nextLine();
                    System.out.print("Kelas: ");
                    String kelas = scanner.nextLine();
                    MahasiswaTugas24 mhs = new MahasiswaTugas24(nama, nim, prodi, kelas);
                    antrian.tambahMahasiswa(mhs);
                    break;
                case 5:
                    antrian.panggilProsesKRS();
                    break;
                case 6:
                    antrian.tampilkanSemua();
                    break;
                case 7:
                    antrian.tampilkanDuaTerdepan();
                    break;
                case 8:
                    antrian.tampilkanAkhir();
                    break;
                case 9:
                    antrian.cetakJumlah();
                    break;
                case 10:
                    antrian.cetakJumlahProses();
                    break;
                case 11:
                    antrian.cetakJumlahBelumProses();
                    break;
            }
        } while (pilihan != 0);
    }
}
```

```
case 0:
    System.out.println("Terima kasih!");
    break;
default:
    System.out.println("Pilihan tidak valid.");
}

} while (pilihan != 0);

}
}
```