

# **ALGORITMA DAN STRUKTUR DATA**

## **“Laporan hasil Praktikum pada Jobsheet 10 “Queue” ”**

Oleh:

Hafiz Rizqi Hernanda

NIM (244107020154)



**Jurusan Teknologi informasi**

**Teknik Informatika**

**Politeknik Negeri Malang**

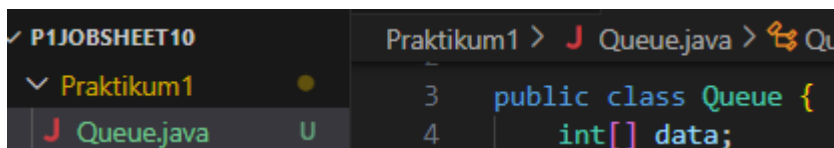
## 2. Praktikum 1 2.1

### Percobaan 1 : Operasi Dasar Queue

Queue
data: int[] front: int rear: int size: int max: int
Queue(n: int) isFull(): boolean isEmpty(): boolean enqueue(dt: int): void dequeue(): int peek: void print(): void clear(): void

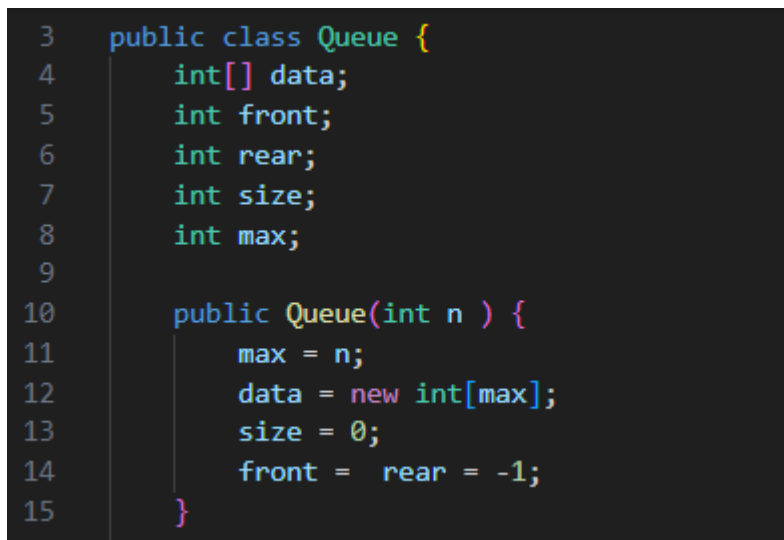
#### 2.1.1. Langkah-langkah Percobaan

1. Buat folder baru bernama P1Jobsheet10 di dalam repository Praktikum ASD, kemudian buat class baru dengan nama Queue.



```
Praktikum1 > J Queue.java > Qu
3 public class Queue {
4     int[] data;
```

2. Tambahkan atribut-atribut Queue sesuai diagram class, kemudian tambahkan pula konstruktornya seperti berikut ini :



```
3 public class Queue {
4     int[] data;
5     int front;
6     int rear;
7     int size;
8     int max;
9
10    public Queue(int n ) {
11        max = n;
12        data = new int[max];
13        size = 0;
14        front = rear = -1;
15    }
16 }
```

3. Buat method isEmpty bertipe boolean yang digunakan untuk mengecek apakah queue kosong.

```

17     public boolean isEmpty() {
18         if (size == 0) {
19             return true;
20         } else {
21             return false;
22         }
23     }

```

4. Buat method IsFull bertipe boolean yang digunakan untuk mengecek apakah queue sudah penuh.

```

25     public boolean isFull() {
26         if (size == max) {
27             return true;
28         } else {
29             return false;
30         }
31     }

```

5. Buat method peek bertipe void untuk menampilkan elemen queue pada posisi paling depan.

```

33     public void peek() {
34         if (!isEmpty()) {
35             System.out.println("Elemen depan: " + data[front]);
36         } else {
37             System.out.println("Queue masih kosong");
38         }
39     }

```

6. Buat method print bertipe void untuk menampilkan seluruh elemen pada queue mulai dari posisi front sampai dengan posisi rear

```

41     public void print() {
42         if (isEmpty()) {
43             System.out.print("Queue masih kosong");
44         } else {
45             int i = front;
46             while (i != rear) {
47                 System.out.print(data[i] + " ");
48                 i = (i + 1) % max;
49             }
50             System.out.println(data[i] + " ");
51             System.out.println("Jumlah elemen = " + size);
52         }
53     }

