

**LAPORAN PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA**

**PRAKTIKUM 8
QUEUE**



**Oleh:
JESSICA AMELIA
2341760185
SIB 1A/16**

**PROGRAM STUDI SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2023/2024**

Praktikum 1

```
D:\3c024a5\bin' 'PRAKTIKUM08.QueueDemo'
Masukkan kapasitas queue: 6
=====
Masukkan operasi yang diinginkan
=====
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
6. Exit
-----
1
Masukkan data baru: 15
=====
Masukkan operasi yang diinginkan
=====
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
6. Exit
-----
1
Masukkan data baru: 23
=====
Masukkan operasi yang diinginkan
=====
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
6. Exit
-----
3
15 23
Jumlah elemen: 2
=====
Masukkan operasi yang diinginkan
=====
```

```
=====
Masukkan operasi yang diinginkan
=====
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
6. Exit
-----
4
Elemen terdepan: 15
=====
Masukkan operasi yang diinginkan
=====
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
6. Exit
-----
2
Data yang dikeluarkan: 15
=====
Masukkan operasi yang diinginkan
=====
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
6. Exit
-----
3
23
Jumlah elemen: 1
=====
Masukkan operasi yang diinginkan
=====
1. Enqueue
```

1. Apakah nilai front selalu lebih kecil dibanding nilai rear? Jelaskan

Jawab : Tidak, Nilai front dan rear adalah petunjuk untuk mengatur elemen-elemen dalam antrian, dan keduanya bergerak sesuai dengan operasi enqueue (penambahan elemen) dan dequeue (penghapusan elemen) yang dilakukan.

2. Pada method enqueue(), jelaskan maksud dan kegunaan dari potongan kode berikut

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

Jawab : Potongan kode tersebut digunakan ketika data paling belakang dari queue berada pada indeks terakhir array.

3. Pada method dequeue(), jelaskan maksud dan kegunaan dari potongan kode berikut

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

Jawab : Potongan kode tersebut digunakan ketika data paling depan dari queue berada pada indeks terakhir array.

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

Jawab : Karena kita ingin mencetak elemen-elemen yang ada dalam antrian, dimana dalam antrian elemen-elemen baru yang dimasukkan akan selalu ditambahkan dibelakang (rear) dan

elemen-elemen lama yang telah diambil akan dihapus dari depan(front). Dengan memulai perulangan dari front, kita mengawali pencetakan dari elemen pertama yang ada dalam antrian, yang sebenarnya adalah elemen yang paling depan dan harus diprioritaskan untuk dicetak terlebih dahulu. Jika kita memulai perulangan dari 0, kita akan mencetak seluruh array, termasuk elemen yang belum tentu ada dalam antrian (elemen yang mungkin telah dihapus sebelumnya).

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

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

Jawab : Potongan kode $i = (i + 1) \% \text{max}$; dalam method print digunakan untuk memperbarui nilai indeks i dengan cara melakukan operasi penambahan satu terhadap i, dan kemudian menghitung sisa hasil bagi dari penjumlahan tersebut dengan max.

6. Apa yang dimaksud dengan queue overflow dan queue underflow? Tunjukkan potongan kode program di mana kondisi tersebut terjadi

Jawab : Queue overflow ini adalah kondisi yang dihasilkan dari mencoba menambahkan elemen ke queue yang sudah penuh, sementara Queue underflow adalah kondisi yang dihasilkan dari mencoba menghapus elemen dari queue yang masih kosong.

- Queue overflow:

```
if (isFull()){
    System.out.println(x:"Queue sudah penuh");
} else {
    if (isEmpty()){
```

- Queue underflow:

```
if(isEmpty()){
    System.out.println(x:"Queue masih kosong");
} else {
```

Praktikum 2

```
Masukkan kapasitas queue: 2
=====
Masukkan operasi yang diinginkan
=====
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Cek Antrian Belakang
6. Clear
7. Exit
-----
1
Masukkan no plat: A 1234 BC
Masukkan no kartu: 852304
Masukkan saldo: 20000
=====
Masukkan operasi yang diinginkan
=====
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Cek Antrian Belakang
6. Clear
7. Exit
-----
1
Masukkan no plat: B 123 NM
Masukkan no kartu: 92934
Masukkan saldo: 40000
=====
```

```
=====
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Cek Antrian Belakang
6. Clear
7. Exit
-----
2
Data yang dikeluarkan: No Plat: A 1234 BC, No Kartu: 852304, Saldo: 20000.0
=====
Masukkan operasi yang diinginkan
=====
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Cek Antrian Belakang
6. Clear
7. Exit
-----
4
Antrian terdepan: No Plat: B 123 NM, No Kartu: 92934, Saldo: 40000.0
=====
Masukkan operasi yang diinginkan
=====
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Cek Antrian Belakang
6. Clear
7. Exit
-----
3
No Plat: B 123 NM, No Kartu: 92934, Saldo: 40000.0
Jumlah elemen: 1
=====
Masukkan operasi yang diinginkan
=====
```

1. Pada class Kendaraan, mengapa perlu dibuat method toString()? Cobalah comment fungsi tersebut kemudian jalankan program? Apakah terjadi compile error? Jika tidak, bagaimana output dari program?

