

LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

JOBSHEET 10



SHAFIQA NABILA MAHARANI KHOIRUNNISA

244107020221

TI – 1B

2.1 Percobaan 1

```
J Queue20.java > Queue20 > Dequeue()
1  public class Queue20 {    Move this file to a named package.
2
3      int[] data;
4      int front;
5      int rear;
6      int size;
7      int max;
8
9      public Queue20(int n) {
10         max = n;
11         data = new int[max];
12         size = 0;
13         front = rear = -1;
14     }
15
16     public boolean isEmpty() {    Rename this method name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
17         if (size == 0) {    Replace this if-then-else statement by a single return statement.
18             return true;
19         } else {
20             return false;
21         }
22     }
23
24     public boolean IsFull() {    Rename this method name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
25         if (size == max) {    Replace this if-then-else statement by a single return statement.
26             return true;
27         } else {
28             return false;
29         }
30     }
31
32     public void peek() {
```

```
J QueueMain20.java > QueueMain20 > main(String[])
1  import java.util.Scanner;
2  public class QueueMain20 {
3      public static void menu() {
4          System.out.println(x:"Masukkan operasi yang diinginkan: ");
5          System.out.println(x:"1. Enqueue");
6          System.out.println(x:"2. Dequeue");
7          System.out.println(x:"3. Print");
8          System.out.println(x:"4. Peek");
9          System.out.println(x:"5. Clear");
10         System.out.println(x:"-----");
11     }
12     Run | Debug
13     public static void main(String[] args) {
14         Scanner sc = new Scanner(System.in);
15         System.out.print(s:"Masukkan kapasitas queue: ");
16         int n = sc.nextInt();
17         Queue20 Q = new Queue20(n);
18
19         int pilih;
20         do {
21             menu();
22             pilih = sc.nextInt();
23             switch (pilih) {
24                 case 1:
25                     System.out.print(s:"Masukkan data baru: ");
26                     int dataMasuk = sc.nextInt();
27                     Q.Enqueue(dataMasuk);
28                     break;
29                 case 2:
30                     int dataKeluar = Q.Dequeue();
31                     if (dataKeluar != 0) {
```

Hasil Run :

```
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
```

```
-----
```

2.1.3. Pertanyaan

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

- Nilai awal front dan rear diatur ke -1 untuk menunjukkan bahwa queue masih kosong. Ini berarti tidak ada elemen yang telah ditambahkan ke dalam queue. Atribut size diatur ke 0 karena tidak ada elemen yang ada di dalam queue pada saat inisialisasi.

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

- Potongan kode dalam method Enqueue bertujuan untuk menambahkan elemen baru ke dalam queue. Jika queue tidak penuh, elemen baru akan ditambahkan pada posisi rear, dan rear akan diperbarui untuk menunjuk ke posisi berikutnya. Jika ini adalah elemen pertama yang ditambahkan, front juga akan diatur ke 0.

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

- Potongan kode dalam method Dequeue digunakan untuk menghapus elemen dari posisi front dalam queue. Jika queue tidak kosong, elemen pada posisi front akan diambil dan front akan diperbarui untuk menunjuk ke elemen berikutnya. Jika queue menjadi kosong setelah penghapusan, front dan rear akan diatur kembali ke -1.

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

- Variabel i dimulai dari front karena kita ingin mencetak elemen-elemen yang ada dalam queue mulai dari posisi depan hingga posisi belakang. Memulai dari 0 tidak akan memberikan hasil yang benar jika queue tidak dimulai dari indeks 0.

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

- Potongan kode dalam method print bertujuan untuk menampilkan semua elemen dalam queue dari posisi front hingga rear. Ini memastikan bahwa semua elemen yang ada dalam queue ditampilkan dengan urutan yang benar.

6. Tunjukkan potongan kode program yang merupakan queue overflow!

- Potongan kode yang menunjukkan queue overflow biasanya terdapat dalam method Enqueue, di mana sebelum menambahkan elemen baru, kita memeriksa apakah queue sudah penuh dengan menggunakan method isFull(). Jika isFull() mengembalikan true, maka itu adalah kondisi overflow.