```

7. Buat method clear bertipe void untuk menghapus semua elemen pada queue

```
55     public void clear() {
56         if (!isEmpty()) {
57             front = rear = -1;
58             size = 0;
59             System.out.println(x:"Queue berhasil dikosongkan");
60         } else {
61             System.out.println(x:"Queue masih kosong");
62         }
63     }
```

8. Buat method Enqueue bertipe void untuk menambahkan isi queue dengan parameter dt yang bertipe integer

```
65     public void enqueue(int dt ) {
66         if (isFull()) {
67             System.out.println(x:"Queue sudah penuh");
68         } else {
69             if (isEmpty()) {
70                 front = rear = 0;
71             } else {
72                 if(rear == max -1) {
73                     rear = 0;
74                 } else {
75                     rear++;
76                 }
77             }
78             data[rear] = dt;
79             size++;
80         }
81     }
```

9. Buat method Dequeue bertipe int untuk mengeluarkan data pada queue di posisi belakang

```

83     public int dequeue() {
84         int dt = 0;
85         if (isEmpty()) {
86             System.out.println(x:"Queue masih kosong");
87         } else {
88             dt = data[front];
89             size--;
90             if(isEmpty()) {
91                 front = rear = -1;
92             } else {
93                 if (front == max -1) {
94                     front = 0;
95                 } else {
96                     front++;
97                 }
98             }
99         }
100     return dt;
101 }

```

10. Selanjutnya, buat class baru dengan nama QueueMain tetap pada package Praktikum1. Buat method menu bertipe void untuk memilih menu program pada saat dijalankan.

```

6  ✓ public static void menu() {
7      System.out.println(x:"Masukkan operasi yang diinginkan:");
8      System.out.println(x:"1. Enqueue");
9      System.out.println(x:"2. Dequeue");
10     System.out.println(x:"3. Print");
11     System.out.println(x:"4. Peek");
12     System.out.println(x:"5. Clear");
13     System.out.println(x:"=====");
14 }

```

11. Buat fungsi main, kemudian deklarasikan Scanner dengan nama sc.

```

1  package Praktikum1;
2
3  import java.util.Scanner;
4  public class QueueMain {
5

```

```

Run | Debug
15     public static void main(String[] args) {
16         Scanner sc = new Scanner(System.in);

```

12. Buat variabel n untuk menampung masukan berupa jumlah maksimal elemen yang dapat disimpan pada queue

```

18     System.out.print(s:"Masukkan kapasitas queue: ");
19     int n = sc.nextInt();

```

13. Lakukan instansiasi objek Queue dengan nama Q dengan mengirimkan parameter n sebagai kapasitas elemen queue

```

21     Queue Q = new Queue(n);

```

14. Deklarasikan variabel dengan nama pilih bertipe integer untuk menampung pilih menu dari pengguna.

```

23     int pilih;

```

15. Lakukan perulangan menggunakan do-while untuk menjalankan program secara terus menerus sesuai masukan yang diberikan. Di dalam perulangan tersebut, terdapat pemilihan kondisi menggunakan switch-case untuk menjalankan operasi queue sesuai dengan masukan pengguna

```

24     do {
25         menu();
26         pilih = sc.nextInt();
27         switch (pilih) {
28             case 1:
29                 System.out.print(s:"Masukkan data baru: ");
30                 int dataMasuk = sc.nextInt();
31                 Q.enqueue(dataMasuk);
32                 break;
33             case 2:
34                 int dataKeluar = Q.dequeue();
35                 if (dataKeluar != 0) {
36                     System.out.println("Data yang dikeluarkan: " + dataKeluar);
37                     break;
38                 }
39             case 3:
40                 Q.print();
41                 break;
42             case 4:
43                 Q.peek();
44                 break;
45             case 5:
46                 Q.clear();
47                 break;
48         }
49     } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5);
50 }
51
52

