

**LAPORAN HASIL PRAKTIKUM ALGORITMA DAN  
STRUKTUR DATA**

**JOBSHEET 10**



**MUHAMMAD AMMAR HAFIZH**

**(2341720074)**

**D-IV TEKNIK INFORMATIKA – 1E**

**Jurusan Teknologi Informasi**

**Politeknik Negeri Malang**

## Hasil Praktikum Percobaan 1

```
Masukan Kapasitas Queue : 4
Masukan operasi yang diinginkan :
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
1
Masukan data baru : 15
Masukan operasi yang diinginkan :
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
1
Masukan data baru : 31
Masukan operasi yang diinginkan :
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
4
Elemen Terdepan : 15
```

## Pertanyaan Percobaan 1

1. Pada konstruktor, mengapa nilai awal atribut front dan rear bernilai -1, sementara atribut size bernilai 0?

- Front Rear bernilai -1 agar tidak menunjuk atau meposisikan diri ke dalam queue atau antrian yang belum ada isinya dan size berawal dari 0 karena elemen masih kosong belum diisi apapun dan jika sudah terisi maka size akan mengikuti panjangnya antrian dan akan berhenti enqueue saat size sama dengan max.

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

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

- Untuk mengubah rear pada posisi terdepan dan front posisi terbelakang karena kolom terbelakang atau rear sudah terisi dan kolom depan masih kosong dan nilai yang dimasukan akan masuk pada kolom depan dan dijadikan rear.

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

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

- Ketika semua antrian sudah dikeluarkan dan sisa baris terbelakang yang ingin dikeluarkan maka front akan direset menjadi paling depan lagi.

4. Pada method **print**, mengapa pada proses perulangan variabel i tidak dimulai dari 0 (**int i=0**), melainkan **int i=front**?

- Karena blm tentu antrian terdepan atau front bernilai indeks 0

5. Perhatikan kembali method **print**, jelaskan maksud dari potongan kode berikut!

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

- Untuk melakukan looping dan jika sudah max antrian makan akan di modulus sehingga I menjadi 0 dan looping diberhentikan

6. Tunjukkan potongan kode program yang merupakan queue overflow!

- 

```
public boolean isFull() {
    if (size == max) {
        return true;
    } else {
        return false;
    }
}

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

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

- **Before Overflow**

```
case 1:
    System.out.print(s:"Masukan data baru : ");
    int dataMasuk = sc.nextInt();
    Q.Enqueue(dataMasuk);
    break;
```

- **After Overflow**

```
case 1:
    if (!Q.isFull()) {
        System.out.print(s:"Masukan data baru : ");
        int dataMasuk = sc.nextInt();
        Q.Enqueue(dataMasuk);
        break;
    } else {
        return;
    }
}
```

- **Before Underflow**

```
case 2:
    int dataKeluar = Q.Dequeue();
    if (dataKeluar != 0) {
        System.out.println("Data yang dikeluarkan : " + dataKeluar);
        break;
    }
}
```

- **After Underflow**

```
case 2:
    if (!Q.isEmpty()) {
        int dataKeluar = Q.Dequeue();
        if (dataKeluar != 0) {
            System.out.println("Data yang dikeluarkan : " + dataKeluar);
            break;
        }
    } else {
        return;
    }
}
```

## Hasil Praktikum Percobaan 2

```
Masukan kapasitas Queue : 8
Pilih Menu :
1. Antrian Baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian
-----
1
No Rekening : 12345
Nama : Dewi
Alamat : Malang
Umur : 23
Saldo : 1300000
Pilih Menu :
1. Antrian Baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian
-----
1
No Rekening : 32940
Nama : Susan
Alamat : Surabaya
Umur : 39
Saldo : 42000000
Pilih Menu :
1. Antrian Baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian
-----
4
Elemen Terdepan : 12345 Dewi Malang 23 1300000.0
Elemen Terdepan : 32940 Susan Surabaya 39 4.2E7
Jumlah Elemen : 2
```

## Pertanyaan Percobaan 2

1. Pada class QueueMain, jelaskan fungsi IF pada potongan kode program berikut!

```
if (!"".equals(data.norek) && !"".equals(data.nama) && !"".equals(data.alamat)
    && data.umur != 0 && data.saldo != 0) {
    System.out.println("Antrian yang keluar: " + data.norek + " " + data.nama + " "
        + data.alamat + " " + data.umur + " " + data.saldo);
    break;
}
```

- Code tersebut mencegah terjadinya underflow

2. Lakukan modifikasi program dengan menambahkan method baru bernama peekRear pada class Queue yang digunakan untuk mengecek antrian yang berada di posisi belakang! Tambahkan pula daftar menu 5. Cek Antrian paling belakang pada class QueueMain sehingga method peekRear dapat dipanggil!

- **Method**

```
public void peekRear(){
    if (!isEmpty()) {
        System.out.println("Elemen Terbelakang : " + data[rear].norek + " "
            + data[rear].nama + " " + data[rear].alamat + " " + data[rear].umur + " " + data[rear].saldo);
    } else {
        System.out.println(x:"Queue masih kosong");
    }
}
```

- **Main**

```
        case 5:
            antri.peekRear();
            break;

        default:
            break;
    }

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

Codeium: Refactor | Explain | Generate Javadoc | X
public static void menu() {
    System.out.println(x:"Pilih Menu : ");
    System.out.println(x:"1. Antrian Baru");
    System.out.println(x:"2. Antrian Keluar");
    System.out.println(x:"3. Cek Antrian Terdepan");
    System.out.println(x:"4. Cek Semua Antrian");
    System.out.println(x:"5. Cek Antrian Terbelakang");
    System.out.println(x:"-----");
}
```

## Hasil Tugas

```
Pilih Menu :
1. Antrian Baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian
5. Cek Antrian Terbelakang
6. Cek Antrian Berdasarkan Nama
7. Daftar Pembeli
```

-----

3

Elemen Terdepan : Rhizqy

```
Pilih Menu :
1. Antrian Baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian
5. Cek Antrian Terbelakang
6. Cek Antrian Berdasarkan Nama
7. Daftar Pembeli
```

-----

4

Nama : Rhizqy Nomor HP : 812345

Nama : Ammar Nomor HP : 832156

Nama : Tomo Nomor HP : 844567

```
Pilih Menu :
1. Antrian Baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian
5. Cek Antrian Terbelakang
6. Cek Antrian Berdasarkan Nama
7. Daftar Pembeli
```

-----

5

Elemen Terbelakang : Tomo

```
Pilih Menu :
1. Antrian Baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian
5. Cek Antrian Terbelakang
6. Cek Antrian Berdasarkan Nama
7. Daftar Pembeli
```

-----

6

Masukan Nama : Ammar

Antrian dengan nama Ammar berada di posisi ke-2

```
Pilih Menu :
1. Antrian Baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian
5. Cek Antrian Terbelakang
6. Cek Antrian Berdasarkan Nama
7. Daftar Pembeli
```

-----

7

Nama : Rhizqy Nomor HP : 812345

Nama : Ammar Nomor HP : 832156

Nama : Tomo Nomor HP : 844567