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!

- Modifikasi dapat dilakukan dengan menambahkan pernyataan `System.exit(0)`; setelah menampilkan pesan kesalahan untuk queue overflow dan underflow. Ini akan menghentikan eksekusi program jika kondisi tersebut terjadi.

2.2. Percobaan 2

```
J Mahasiswa20.java > Mahasiswa20
1 public class Mahasiswa20 { Move this file to a named package.
2
3     String nim;
4     String nama;
5     String prodi;
6     String kelas;
7
8     public Mahasiswa20(String nim, String nama, String prodi, String kelas) {
9         this.nim = nim;
10        this.nama = nama;
11        this.prodi = prodi;
12        this.kelas = kelas;
13    }
14    public void tampilkanData() {
15        System.out.println(nim + " - " + nama + " - " + prodi + " - " + kelas); Replace this use of System.out by a logger.
16    }
17 }
18
19
```

```
J AntrianLayanan20.java > AntrianLayanan20 > tambahAntrian(Mahasiswa20)
1 public class AntrianLayanan20 { Move this file to a named package.
2     Mahasiswa20[] data;
3     int front;
4     int rear;
5     int size;
6     int max;
7
8     public AntrianLayanan20(int max) {
9         this.max = max;
10        this.data = new Mahasiswa20[max];
11        this.front = 0;
12        this.rear = -1;
13        this.size = 0;
14    }
15
16    public boolean isEmpty() { Rename this method name to match the regular expression '^([a-z][a-zA-Z0-9])*$'.
17        if (size == 0) { Replace this if-then-else statement by a single return statement.
18            return true;
19        } else {
20            return false;
21        }
22    }
23
24    public boolean IsFull() { Rename this method name to match the regular expression '^([a-z][a-zA-Z0-9])*$'.
25        if (size == max) { Replace this if-then-else statement by a single return statement.
26            return true;
27        } else {
28            return false;
29        }
30    }
31
32    public void tambahAntrian(Mahasiswa20 mhs) {
```

```
J LayananAkademikSIADKAD.java > LayananAkademikSIADKAD
1 import java.util.Scanner; Move this file to a named package.
2 public class LayananAkademikSIADKAD {
3     Run | Debug
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         AntrianLayanan20 antrian = new AntrianLayanan20(max:5);
7         int pilihan;
8
9         do {
10             System.out.println(x:"\n--- Menu Layanan Akademik ---");
11             System.out.println(x:"1. Tambah Mahasiswa ke Antrian"); Replace this use of System.out by a logger.
12             System.out.println(x:"2. Layani Mahasiswa"); Replace this use of System.out by a logger.
13             System.out.println(x:"3. Lihat Mahasiswa Terdepan"); Replace this use of System.out by a logger.
14             System.out.println(x:"4. Lihat Semua Antrian"); Replace this use of System.out by a logger.
15             System.out.println(x:"5. Jumlah Mahasiswa dalam Antrian"); Replace this use of System.out by a logger.
16             System.out.println(x:"0. Keluar"); Replace this use of System.out by a logger.
17             System.out.print(s:"Pilih Menu: "); Replace this use of System.out by a logger.
18             pilihan = sc.nextInt(); sc.nextLine(); Replace this use of System.out by a logger.
19
20             switch (pilihan) {
21                 case 1:
22                     System.out.print(s:"NIM : ");
23                     String nim = sc.nextLine(); Replace this use of System.out by a logger.
24                     System.out.print(s:"Nama : ");
25                     String nama = sc.nextLine(); Replace this use of System.out by a logger.
26                     System.out.print(s:"Prodi : ");
27                     String prodi = sc.nextLine(); Replace this use of System.out by a logger.
28                     System.out.print(s:"Kelas : ");
29                     String kelas = sc.nextLine(); Replace this use of System.out by a logger.
30                     Mahasiswa20 mhs = new Mahasiswa20(nim, nama, prodi, kelas);
31                     antrian.tambahAntrian(mhs);
32                     break;
33             }
34         } while (pilihan != 0);
35     }
36 }
```