```

16. Compile dan jalankan class QueueMain, kemudian amati hasilnya

### 2.1.2. Verifikasi Hasil Percobaan

Samakan hasil compile kode program Anda dengan gambar berikut ini

```
Output - Queue (run) x
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
```

```

PS C:\Regulus\Praktikum-ASD\P1Jobsheet10> & 'C:\V
e' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp
g\Code\User\workspaceStorage\9fb65d7e2ef473e3518a
sheet10_cd020668\bin' 'Praktikum1.QueueMain'
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 depan: 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?

Untuk front dan rear itu mengambil sebuah indeks di arraynya karena indeks itu dimulai dari 0 sedangkan untuk size itu mengambil sebuah nilai dari berapa banyak data di array yang pasti dimulai dari 1



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

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

Kegunaan nya adalah apakah rear itu berada di posisi akhir queue jika iya maka rear = 0 yang berarti rear akan kembali ke indeks 0, kode ini berguna agar bisa berputar kembali untuk mengisi data di dalam queue

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

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

If (front == max -1) mengecek apakah front itu berada di posisi akhir queue jika iya maka front = 0 yang berarti front akan kembali ke indeks 0, kode ini berguna agar bisa berputar kembali dalam menghapus data di dalam queue

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

Karena front itu tidak pasti berada di indeks 0, jika kita menghapus data(dequeue) pasti indeks front nya berubah, tidak pasti itu di indeks 0.

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

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

Ini sebuah iterasi, kegunaan kode itu agar jika kita mengisi data di queue jika sudah berada di indeks terakhir dan misal di indeks awal nya ada yang kosong maka dapat di isi dengan data.

6. Tunjukkan potongan kode program yang merupakan queue overflow!

```
65 public void enqueue(int dt ) {  
66     if (isFull()) {  
67         System.out.println(x:"Queue sudah penuh");
```

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 pada method enqueue

```
65     public void enqueue(int dt ) {  
66         if (isFull()) {  
67             System.out.println(x:"Queue sudah penuh! Program dihentikan");  
68             System.exit(status:0);  
        }
```

```
PS C:\Regulus\Praktikum-ASD\Jobsheet10> c:; cd 'c:\Regulus\Prakt
ata\Roaming\Code\User\workspaceStorage\14caa275eb597ef3281550a5cc
Masukkan kapasitas queue: 4
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
=====
1
Masukkan data baru: 10
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
=====
1
Masukkan data baru: 20
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: 16
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
=====
1
Masukkan data baru: 25
Queue sudah penuh! Program dihentikan
```

Modifikasi pada method dequeue

```
84     public int dequeue() {
85         int dt = 0;
86         if (isEmpty()) {
87             System.out.println(x:"Queue masih kosong! Program dihentikan");
88             System.exit(status:0);
89         }
```

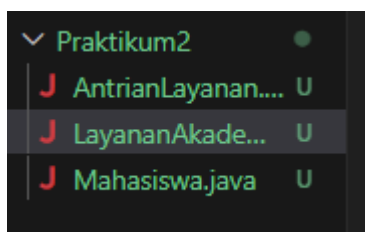
```
PS C:\Regulus\Praktikum-ASD\Jobsheet10> & '5eb597ef3281550a5cc42dce1\redhat.java\jdt_ws
Masukkan kapasitas queue: 4
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
=====
2
Queue masih kosong! Program dihentikan
PS C:\Regulus\Praktikum-ASD\Jobsheet10> ^C
```

## 2.2. Percobaan 2 : Antrian Layanan Akademik

Mahasiswa
nim:String
nama: String
prodi: String
kelas: String
Mahasiswa(nim: String, nama: String, prodi: String, kelas: String)
void tampilkanData()

### 2.2.1. Langkah-langkah Percobaan

1. Buat folder baru bernama P2Jobsheet10 di dalam repository Praktikum ASD, kemudian buat class baru dengan nama Mahasiswa.



2. Tambahkan atribut-atribut Nasabah seperti pada Class Diagram, kemudian tambahkan pula konstruktornya seperti gambar berikut ini.

```

1  package Praktikum2;
2
3  public class Mahasiswa {
4      String nim;
5      String nama;
6      String prodi;
7      String kelas;
8
9      public Mahasiswa(String nim, String nama, String prodi, String kelas)
10         this.nim = nim;
11         this.nama = nama;
12         this.prodi = prodi;
13         this.kelas = kelas;
14     }
15

```

Dan tambahkan method tampilkanData berikut :

```

16     public void tampilkanData() {
17         System.out.println(nim + " - " + nama + " - " + prodi + " - " + kelas);
18     }
19 }

```

3. Salin kode program class Queue pada Praktikum 1 untuk digunakan kembali pada Praktikum 2 ini, ganti nama class-nya dengan AntrianLayanan. Karena pada Praktikum 1, data yang disimpan pada queue hanya berupa array bertipe integer, sedangkan pada Praktikum 2 data yang digunakan adalah object, maka perlu dilakukan modifikasi pada class AntrianLayanan tersebut.

