

Jobsheet 8

Praktikum Algoritma & Struktur Data



Rafi Ody Prasetyo
(2341720180)

D-IV Teknik Informatika
Politeknik Negeri Malang
Semester 2
2024

Hasil 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
-----
2
Data yang keluar: 15
```

```
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
3
31
Jumlah elemen: 1
```

Pertanyaan Percobaan 1

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

Jawab:

Nilai awal front & rear bernilai diatur -1 untuk menunjukkan bahwa antrian kosong. Hal ini dikarenakan front adalah elemen pertama dan rear adalah elemen terakhir. Sedangkan atribut size bernilai 0 dikarenakan belum ada elemen yang dimasukkan saat konstruktor dipanggil.

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

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

Jawab:

Untuk mengatur jika rear sudah mencapai batas akhir dari array, maka akan kembali ke elemen awal array untuk mengisi elemen yang kosong.

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

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

Jawab:

Untuk memperbarui posisi dari front apabila masih ada elemen yang masih bisa dihapus.

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

Jawab:

Karena data yang dicetak akan dimulai pada front, dan front tidak selalu berada pada elemen ke 0. Apabila telah terjadi dequeue maka elemen front akan berubah pada elemen selanjutnya.

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

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

Jawab:

Variabel i digunakan untuk mencetak elemen ke i, ketika sudah dicetak operasi “(i+1)%max” digunakan untuk menggeser elemen i ke elemen selanjutnya.

6. Tunjukkan potongan kode program yang merupakan queue overflow!

Jawab:

```
Codeium: Refactor | Explain | Generate Javadoc | X  
public void Enqueue(int dt) {  
    if (isFull()) {  
        System.out.println(x:"Queue sudah penuh");  
    } else {  
        if (isEmpty()) {
```

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!

Jawab:

Code:

```
Codeium: Refactor | Explain | Generate Javadoc | X
public void Enqueue(int dt) {
    if (isFull()) {
        System.out.println(x:"Queue sudah penuh!! Program akan berhenti...");
        System.exit(status:0);
    } else {
```

Output:

```
Masukkan kapasitas queue: 2
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
1
Masukkan data baru: 1
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
1
Masukkan data baru: 1
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
1
Masukkan data baru: 1
Queue sudah penuh!! Program akan berhenti...
PS D:\tugas_kuliah\Semester 2\Praktikum Algoritma dan Struktur Data (smt 2)>
```

8. Push & Commit Github

<https://github.com/rafiody16/Praktikum-Algoritma-dan-Struktur-Data--smt-2-/tree/main/Jobsheet%208/Percobaan%201>

Hasil Percobaan 2

```
Masukkan kapasitas queue: 8
Masukkan operasi yang diinginkan:
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
Masukkan operasi yang diinginkan:
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
Masukkan operasi yang diinginkan:
1. Antrian Baru
2. Antrian Keluar
3. Cek antrian terdepan
4. Cek semua antrian
-----
1
Antrian yang keluar: 12345 Dewi Malang 23 1300000.0
```

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;
}
```

Jawab:

Fungsi tersebut digunakan untuk memastikan semua atribut memiliki nilai. Jika atribut memiliki nilai maka “data” akan mencetak nilai dari atribut tersebut.

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!

Jawab:

Code:

```
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");
    }
}
```

Output:

```
93239289 Jusa Kediri 19 1999000.0
1289189 Lundi Bogor 28 2000000.0
81291298 Zalfa Sidoarjo 21 2.0000001E8
Jumlah elemen: 3
Masukkan operasi yang diinginkan:
1. Antrian Baru
2. Antrian Keluar
3. Cek antrian terdepan
4. Cek semua antrian
5. Cek antrian paling belakang
-----
3
Elemen terdepan: 93239289 Jusa Kediri 19 1999000.0
Masukkan operasi yang diinginkan:
1. Antrian Baru
2. Antrian Keluar
3. Cek antrian terdepan
4. Cek semua antrian
5. Cek antrian paling belakang
-----
5
Elemen terbelakang: 81291298 Zalfa Sidoarjo 21 2.0000001E8
```

3. Push & Commit Github

https://github.com/rafiody16/Praktikum-Algoritma-dan-Struktur-Data--smt-2-/tree/main/Jobsheet%208/Percobaan_2

Latihan Praktikum

Code: <https://github.com/rafiody16/Praktikum-Algoritma-dan-Struktur-Data--smt-2-/tree/main/Jobsheet%208/LatihanPraktikum>

Output:

```
Masukkan kapasitas queue: 8
Masukkan operasi yang diinginkan:
1. Antrian Baru
2. Antrian Keluar
3. Cek antrian terdepan
4. Tampil data pembeli
5. Cek antrian paling belakang
6. Cari antrian
-----
1
Nama: yusuf
No.Hp: 212112121
Masukkan operasi yang diinginkan:
1. Antrian Baru
2. Antrian Keluar
3. Cek antrian terdepan
4. Tampil data pembeli
5. Cek antrian paling belakang
6. Cari antrian
-----
1
Nama: juju
No.Hp: 212111111
Masukkan operasi yang diinginkan:
1. Antrian Baru
2. Antrian Keluar
3. Cek antrian terdepan
4. Tampil data pembeli
5. Cek antrian paling belakang
6. Cari antrian
-----
1
Nama: goni
No.Hp: 21211211
```

```
Masukkan operasi yang diinginkan:
1. Antrian Baru
2. Antrian Keluar
3. Cek antrian terdepan
4. Tampil data pembeli
5. Cek antrian paling belakang
6. Cari antrian
-----
2
Antrian yang keluar: yusuf 212112121
Masukkan operasi yang diinginkan:
1. Antrian Baru
2. Antrian Keluar
3. Cek antrian terdepan
4. Tampil data pembeli
5. Cek antrian paling belakang
6. Cari antrian
-----
3
Elemen terdepan: juju 212111111
Masukkan operasi yang diinginkan:
1. Antrian Baru
2. Antrian Keluar
3. Cek antrian terdepan
4. Tampil data pembeli
5. Cek antrian paling belakang
6. Cari antrian
-----
4
juju 212111111
goni 21211211
Jumlah elemen: 2
```

```
Masukkan operasi yang diinginkan:
1. Antrian Baru
2. Antrian Keluar
3. Cek antrian terdepan
4. Tampil data pembeli
5. Cek antrian paling belakang
6. Cari antrian
-----
5
Elemen terbelakang: goni 21211211
Masukkan operasi yang diinginkan:
1. Antrian Baru
2. Antrian Keluar
3. Cek antrian terdepan
4. Tampil data pembeli
5. Cek antrian paling belakang
6. Cari antrian
-----
6
Masukkan nama yang dicari: juju
juju berada di antrian ke- 1
```

