

LAPORAN PRAKTIKUM

Algoritma Dan Struktur Data

Jobsheet - 8 : Queue



Nama : Ghoffar Abdul Ja'far

NIM : 41720035

Kelas : 1E

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2023/2024

Praktikum 1

Hasil:

```
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
```

Pernyataan

1. Pada konstruktor, mengapa nilai awal atribut front dan rear bernilai -1, sementara atribut size bernilai 0?
= Karena front dan rear bernilai -1 yang mana front dan rear tidak menunjuk nilai pada queue, maka size bernilai 0.
2. Pada method Enqueue, jelaskan maksud dan kegunaan dari potongan kode berikut!

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


= Digunakan saat rear baru ditambahkan dan posisi rear sudah sampai index terakhir yaitu $\text{max} - 1$, maka rear baru akan berada pada index 0.
3. Pada method Dequeue, jelaskan maksud dan kegunaan dari potongan kode berikut!

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


= Digunakan saat front lama dikeluarkan dan posisi front sudah sampai index terakhir yaitu $\text{max} - 1$, maka front baru akan berada pada index 0.
4. Pada method print, mengapa pada proses perulangan variabel i tidak dimulai dari 0 ($\text{int } i=0$), melainkan $\text{int } i = \text{front}$?
= Karena queue di print dimulai dari front dan front tidak selalu dimulai dari indeks ke 0 maka I perlu diawali dari front.
5. Perhatikan kembali method print, jelaskan maksud dari potongan kode berikut!

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


= Digunakan untuk menunjuk front jika front sudah sampai index terakhir maka akan dimodulus sebanyak max agar dapat Kembali ke indeks 0
6. Tunjukkan potongan kode program yang merupakan queue overflow!

```

if (isFull()) {
    System.out.println(x:"Queue sudah penuh")
} else {
    if (isEmpty()) {
        front = rear = 0;
    } else {
        if (rear == max -1) {
            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!

Enqueue:

```

public void Enqueue(int dt){
    if (isFull()) {
        System.out.println(x:"Queue sudah penuh");
        System.exit(status:0);
    } else {

```

Dequeue:

```

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

```

Hasil:

```

1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
1
Masukkan data baru: 1
Queue sudah penuh
PS E:\Kuliah\P_Algoritma_Struktur_Data>

```

Praktium 2

Hasil:

```

Masukkan kapasitas queue: 8
Pilih menu:
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
-----
1
No Rekening: 12345
Nama: Devi
Alamat: Malang
Umur: 23
Saldo: 1300000
Pilih menu:
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
-----

```

```

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
12345 Devi Malang 23 1300000.0
32940 Susan Surabaya 39 4.2E7
Jumlah elemen = 2
Pilih menu:
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
-----

```

Pertanyaan

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;
}

```

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!

Code:

```

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

public static void menu() {
    System.out.println("Pilih menu: ");
    System.out.println("1. Antrian baru");
    System.out.println("2. Antrian Keluar");
    System.out.println("3. Cek Antrian terdepan");
    System.out.println("4. Cek Semua Antrian");
    System.out.println("5. Cek Antrian paling belakang");
    System.out.println("-----");
}

```

Hasil:

```

Masukkan kapasitas queue: 4
Pilih menu:
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian paling belakang
-----
1
No Rekening: 11
Nama: aki
Alamat: lawang
Umur: 23
Saldo: 200000
Pilih menu:
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian paling belakang
-----
1
No Rekening: 22
Nama: aka
Alamat: lawang
Umur: 24
Saldo: 3000000
Pilih menu:
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian paling belakang
-----
5
Elemen terakhir adalah: 22 aka lawang 24 3000000.0

```

Tugas

Hasil:

```
Masukkan kapasitas queue: 4
Pilih menu:
1. Pembeli baru
2. Pembeli Keluar
3. Cek Pembeli terdepan
4. Cek Semua Pembeli
5. Cek Pembeli paling belakang
-----
1
Nama: asep
No HP: 87236512
Pilih menu:
1. Pembeli baru
2. Pembeli Keluar
3. Cek Pembeli terdepan
4. Cek Semua Pembeli
5. Cek Pembeli paling belakang
-----
1
Nama: jeki
No HP: 82735261
Pilih menu:
1. Pembeli baru
2. Pembeli Keluar
3. Cek Pembeli terdepan
4. Cek Semua Pembeli
5. Cek Pembeli paling belakang
-----
1
Nama: abe
No HP: 826235135
```

```
Pilih menu:
1. Pembeli baru
2. Pembeli Keluar
3. Cek Pembeli terdepan
4. Cek Semua Pembeli
5. Cek Pembeli paling belakang
-----
3
Elemen terdepan: asep 87236512
Pilih menu:
1. Pembeli baru
2. Pembeli Keluar
3. Cek Pembeli terdepan
4. Cek Semua Pembeli
5. Cek Pembeli paling belakang
-----
4
asep 87236512
jeki 82735261
abe 826235135
Jumlah elemen = 3
Pilih menu:
1. Pembeli baru
2. Pembeli Keluar
3. Cek Pembeli terdepan
4. Cek Semua Pembeli
5. Cek Pembeli paling belakang
-----
5
Elemen terakhir adalah: abe 826235135
```