```

Praktikum2 > J AntrianLayanan.java > AntrianLayanan > max
1  package Praktikum2;
2
3  public class AntrianLayanan {
4      Mahasiswa[] data;
5      int front;
6      int rear;
7      int size;
8      int max;
9
10
11
12     public AntrianLayanan(int max) {
13         this.max = max;
14         this.data = new Mahasiswa[max];
15         this.front = 0;
16         this.rear = -1;
17         this.size = 0;
18     }
19

```

4. Lakukan modifikasi pada class AntrianLayanan dengan mengubah tipe int[] data menjadi Mahasiswa[] data karena pada kasus ini data yang akan disimpan berupa object Mahasiswa. Modifikasi perlu dilakukan pada atribut, method Enqueue, dan method Dequeue.

```
74     public void tambahkanAntrian(Mahasiswa mhs) {
75         if (isFull()) {
76             System.out.println(x:"Antrian penuh, tidak bisa menambahkan mahasiswa");
77             return;
78         }
79         rear = (rear + 1) % max;
80         data[rear] = mhs;
81         size++;
82         System.out.println(mhs.nama + " berhasil masuk ke antrian");
83     }
84
85     public Mahasiswa layaniMahasiswa() {
86         if (isEmpty()) {
87             System.out.println(x:"Antrian kosong");
88             return null;
89         }
90         Mahasiswa mhs = data[front];
91         front = (front + 1) % max;
92         size--;
93         return mhs;
94     }
95 }
```

5. Berikutnya method peek dan print yaitu untuk menampilkan data antrian layanan paling depan dan menampilkan semua data antrian layanan.

```
36     public void lihatTerdepan() {
37         if (isEmpty()) {
38             System.out.println(x:"Antrian kosong");
39         } else {
40             System.out.print(s:"Mahasiswa terdepan: ");
41             System.out.println(x:"NIM - NAMA - PRODI - KELAS");
42             data[front].tampilkanData();
43         }
44     }
45
46     public void tampilkanSemua() {
47         if (isEmpty()) {
48             System.out.print(s:"Antrian kosong");
49             return;
50         }
51         System.out.println(x:"Daftar Mahasiswa dalam antrian:");
52         System.out.println(x:"NIM - NAMA - PRODI - KELAS");
53         for (int i = 0; i < size; i++) {
54             int index = (front + i) % max;
55             System.out.print((i + 1) + ". ");
56             data[index].tampilkanData();
57         }
58     }
59 }
```

Ditambahkan dengan method getJumlahAntrian yaitu menampilkan nilai size

```

60     public int getJumlahAntrian() {
61         return size;
62     }

```

6. Selanjutnya, buat class baru dengan nama LayananAkademikSIKAD tetap pada package yang sama. Buat fungsi main, deklarasikan Scanner dengan nama sc.

```

Praktikum2 > J LayananAkademikSIKAD.java > LayananAkad
1  package Praktikum2;
2  import java.util.Scanner;
3  public class LayananAkademikSIKAD {

```

7. Kemudian lakukan instansiasi objek AntrianLayanan dengan nama antrian dan nilai parameter nya adalah nilai maksimal antrian yang ditentukan (misal sama dengan 5).

8. Deklarasikan variabel dengan nama pilihan bertipe integer untuk menampung pilih menu dari pengguna.

```

3  public class LayananAkademikSIKAD {
    Run | Debug
4  public static void main(String[] args) {
5      Scanner sc = new Scanner(System.in);
6      AntrianLayanan antrian = new AntrianLayanan(max:5);
7      int pilihan;
8

```

9. Tambahkan kode berikut untuk melakukan perulangan menu sesuai dengan masukan yang diberikan oleh pengguna.

