

LAPORAN PRAKTIKUM

MATA KULIAH ALGORITMA DAN STRUKTUR DATA

Dosen Pengampu : Triana Fatmawati, S.T, M.T

PERTEMUAN - 10 - Queue



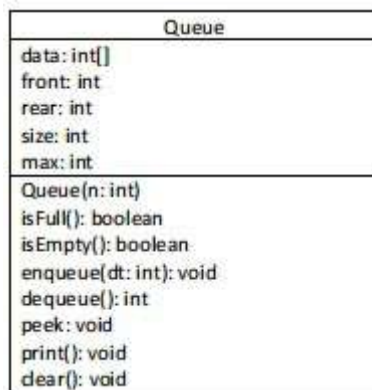
Nama : M. Zidna Billah Faza
NIM : 2341760030
Prodi : D-IV Sistem Informasi Bisnis

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2024

Percobaan 1

- 1) Perhatikan Diagram Class Queue berikut ini:



- 2) Buat package dengan nama Praktikum1, kemudian buat class baru dengan nama Queue.



- 3) Tambahkan atribut-atribut Queue sesuai diagram class, kemudian tambahkan pula konstruktornya seperti gambar berikut ini.

```
public class Queue_18 {  
  
    int[] data;  
    int front;  
    int rear;  
    int size;  
    int max;  
  
    public Queue_18(int n) {  
        max = n;  
        data = new int[max];  
        size = 0;  
        front = rear = -1;  
    }  
}
```

- 4) Buat method IsEmpty bertipe boolean yang digunakan untuk mengecek apakah queue kosong.

```
public boolean IsEmpty() {  
    if (size == 0) {  
        return true;  
    } else {  
        return false;  
    }  
}
```

- 5) Buat method IsFull bertipe boolean yang digunakan untuk mengecek apakah queue sudah penuh.

```
public boolean IsFull() {  
    if (size == max) {  
        return true;  
    } else {  
        return false;  
    }  
}
```

- 6) Buat method peek bertipe void untuk menampilkan elemen queue pada posisi paling depan.

```
public void peek() {  
    if (!IsEmpty()) {  
        System.out.println("Elemen terdepan : " + data[front]);  
    } else {  
        System.out.println(x:"Queue masih kosong");  
    }  
}
```

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

```
public void print() {  
    if (IsEmpty()) {  
        System.out.println(x:"Queue masih kosong");  
    } else {  
        System.out.println(x:"-----");  
        int i = front;  
        while (i != rear) {  
            System.out.println(data[i] + " ");  
            i = (i + 1) % max;  
        }  
        System.out.println(data[i] + " ");  
        System.out.println(x:"-----");  
        System.out.println("Jumlah elemen = " + size);  
    }  
}
```

- 8) Buat method clear bertipe void untuk menghapus semua elemen pada queue

```
public void clear() {  
    if (!IsEmpty()) {  
        front = rear = -1;  
        size = 0;  
        System.out.println(x:"Queue berhasil dikosongkan");  
    } else {  
        System.out.println(x:"Queue masih kosong");  
    }  
}
```

- 9) Buat method Enqueue bertipe void untuk menambahkan isi queue dengan parameter dt yang bertipe integer

```
public void Enqueue(int dt) {  
    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++;  
    }  
}
```

- 10) Buat method Dequeue bertipe int untuk mengeluarkan data pada queue di posisi paling depan

```
public int Dequeue() {  
    int dt = 0;  
    if (IsEmpty()) {  
        System.out.println(x:"Queue masih kosong");  
    } else {  
        dt = data[front];  
        size--;  
        if (IsEmpty()) {  
            front = rear = -1;  
        } else {  
            if (front == max - 1) {  
                front = 0;  
            } else {  
                front++;  
            }  
        }  
    }  
    return dt;  
}
```

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

```
public static void menu() {  
    System.out.println(x: "\nMasukkan operasi yang diinginkan");  
    System.out.println(x: "1. Enqueue");  
    System.out.println(x: "2. Dequeue");  
    System.out.println(x: "3. Print");  
    System.out.println(x: "4. Peek");  
    System.out.println(x: "5. Clear");  
    System.out.println(x: "-----");  
}
```

- 12) Buat fungsi main, kemudian deklarasikan Scanner dengan nama input18.

```
Scanner input18 = new Scanner(System.in);
```

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

```
System.out.print(s: "Masukkan kapasitas queue : ");  
int n = input18.nextInt();
```

- 14) Lakukan instansiasi objek Queue dengan nama Q dengan mengirimkan parameter n sebagai kapasitas elemen queue

```
Queue_18 Q = new Queue_18(n);
```

- 15) Deklarasikan variabel dengan nama pilih bertipe integer untuk menampung pilih menu dari pengguna.

```
int pilihan = 0;
```