Hasil Run :

```
--- Menu Layanan Akademik ---
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua 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 Layanan Akademik ---
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua 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 Layanan Akademik ---

1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
0. Keluar

Pilih Menu: 4

Daftar Mahasiswa dalam Antrian:

NIM - NAMA - PRODI - KELAS

1. 123 - Aldi - TI - 1A
2. 124 - Bobi - TI - 1G

--- Menu Layanan Akademik ---

1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
0. Keluar

Pilih Menu: 2

Melayani Mahasiswa: 123 - Aldi - TI - 1A

--- Menu Layanan Akademik ---

1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua 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 Layanan Akademik ---
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
0. Keluar
Pilih Menu: 5
Jumlah dalam antrian: 1
```

```
--- Menu Layanan Akademik ---
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
0. Keluar
Pilih Menu: 0
Terima kasih.
```

2.2.3 Pertanyaan

Lakukan modifikasi program dengan menambahkan method baru bernama **LihatAkhir** pada class **AntrianLayanan** yang digunakan untuk mengecek antrian yang berada di posisi belakang. Tambahkan pula daftar menu 6. Cek Antrian paling belakang pada class **LayananAkademikSIAKAD** sehingga method **LihatAkhir** dapat dipanggil!

```
public Mahasiswa20 LihatAkhir() {    Rename this method name to match the regular expression '^[a-z][a-zA-Z0-9]*$'.
    if (IsEmpty()) {
        System.out.println(x:"Antrian kosong!");    Replace this use of System.out by a logger.
        return null;
    }
    return data[rear]; // Mengembalikan elemen di posisi belakang
}
```

```
System.out.println(x:"6. Cek Antrian paling belakang");
```



```

case 6:
    Mahasiswa20 terakhir = antrian.LihatAkhir();    Replace this use of System.out by a logger.
    if (terakhir != null) {
        System.out.println("Mahasiswa di belakang antrian: " + terakhir.nama);
    }    Replace this use of System.out by a logger.
    break;

```

2.3 Tugas

```

J LayananAkademikSiakadTugas.java > LayananAkademikSiakadTugas > main(String[])
1  import java.util.Scanner;    Move this file to a named package.
2
3  public class LayananAkademikSiakadTugas {
4      Run | Debug
5      public static void main(String[] args) {    A "Brain Method" was detected. Refactor it to reduce at least one of
6          Scanner sc = new Scanner(System.in);
7          AntrianLayanan20 antrian = new AntrianLayanan20(max:10);
8          int jumlahKRS = 0;    Remove this unused "jumlahKRS" local variable.
9
10         while (true) {
11             System.out.println(x:"\nMenu:");    Replace this use of System.out by a logger.
12             System.out.println(x:"1. Tambah Antrian");    Replace this use of System.out by a logger.
13             System.out.println(x:"2. Proses KRS (2 Mahasiswa)");    Replace this use of System.out by a logger.
14             System.out.println(x:"3. Tampilkan Semua Antrian");    Replace this use of System.out by a logger.
15             System.out.println(x:"4. Tampilkan 2 Antrian Terdepan");    Replace this use of System.out by a logger.
16             System.out.println(x:"5. Tampilkan Antrian Paling Akhir");    Replace this use of System.out by a logger.
17             System.out.println(x:"6. Cek Jumlah Antrian");    Replace this use of System.out by a logger.
18             System.out.println(x:"7. Kosongkan Antrian");    Replace this use of System.out by a logger.
19             System.out.println(x:"8. Keluar");    Replace this use of System.out by a logger.
20             System.out.print(s:"Pilih menu: ");    Replace this use of System.out by a logger.
21             int pilihan = sc.nextInt();
22             sc.nextLine(); // Clear buffer
23
24             switch (pilihan) {
25                 case 1:
26                     System.out.print(s:"Masukkan NIM: ");    Replace this use of System.out by a logger.
27                     String nim = sc.nextLine();
28                     System.out.print(s:"Masukkan Nama: ");    Replace this use of System.out by a logger.
29                     String nama = sc.nextLine();
30                     System.out.print(s:"Masukkan Prodi: ");    Replace this use of System.out by a logger.
31                     String prodi = sc.nextLine();
32                     System.out.print(s:"Masukkan Kelas: ");    Replace this use of System.out by a logger.