```

9      do {
10         System.out.println(x:"=== Menu Antrian Layanan Akademik ===");
11         System.out.println(x:"1. Tambah Mahasiswa ke Antrian");
12         System.out.println(x:"2. Layani Mahasiswa");
13         System.out.println(x:"3. Lihat Mahasiswa Terdepan");
14         System.out.println(x:"4. Lihat Semua antrian");
15         System.out.println(x:"5. Jumlah Mahasiswa dalam antrian");
16         System.out.println(x:"0. Keluar");
17         System.out.print(s:"Pilih menu: ");
18         pilihan = sc.nextInt(); sc.nextLine();
19
20         switch (pilihan) {
21             case 1:
22                 System.out.print(s:"NIM   :");
23                 String nim = sc.nextLine();
24                 System.out.print(s:"Nama   :");
25                 String nama = sc.nextLine();
26                 System.out.print(s:"Prodi  :");
27                 String prodi = sc.nextLine();
28                 System.out.print(s:"Kelas :");
29                 String kelas = sc.nextLine();
30                 Mahasiswa mhs = new Mahasiswa(nim, nama, prodi, kelas);
31                 antrian.tambahkanAntrian(mhs);
32                 break;
33             case 2:
34                 Mahasiswa dilayani = antrian.layaniMahasiswa();
35                 if (dilayani != null) {
36                     System.out.print(s:"Melayani Mahasiswa: ");
37                     dilayani.tampilkanData();
38                 }
39                 break;
40             case 3:
41                 antrian.lihatTerdepan();
42                 break;
43             case 4:
44                 antrian.tampilkanSemua();
45                 break;
46             case 5:
47                 System.out.println("Jumlah dalam antrian: " + antrian.getJumlahAntrian());
48                 break;
49             case 0:
50                 System.out.println(x:"Terima kasih");
51                 break;
52             default:
53                 System.out.println(x:"Pilihan tidak valid.");
54         }
55     } while (pilihan != 0);
56
57     sc.close();
58 }
59 }

```

10. Compile dan jalankan class LayananAkademikSIKAD, kemudian amati hasilnya.

### 2.2.2 Verifikasi Hasil Percobaan

Samakan hasil compile kode program Anda dengan gambar berikut ini.



=== Menu Antrian 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 Antrian 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 Antrian 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 Antrian 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 Antrian 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 Antrian 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 Antrian 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.

```
=== Menu Antrian 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 Antrian 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 Antrian 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 Antrian 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 Antrian 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 Antrian 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 Antrian 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
PS C:\Regulus\Praktikum-ASD\Jobsheet10> 

```

### 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 LayananAkademikSIKAD sehingga method LihatAkhir dapat dipanggil!

```

96     public void lihatAkhir() {
97         if (isEmpty()) {
98             System.out.println(x:"Antrian kosong");
99         } else {
100             System.out.print(s:"Antrian paling belakang: ");
101             System.out.println(x:"NIM - NAMA - PRODI - KELAS");
102             data[rear].tampilkanData();
103         }
104     }
105 }
106
107

```

```

9         do {
10             System.out.println(x:"=== Menu Antrian Layanan Akademik ===");
11             System.out.println(x:"1. Tambah Mahasiswa ke Antrian");
12             System.out.println(x:"2. Layani Mahasiswa");
13             System.out.println(x:"3. Lihat Mahasiswa Terdepan");
14             System.out.println(x:"4. Lihat Semua antrian");
15             System.out.println(x:"5. Jumlah Mahasiswa dalam antrian");
16             System.out.println(x:"6. Cek Antrian paling belakang");
17             System.out.println(x:"0. Keluar");
18             System.out.print(s:"Pilih menu: ");
19             pilihan = sc.nextInt(); sc.nextLine();

```

```

49             break;
50         case 6:
51             antrian.lihatAkhir();
52             break;

```