Jawab : Method toString() dibuat untuk memberikan representasi string dari objek Kendaraan. Method ini sangat berguna karena memungkinkan kita untuk mencetak informasi tentang objek Kendaraan ke layar atau menyimpannya dengan mudah. Jika method toString di comment dan program dijalankan tidak akan terjadi compile error, karena method toString() adalah method bawaan dari kelas objek dalam java. Namun Output dari program setelah di comment akan berbeda dari yang diharapkan, karena tidak akan mencetak informasi yang berguna tentang objek Kendaraan. Contoh outputnya yaitu Data yang dikeluarkan: PRAKTIKUM08.GerbangTol.Kendaraan@55f96302

2. Lakukan modifikasi program dengan menambahkan method baru bernama peekRear() pada class Queue yang digunakan untuk mengecek kendaraan yang berada di posisi paling belakang! Tambahkan pula menu baru Cek antrian paling belakang pada daftar menu dan lakukan pemanggilan method peekRear()

Jawab :

```
public void peekRear(){
    if (!isEmpty()){
        System.out.println("Antrian paling belakang: " + data[rear]);
    }else {
        System.out.println(x:"Queue kosong");
    }
}
```

```
Masukkan operasi yang diinginkan
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Cek Antrian Belakang
6. Clear
7. Exit
-----
3
No Plat: Ag 2345 BC, No Kartu: 456, Saldo: 70000.0
No Plat: S 6378 HJ, No Kartu: 5231, Saldo: 90000.0
No Plat: N 456 GH, No Kartu: 6234, Saldo: 40000.0
Jumlah elemen: 3

Masukkan operasi yang diinginkan
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Cek Antrian Belakang
6. Clear
7. Exit
-----
5
Antrian paling belakang: No Plat: N 456 GH, No Kartu: 6234, Saldo: 40000.0
```

Tugas

Masukkan kapasitas pasien: 4

Daftar Menu Klinik:

1. Tambah Pasien
2. Keluarkan Data Pasien
3. Lihat Daftar Antrian
4. Lihat Pasien Antrian Terdepan
5. Lihat Pasien Antrian Paling Belakang
6. Cari Posisi Antrian Pasien
7. Menghapus Semua Data Pasien
8. Exit

1
Masukkan Nama: Yudha
Masukkan no ID: 123
Masukkan jenis kelamin (L/P): 1

Daftar Menu Klinik:

1. Tambah Pasien
2. Keluarkan Data Pasien
3. Lihat Daftar Antrian
4. Lihat Pasien Antrian Terdepan
5. Lihat Pasien Antrian Paling Belakang
6. Cari Posisi Antrian Pasien
7. Menghapus Semua Data Pasien
8. Exit

1
Masukkan Nama: Rara
Masukkan no ID: 345
Masukkan jenis kelamin (L/P): p

Daftar Menu Klinik:

1. Tambah Pasien
2. Keluarkan Data Pasien
3. Lihat Daftar Antrian
4. Lihat Pasien Antrian Terdepan
5. Lihat Pasien Antrian Paling Belakang

3. Lihat Daftar Antrian
4. Lihat Pasien Antrian Terdepan
5. Lihat Pasien Antrian Paling Belakang
6. Cari Posisi Antrian Pasien
7. Menghapus Semua Data Pasien
8. Exit

1
Masukkan Nama: Wisnu
Masukkan no ID: 567
Masukkan jenis kelamin (L/P): 1

Daftar Menu Klinik:

1. Tambah Pasien
2. Keluarkan Data Pasien
3. Lihat Daftar Antrian
4. Lihat Pasien Antrian Terdepan
5. Lihat Pasien Antrian Paling Belakang
6. Cari Posisi Antrian Pasien
7. Menghapus Semua Data Pasien
8. Exit

1
Masukkan Nama: Siska
Masukkan no ID: 789
Masukkan jenis kelamin (L/P): p

Daftar Menu Klinik:

1. Tambah Pasien
2. Keluarkan Data Pasien
3. Lihat Daftar Antrian
4. Lihat Pasien Antrian Terdepan
5. Lihat Pasien Antrian Paling Belakang
6. Cari Posisi Antrian Pasien
7. Menghapus Semua Data Pasien
8. Exit

1
Antrian sudah penuh.

Daftar Menu Klinik:

1. Tambah Pasien
2. Keluarkan Data Pasien
3. Lihat Daftar Antrian
4. Lihat Pasien Antrian Terdepan
5. Lihat Pasien Antrian Paling Belakang
6. Cari Posisi Antrian Pasien
7. Menghapus Semua Data Pasien
8. Exit