- 16) 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.

```
do {  
    menu();  
    pilihan = input18.nextInt();  
    switch (pilihan) {  
        case 1:  
            System.out.print(s:"Masukkan data baru : ");  
            int dataMasuk = input18.nextInt();  
            Q.Enqueue(dataMasuk);  
            break;  
        case 2:  
            int dataKeluar = Q.Dequeue();  
            if (dataKeluar != 0) {  
                System.out.println("Data yang dikeluarkan : " + dataKeluar);  
            }  
            break;  
        case 3:  
            Q.print();  
            break;  
        case 4:  
            Q.peek();  
            break;  
        case 5:  
            Q.clear();  
            break;  
        default:  
            break;  
    }  
} while (pilihan != 6);
```

17) Compile dan jalankan class QueueMain, kemudian amati hasilnya dan verifikasi.

```
Masukkan kapasitas queue : 6

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 : 23

Masukkan operasi yang diinginkan
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
3
-----
15
23
-----
Jumlah elemen = 2

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 dikeluarkan : 15

Masukkan operasi yang diinginkan
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
3
-----
23
-----
Jumlah elemen = 1
```


Pertanyaan Percobaan 1

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

Karena pada front dan rear ketika diberikan nilai awal -1 menandakan bahwa antrian kosong atau tidak ada elem dalam antrian, sementara pada size diberikan nilai awal 0 karena pada awal program tidak ada antrian yang ada didalamnya.

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

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

Potongan kode tersebut bertujuan untuk menangani situasi ketika variabel 'rear' mencapai batas maksimalnya yang telah ditentukan oleh 'max - 1'. Ketika kondisi 'rear == max - 1' terpenuhi maka elemen terakhir dari array sudah diisi dan apabila akan menambahkan elemen baru maka dimulai dari indeks 0 lagi

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

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

Potongan kode tersebut bertujuan untuk menangani situasi ketika variabel 'front' mencapai batas maksimal antrian. Ketika kondisi 'rear == max - 1' terpenuhi maka elemen terakhir dalam array telah diakses dan akan dikeluarkan dari antrian namun apabila kita ingin menggunakan array pada indeks awal jika memungkinkan program akan 'front' pada indeks awal yaitu 0

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

Karena program 'print' digunakan untuk menengeprint elemen dari awal antrian bukan awal indeks karena pada queue urutan antrian tidak mesti dari indeks ke 0

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

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

Potongan kode 'i = (i + 1) % max;' digunakan untuk memperbarui nilai 'i' secara bertahap dari 'front' ke 'rear' dalam array antrian. Operasi modulus (%) memastikan bahwa jika 'i' mencapai batas maksimum 'max', ia akan kembali ke 0, sehingga perulangan dapat berlanjut secara siklikal melalui array antrian.

6) Tunjukkan potongan kode program yang merupakan queue overflow!

```
public boolean IsFull() {  
    if (size == max) {  
        return true;  
    } else {  
        return false;  
    }  
}  
  
public void Enqueue(int dt) {  
    if (IsFull()) {  
        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 dilakukan dengan menambahkan boolean dapat dilihat pada github

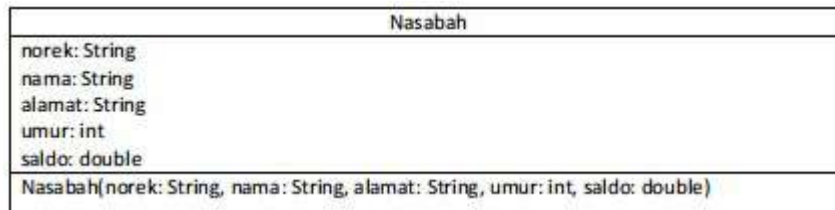
```
public Queue_18(int n) {  
    max = n;  
    data = new int[max];  
    size = 0;  
    front = rear = -1;  
    stopped = false;  
}  
  
System.out.println(x:"Queue masih kosong");  
stopProgram();
```

```
private void stopProgram() {  
    stopped = true;  
}  
  
public boolean isStopped() {  
    return stopped;  
}  
  
Q.print();  
checkProgram(Q);  
break;
```

```
public static void checkProgram(Queue_18 Q) {  
    if (Q.isStopped()) {  
        System.out.println(x:"Program dihentikan karena queue overflow atau queue underflow!");  
    }  
}
```

Percobaan 2

- 1) Perhatikan Diagram Class berikut ini:



- 2) Buat package dengan nama Praktikum2, kemudian buat class baru dengan nama Nasabah.

 Nasabah_18.java

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

```
public class Nasabah_18 {

    String norek;
    String nama;
    String alamat;
    int umur;
    double saldo;