```

=== Menu Antrian Layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua antrian
5. Jumlah Mahasiswa dalam antrian
6. Cek Antrian paling belakang
0. Keluar
Pilih menu: 1
NIM : 123
Nama : Aldi
Prodi : TI
Kelas : 1A
Aldi berhasil masuk ke antrian
=== Menu Antrian Layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua antrian
5. Jumlah Mahasiswa dalam antrian
6. Cek Antrian paling belakang
0. Keluar
Pilih menu: 1
NIM : 124
Nama : Bobi
Prodi : TI
Kelas : 1G
Bobi berhasil masuk ke antrian
=== Menu Antrian Layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua antrian
5. Jumlah Mahasiswa dalam antrian
6. Cek Antrian paling belakang
0. Keluar
Pilih menu: 3
Mahasiswa terdepan: NIM - NAMA - PRODI - KELAS
123 - Aldi - TI - 1A
=== Menu Antrian Layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua antrian
5. Jumlah Mahasiswa dalam antrian
6. Cek Antrian paling belakang
0. Keluar
Pilih menu: 6
Antrian paling belakang: NIM - NAMA - PRODI - KELAS
124 - Bobi - TI - 1G

```

## 2.3 Tugas

Waktu : 120 Menit

Buatlah program antrian untuk mengilustrasikan antrian persetujuan Kartu Rencana Studi (KRS) Mahasiswa oleh Dosen Pembina Akademik (DPA). Ketika seorang mahasiswa akan mengantri, maka dia harus mendaftarkan datanya (data mahasiswa

seperti pada praktikum 2). Gunakan class untuk antrian seperti pada Praktikum 1 dan 2, dengan method-method yang berfungsi :

- Cek antrian kosong, Cek antrian penuh, Mengosongkan antrian.
- Menambahkan antrian, Memanggil antrian untuk proses KRS – setiap 1x panggilan terdiri dari 2 mahasiswa (pada antrian no 1 dan 2)
- Menampilkan semua antrian, Menampilkan 2 antrian terdepan, Menampilkan antrian paling akhir.
- Cetak jumlah antrian, Cetak jumlah yang sudah melakukan proses KRS
- Jumlah antrian maximal 10, jumlah yang ditangani masing-masing DPA 30 mahasiswa, cetak jumlah mahasiswa yang belum melakukan proses KRS. Gambarkan Diagram Class untuk antriannya. Implementasikan semua method menggunakan menu pilihan pada fungsi main.

Mahasiswa
nim:String nama: String prodi: String kelas: String
Mahasiswa(nim: String, nama: String, prodi: String, kelas: String) void tampilkanData()

PersetujuanKRS
Data: Mahasiswa[]  Size:int  Front:int  Rear:int  Max:int  sudahDilayani=0:int  totalDilayani=0:int
PersetujuanKRS(int n)  isEmpty():boolean  isFull():boolean



tambahkanAntrian(Mahasiswa mhs):void

Mahasiswa prosesKRS()

clear():void

tampilkanSemua():void

lihatDepan():void

lihatBelakang():void

lihat2Terdepan():void

jumlahAntrian():int

jumlahSudahDilayani():int

jumlahBelumDilayani():int

```

1  package Tugas;
2
3  public class PersetujuanKRS {
4
5      Mahasiswa[] data;
6      int front;
7      int rear;
8      int size;
9      int max;
10     int sudahDilayani = 0;
11
12     public PersetujuanKRS(int n) {
13         max = n;
14         data = new Mahasiswa[max];
15         front = 0;
16         rear = -1;
17         size = 0;
18     }
19
20     public boolean isEmpty() {
21         return size == 0;
22     }
23
24     public boolean isFull() {
25         return size == max;
26     }
27
28     public void tambahkanAntrian(Mahasiswa mhs) {
29         if (isFull()) {
30             System.out.println(x:"Antrian penuh!");
31             return;
32         }
33         rear = (rear + 1) % max;
34         data[rear] = mhs;
35         size++;
36         System.out.println(mhs.nama + " berhasil masuk ke antrian");
37     }
38

```

```

39     int totalDilayani = 0;
40     public Mahasiswa prosesKRS() {
41         System.out.println(x:"Memproses 2 mahasiswa sekaligus:");
42
43         for (int i = 0; i < 2; i++) {
44             if (totalDilayani >= 30) {
45                 System.out.println(x:"DPA sudah menangani maksimal 30 mahasiswa. Proses dihentikan.");
46                 break;
47             }
48
49             if (isEmpty()) {
50                 System.out.println(x:"Antrian kosong!");
51                 break;
52             }
53
54             Mahasiswa mhs = data[front];
55             front = (front + 1) % max;
56             size--;
57             System.out.println("Mahasiswa diproses: " + mhs.nama);
58             sudahDilayani++;
59             totalDilayani++;
60         }
61         return null;
62     }
63 }
64
65     public void clear() {
66         front = rear = -1;
67         size = 0;
68     }
69
70     public void tampilkanSemua() {
71         if (isEmpty()) {
72             System.out.print(s:"Antrian kosong");
73             return;
74         }
75         System.out.println(x:"Daftar Mahasiswa dalam antrian:");
76         System.out.println(x:"NIM - NAMA - PRODI - KELAS");
77         for (int i = 0; i < size; i++) {
78             int index = (front + i) % max;
79             System.out.print((i + 1) + ". ");
80             data[index].tampilkanData();
81         }
82     }

```