```

```

J MahasiswaTugas.java > MahasiswaTugas > MahasiswaTugas(String, String, String, String)
1  public class MahasiswaTugas {    Move this file to a named package.
2      String nim;
3      String nama;
4      String prodi;
5      String kelas;
6
7      MahasiswaTugas(String nim, String nama, String prodi, String kelas) {
8          this.nim = nim;
9          this.nama = nama;
10         this.prodi = prodi;
11         this.kelas = kelas;
12     }
13
14     void tampilkanData() {
15         System.out.println("NIM: " + nim + ", Nama: " + nama + ", Prodi: " + prodi + ", Kelas: " + kelas);    Replace
16     }
17 }
18

```

```
J AntrianLayananTugas.java > AntrianLayananTugas > clear()
1 public class AntrianLayananTugas { Move this file to a named package.
2     Mahasiswa20[] data;
3     int front, rear, size, max; Declare "rear" and all following declarations on a separate line. [+2 locations]
4
5     AntrianLayananTugas(int n) {
6         max = n;
7         data = new Mahasiswa20[max];
8         front = -1;
9         rear = -1;
10        size = 0;
11    }
12
13    boolean isEmpty() {
14        return size == 0;
15    }
16
17    boolean isFull() {
18        return size == max;
19    }
20
21    void enqueue(Mahasiswa20 mhs) {
22        if (isFull()) {
23            System.out.println(x:"Antrian penuh!"); Replace this use of System.out by a logger.
24            return;
25        }
26        if (isEmpty()) {
27            front = 0;
28        }
29        rear = (rear + 1) % max;
30        data[rear] = mhs;
31        size++;
32    }
}
```

Menu:

1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Cek Jumlah Antrian
7. Kosongkan Antrian
8. Keluar

Pilih menu: 1

Masukkan NIM: 1111

Masukkan Nama: Fika

Masukkan Prodi: TI

Masukkan Kelas: 1B

Menu:

1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Cek Jumlah Antrian
7. Kosongkan Antrian
8. Keluar

Pilih menu: 1

Masukkan NIM: 2222

Masukkan Nama: Shafiqa

Masukkan Prodi: TI

Masukkan Kelas: 1B

Menu:

1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Cek Jumlah Antrian
7. Kosongkan Antrian
8. Keluar

Pilih menu: 1

Masukkan NIM: 3333

Masukkan Nama: Vika

Masukkan Prodi: TI

Masukkan Kelas: 1H

Menu:

1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Cek Jumlah Antrian
7. Kosongkan Antrian
8. Keluar

Pilih menu: 2

Memproses KRS untuk 2 Mahasiswa:

1111 - Fika - TI - 1B

2222 - Shafiqa - TI - 1B

Menu:

1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Cek Jumlah Antrian
7. Kosongkan Antrian
8. Keluar

Pilih menu: 3

Daftar Antrian:

3333 - Vika - TI - 1H

Menu:

1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Cek Jumlah Antrian
7. Kosongkan Antrian
8. Keluar

Pilih menu: 4

2 Antrian Terdepan:

3333 - Vika - TI - 1H

Antrian kosong!

Menu:

1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Cek Jumlah Antrian
7. Kosongkan Antrian
8. Keluar

Pilih menu: 5

Antrian kosong!

Menu:

1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Cek Jumlah Antrian
7. Kosongkan Antrian
8. Keluar

Pilih menu: 6

Jumlah Antrian: 0

Menu:

1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Cek Jumlah Antrian
7. Kosongkan Antrian
8. Keluar

Pilih menu: 7

Antrian telah dikosongkan.

Menu:

1. Tambah Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Cek Jumlah Antrian
7. Kosongkan Antrian
8. Keluar

Pilih menu: 8

Keluar dari program.

PS C:\Users\fika\MATKUL SEMESTER 2\praktikum-ASD\Jobsheet 10> |