3
Daftar Pasien dalam Antrian:
Nama Pasien: Yudha, No ID: 123, Jenis Kelamin: 1
Nama Pasien: Rara, No ID: 345, Jenis Kelamin: p
Nama Pasien: Wisnu, No ID: 567, Jenis Kelamin: 1
Nama Pasien: Siska, No ID: 789, Jenis Kelamin: p
Jumlah pasien: 4

Daftar Menu Klinik:

1. Tambah Pasien
2. Keluarkan Data Pasien
3. Lihat Daftar Antrian
4. Lihat Pasien Antrian Terdepan
5. Lihat Pasien Antrian Paling Belakang
6. Cari Posisi Antrian Pasien
7. Menghapus Semua Data Pasien
8. Exit

2
Data Pasien selesai diperiksa: Nama Pasien: Yudha, No ID: 123, Jenis Kelamin: 1

Daftar Menu Klinik:

1. Tambah Pasien
2. Keluarkan Data Pasien
3. Lihat Daftar Antrian
4. Lihat Pasien Antrian Terdepan
5. Lihat Pasien Antrian Paling Belakang
6. Cari Posisi Antrian Pasien

6. Cari Posisi Antrian Pasien
7. Menghapus Semua Data Pasien
8. Exit

4
Pasien diantrian depan: Nama Pasien: Rara, No ID: 345, Jenis Kelamin: p

Daftar Menu Klinik:

1. Tambah Pasien
2. Keluarkan Data Pasien
3. Lihat Daftar Antrian
4. Lihat Pasien Antrian Terdepan
5. Lihat Pasien Antrian Paling Belakang
6. Cari Posisi Antrian Pasien
7. Menghapus Semua Data Pasien
8. Exit

5
Pasien diantrian belakang: Nama Pasien: Siska, No ID: 789, Jenis Kelamin: p

Daftar Menu Klinik:

1. Tambah Pasien
2. Keluarkan Data Pasien
3. Lihat Daftar Antrian
4. Lihat Pasien Antrian Terdepan
5. Lihat Pasien Antrian Paling Belakang
6. Cari Posisi Antrian Pasien
7. Menghapus Semua Data Pasien
8. Exit

3
Daftar Pasien dalam Antrian:
Nama Pasien: Rara, No ID: 345, Jenis Kelamin: p
Nama Pasien: Wisnu, No ID: 567, Jenis Kelamin: 1
Nama Pasien: Siska, No ID: 789, Jenis Kelamin: p
Jumlah pasien: 3

Daftar Menu Klinik:

```

5. Lihat Pasien Antrian Paling Belakang
6. Cari Posisi Antrian Pasien
7. Menghapus Semua Data Pasien
8. Exit
-----
3
Daftar Pasien dalam Antrian:
Nama Pasien: Rara, No ID: 345, Jenis Kelamin: p
Nama Pasien: Wisnu, No ID: 567, Jenis Kelamin: l
Nama Pasien: Siska, No ID: 789, Jenis Kelamin: p
Jumlah pasien: 3
=====
Daftar Menu Klinik:
=====
1. Tambah Pasien
2. Keluarkan Data Pasien
3. Lihat Daftar Antrian
4. Lihat Pasien Antrian Terdepan
5. Lihat Pasien Antrian Paling Belakang
6. Cari Posisi Antrian Pasien
7. Menghapus Semua Data Pasien
8. Exit
-----
6
Masukkan nama pasien yang ingin dicari: wisnu
Posisi pasien: wisnu dalam antrian: 2
ID Pasien: 567
Jenis Kelamin Pasien: l
=====
Daftar Menu Klinik:
=====
1. Tambah Pasien
2. Keluarkan Data Pasien
3. Lihat Daftar Antrian
4. Lihat Pasien Antrian Terdepan
5. Lihat Pasien Antrian Paling Belakang
6. Cari Posisi Antrian Pasien
7. Menghapus Semua Data Pasien
8. Exit
-----

```

```

Jenis Kelamin Pasien: l
=====
Daftar Menu Klinik:
=====
1. Tambah Pasien
2. Keluarkan Data Pasien
3. Lihat Daftar Antrian
4. Lihat Pasien Antrian Terdepan
5. Lihat Pasien Antrian Paling Belakang
6. Cari Posisi Antrian Pasien
7. Menghapus Semua Data Pasien
8. Exit
-----
7
Data Telah Terhapus Seluruhnya
=====
Daftar Menu Klinik:
=====
1. Tambah Pasien
2. Keluarkan Data Pasien
3. Lihat Daftar Antrian
4. Lihat Pasien Antrian Terdepan
5. Lihat Pasien Antrian Paling Belakang
6. Cari Posisi Antrian Pasien
7. Menghapus Semua Data Pasien
8. Exit
-----
8
Program selesai
PS D:\1\Semester 2\ASD PRAKTIKUM\ASD>

```