```

83     public void lihatDepan() {
84         if (isEmpty()) {
85             System.out.println(x:"Antrian kosong");
86         } else {
87             System.out.print(s:"Mahasiswa terdepan: ");
88             System.out.println(x:"NIM - NAMA - PRODI - KELAS");
89             data[front].tampilkanData();
90         }
91     }
92
93     public void lihatBelakang() {
94         if (isEmpty()) {
95             System.out.println(x:"Antrian kosong");
96         } else {
97             System.out.print(s:"Antrian paling belakang: ");
98             System.out.println(x:"NIM - NAMA - PRODI - KELAS");
99             data[rear].tampilkanData();
100         }
101     }
102
103     public void lihat2Terdepan() {
104         if (isEmpty()) {
105             System.out.println(x:"Antrian kosong");
106             return;
107         }
108         System.out.println(x:"2 Terdepan:");
109         for (int i = 0; i < 2 && i < size; i++) {
110             int index = (front + i) % max;
111             System.out.print((i + 1) + ". ");
112             data[index].tampilkanData();
113         }
114     }
115
116     public int jumlahAntrian() {
117         return size;
118     }
119
120     public int jumlahSudahDilayani() {
121         return sudahDilayani;
122     }
123
124     public int jumlahBelumDilayani() {
125         return size;
126     }
127
128 }

```

Tugas > J MainKRS.java > MainKRS > main(String[])

```
1 package Tugas;
2 import java.util.Scanner;
3 public class MainKRS {
    Run | Debug
4     public static void main(String[] args) {
5         PeretujuanKRS antrian = new PeretujuanKRS(n:10);
6         Scanner sc = new Scanner(System.in);
7
8         int pilih;
9         do {
10             System.out.println(x:"\n=== Menu Antrian KRS ===");
11             System.out.println(x:"1. Tambah Antrian");
12             System.out.println(x:"2. Proses KRS (panggil antrian)");
13             System.out.println(x:"3. Tampilkan Semua Antrian");
14             System.out.println(x:"4. Tampilkan 2 Terdepan");
15             System.out.println(x:"5. Tampilkan Paling Belakang");
16             System.out.println(x:"6. Cek Jumlah Antrian");
17             System.out.println(x:"7. Jumlah Sudah Dilayani");
18             System.out.println(x:"8. Jumlah Belum Dilayani");
19             System.out.println(x:"9. Kosongkan Antrian");
20             System.out.println(x:"0. Keluar");
21             System.out.print(s:"Pilihan: ");
22             pilih = sc.nextInt(); sc.nextLine();

```