    Nasabah_18(String norek, String nama, String alamat, int umur, double saldo) {
        this.norek = norek;
        this.nama = nama;
        this.alamat = alamat;
        this.umur = umur;
        this.saldo = saldo;
    }
}
```

- 4) Salin kode program class Queue pada Praktikum 1 untuk digunakan kembali pada Praktikum 2 ini. 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 Queue tersebut.

- 5) Lakukan modifikasi pada class Queue dengan mengubah tipe int[] data menjadi Nasabah[] data karena pada kasus ini data yang akan disimpan pada queue berupa object Nasabah. Modifikasi perlu dilakukan pada atribut, method Enqueue, dan method Dequeue. Baris program Nasabah dt = new Nasabah(); akan ditandai sebagai error, untuk mengatasinya, tambahkan konstruktor default di dalam class Nasabah.

```
public class QueueNasabah_18 {  
  
    Nasabah_18[] data;  
    int front;  
    int rear;  
    int size;  
    int max;  
  
    public QueueNasabah_18(int n) {  
        max = n;  
        data = new Nasabah_18[max];  
        size = 0;  
        front = rear = -1;  
    }  
}
```

```
public void Enqueue(Nasabah_18 dt) {  
    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++;  
    }  
}
```

```

public Nasabah_18 Dequeue() {
    Nasabah_18 dt = new Nasabah_18();
    if (IsEmpty()) {
        System.out.println(x:"Queue masih kosong");
    } else {
        dt = data[front];
        size--;
        if (IsEmpty()) {
            front = rear = -1;
        } else {
            if (front == max - 1) {
                front = 0;
            } else {
                front++;
            }
        }
    }
    return dt;
}

```

```

public Nasabah_18() {
}

```

- 6) Karena satu elemen queue terdiri dari beberapa informasi (norek, nama, alamat, umur, dan saldo), maka ketika mencetak data juga perlu ditampilkan semua informasi tersebut, sehingga modifikasi perlu dilakukan pada method peek dan method print.

```

public void peek() {
    if (!IsEmpty()) {
        System.out.println(x:"=====");
        System.out.println(x:"          ELEMEN TERDEPAN");
        System.out.println(x:"=====");
        System.out.println("Nomor Rekening : " + data[front].norek);
        System.out.println("Nama          : " + data[front].nama);
        System.out.println("Alamat       : " + data[front].alamat);
        System.out.println("Umur        : " + data[front].umur);
        System.out.println("Saldo       : " + data[front].saldo);
    } else {
        System.out.println(x:"Queue masih kosong");
    }
}

```



```

public void print() {
    if (IsEmpty()) {
        System.out.println(x:"Queue masih kosong");
    } else {
        System.out.println(x:"=====");
        System.out.println(x:"          DATA SEMUA ANTRIAN");
        System.out.println(x:"=====");
        int i = front;
        while (i != rear) {
            System.out.println("Nomor Rekening : " + data[front].norek);
            System.out.println("Nama          : " + data[front].nama);
            System.out.println("Alamat       : " + data[front].alamat);
            System.out.println("Umur         : " + data[front].umur);
            System.out.println("Saldo        : " + data[front].saldo);
            System.out.println(x:"=====");
            i = (i + 1) % max;
        }
        System.out.println("Nomor Rekening : " + data[i].norek);
        System.out.println("Nama          : " + data[i].nama);
        System.out.println("Alamat       : " + data[i].alamat);
        System.out.println("Umur         : " + data[i].umur);
        System.out.println("Saldo        : " + data[i].saldo);
        System.out.println(x:"=====");
        System.out.println("Jumlah elemen = " + size);
    }
}

```

- 7) Selanjutnya, buat class baru dengan nama QueueMain tetap pada package Praktikum2. Buat method menu untuk mengakomodasi pilihan menu dari masukan pengguna

```

public static void menu() {
    System.out.println(x:"=====");
    System.out.println(x:"          MENU");
    System.out.println(x:"=====");
    System.out.println(x:"1. Antrian Baru");
    System.out.println(x:"2. Antrian Keluar");
    System.out.println(x:"3. Cek Antrian Terdepan");
    System.out.println(x:"4. Cek Semua Antrian");
    System.out.println(x:"5. Keluar");
    System.out.println(x:"=====");
}

```

- 8) Buat fungsi main, deklarasikan Scanner dengan nama input18

```

Scanner input18 = new Scanner(System.in);
Scanner inputString18 = new Scanner(System.in);

```

- 9) Buat variabel jumlah untuk menampung kapasitas elemen pada queue. Kemudian lakukan instansiasi objek queue dengan nama antri dan nilai parameternya adalah variabel jumlah.