```

24     switch (pilih) {
25         case 1:
26             System.out.print(s:"NIM: ");
27             String nim = sc.nextLine();
28             System.out.print(s:"Nama: ");
29             String nama = sc.nextLine();
30             System.out.print(s:"Prodi: ");
31             String prodi = sc.nextLine();
32             System.out.print(s:"Kelas: ");
33             String kelas = sc.nextLine();
34             Mahasiswa mhs = (new Mahasiswa(nim, nama, prodi, kelas));
35             antrian.tambahkanAntrian(mhs);
36             break;
37         case 2:
38             Mahasiswa proses = antrian.prosesKRS();
39             if ( proses != null) {
40                 System.out.println("Dilayani: " + proses);
41             }
42             break;
43         case 3:
44             antrian.tampilkanSemua();
45             break;
46         case 4:
47             antrian.lihat2Terdepan();
48             break;
49         case 5:
50             antrian.lihatBelakang();
51             break;
52         case 6:
53             System.out.println("Jumlah dalam antrian: " + antrian.jumlahAntrian());
54             break;
55         case 7:
56             System.out.println("Jumlah sudah dilayani: " + antrian.jumlahSudahDilayani());
57             break;
58         case 8:
59             System.out.println("Jumlah belum dilayani: " + antrian.jumlahBelumDilayani());
60             break;
61         case 9:
62             antrian.clear();
63             System.out.println(x:"Antrian dikosongkan.");
64             break;
65         case 0:
66             System.out.println(x:"Terima kasih!");
67             break;
68         default:
69             System.out.println(x:"Pilihan tidak valid!");
70     }
71 } while (pilih != 0);
72 }
73 }

```

=== Menu Antrian KRS ===

1. Tambah Antrian
2. Proses KRS (panggil antrian)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Paling Belakang
6. Cek Jumlah
7. Jumlah Sudah Dilayani
8. Jumlah Belum Dilayani
9. Kosongkan Antrian
0. Keluar

Pilihan: 1

NIM: 123

Nama: Aldi

Prodi: TI

Kelas: 1A

=== Menu Antrian KRS ===

1. Tambah Antrian
2. Proses KRS (panggil antrian)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Paling Belakang
6. Cek Jumlah
7. Jumlah Sudah Dilayani
8. Jumlah Belum Dilayani
9. Kosongkan Antrian
0. Keluar

Pilihan: 1

NIM: 124

Nama: Bobi

Prodi: TI

Kelas: 1G

=== Menu Antrian KRS ===

1. Tambah Antrian
2. Proses KRS (panggil antrian)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Paling Belakang
6. Cek Jumlah
7. Jumlah Sudah Dilayani
8. Jumlah Belum Dilayani
9. Kosongkan Antrian
0. Keluar

Pilihan: 1

NIM: 125

Nama: Arif

Prodi: SIB

Kelas: 1B

=== Menu Antrian KRS ===

1. Tambah Antrian
2. Proses KRS (panggil antrian)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Paling Belakang
6. Cek Jumlah
7. Jumlah Sudah Dilayani
8. Jumlah Belum Dilayani
9. Kosongkan Antrian
0. Keluar

Pilihan: 3

Daftar Mahasiswa dalam antrian:

NIM - NAMA - PRODI - KELAS

1. 123 - Aldi - TI - 1A
2. 124 - Bobi - TI - 1G
3. 125 - Arif - SIB - 1B

=== Menu Antrian KRS ===

1. Tambah Antrian
2. Proses KRS (panggil antrian)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Paling Belakang
6. Cek Jumlah
7. Jumlah Sudah Dilayani
8. Jumlah Belum Dilayani
9. Kosongkan Antrian
0. Keluar

Pilihan: 4

2 Terdepan:

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

=== Menu Antrian KRS ===

1. Tambah Antrian
2. Proses KRS (panggil antrian)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Paling Belakang
6. Cek Jumlah
7. Jumlah Sudah Dilayani
8. Jumlah Belum Dilayani
9. Kosongkan Antrian
0. Keluar

Pilihan: 5

Antrian paling belakang: NIM - NAMA - PRODI - KELAS

125 - Arif - SIB - 1B



```
=== Menu Antrian KRS ===
1. Tambah Antrian
2. Proses KRS (panggil antrian)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Paling Belakang
6. Cek Jumlah
7. Jumlah Sudah Dilayani
8. Jumlah Belum Dilayani
9. Kosongkan Antrian
0. Keluar
Pilihan: 6
Jumlah dalam antrian: 3
```

```
=== Menu Antrian KRS ===
1. Tambah Antrian
2. Proses KRS (panggil antrian)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Paling Belakang
6. Cek Jumlah
7. Jumlah Sudah Dilayani
8. Jumlah Belum Dilayani
9. Kosongkan Antrian
0. Keluar
Pilihan: 2
Memproses 2 mahasiswa sekaligus:
Mahasiswa diproses: Aldi
Mahasiswa diproses: Bobi
```

```
=== Menu Antrian KRS ===
1. Tambah Antrian
2. Proses KRS (panggil antrian)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Paling Belakang
6. Cek Jumlah
7. Jumlah Sudah Dilayani
8. Jumlah Belum Dilayani
9. Kosongkan Antrian
0. Keluar
Pilihan: 6
Jumlah dalam antrian: 1
```

```
=== Menu Antrian KRS ===
1. Tambah Antrian
2. Proses KRS (panggil antrian)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Paling Belakang
6. Cek Jumlah
7. Jumlah Sudah Dilayani
8. Jumlah Belum Dilayani
9. Kosongkan Antrian
0. Keluar
Pilihan: 7
Jumlah sudah dilayani: 2
```

```
=== Menu Antrian KRS ===
1. Tambah Antrian
2. Proses KRS (panggil antrian)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Terdepan
5. Tampilkan Paling Belakang
6. Cek Jumlah
7. Jumlah Sudah Dilayani
8. Jumlah Belum Dilayani
9. Kosongkan Antrian
0. Keluar
Pilihan: 8
Jumlah belum dilayani: 1
```