```
System.out.print(s:"Masukan kapasitas antrian  : ");  
int jumlah = input18.nextInt();  
  
QueueNasabah_18 antri = new QueueNasabah_18(jumlah);
```

- 10) Deklarasikan variabel dengan nama pilih bertipe integer untuk menampung pilih menu dari pengguna.

```
int pilihan = 0;
```

- 11) Tambahkan kode berikut untuk melakukan perulangan menu sesuai dengan masukan yang diberikan oleh pengguna.

```

1  do {
2      menu();
3      pilihan = input18.nextInt();
4      switch (pilihan) {
5          case 1:
6              System.out.println("=====");
7              System.out.println("          MASUKKAN DATA DIRI");
8              System.out.println("=====");
9              System.out.print("Nomor Rekening : ");
10             String norek = inputString18.nextLine();
11             System.out.print("Nama          : ");
12             String nama = inputString18.nextLine();
13             System.out.print("Alamat         : ");
14             String alamat = inputString18.nextLine();
15             System.out.print("Umur           : ");
16             int umur = input18.nextInt();
17             input18.nextLine();
18             System.out.print("Saldo           : ");
19             double saldo = input18.nextDouble();
20             input18.nextLine();
21             Nasabah_18 nb = new Nasabah_18(norek, nama, alamat, umur, saldo);
22             antri.Enqueue(nb);
23             break;
24          case 2:
25             Nasabah_18 data = antri.Dequeue();
26             if (!"".equalsIgnoreCase(data.norek) && !"".equalsIgnoreCase(data.nama)
27                 && !"".equalsIgnoreCase(data.alamat) && data.umur != 0 && data.saldo != 0) {
28                 System.out.println("=====");
29                 System.out.println("          ANTRIAN YANG KELUAR");
30                 System.out.println("=====");
31                 System.out.println("Nomor Rekening : " + data.norek);
32                 System.out.println("Nama            : " + data.nama);
33                 System.out.println("Alamat          : " + data.alamat);
34                 System.out.println("Umur            : " + data.umur);
35                 System.out.println("Saldo           : " + data.saldo);
36             }
37             break;
38          case 3:
39             antri.peek();
40             break;
41          case 4:
42             antri.print();
43             break;
44          default:
45             break;
46      }
47      } while (pilihan != 5);
48
49      input18.close();
50      inputString18.close();
51
52  }

```

12) Compile dan jalankan class QueueMain, kemudian amati hasilnya dan verifikasi.


```
Masukan kapasitas antrian : 4
=====
                        MENU
=====
1. Antrian Baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian
5. Keluar
=====
1
=====
                        MASUKKAN DATA DIRI
=====
Nomor Rekening : 1200046675
Nama           : Arif Didu
Alamat         : Sukun, Malang
Umur           : 12
Saldo          : 12,000,000
=====
                        MENU
=====
1. Antrian Baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian
5. Keluar
=====
1
=====
                        MASUKKAN DATA DIRI
=====
Nomor Rekening : 1200198733
Nama           : Dewi Sri
Alamat         : Rungkut, Surabaya
Umur           : 30
Saldo          : 8,600,000
```

```

=====
                        MENU
=====
1. Antrian Baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian
5. Keluar
=====
4
=====
Nomor Rekening : 1200046675
Nama           : Arif Didu
Alamat        : Sukun, Malang
Umur          : 12
Saldo         : 1.2E7
=====
Nomor Rekening : 1200198733
Nama           : Dewi Sri
Alamat        : Rungkut, Surabaya
Umur          : 30
Saldo         : 8600000.0
=====
Jumlah elemen = 2
=====
                        MENU
=====
1. Antrian Baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian
5. Keluar
=====
3
Elemen terdepan :
Nomor Rekening : 1200046675
Nama           : Arif Didu
Alamat        : Sukun, Malang
Umur          : 12
Saldo         : 1.2E7

```

```

=====
                        MENU
=====
1. Antrian Baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian
5. Keluar
=====
2
=====
                        ANTRIAN YANG KELUAR
=====
Nomor Rekening : 1200046675
Nama           : Arif Didu
Alamat         : Sukun, Malang
Umur           : 12
Saldo          : 1.2E7
=====
                        MENU
=====
1. Antrian Baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian
5. Keluar
=====
4
=====
Nomor Rekening : 1200198733
Nama           : Dewi Sri
Alamat         : Rungkut, Surabaya
Umur           : 30
Saldo          : 8600000.0
=====
Jumlah elemen = 1
=====
                        MENU
=====
1. Antrian Baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian
5. Keluar
=====
5

```

Pertanyaan Percobaan 2

- 1) Pada class QueueMain, jelaskan fungsi IF pada potongan kode program berikut!

Kode ini memeriksa apakah data nasabah yang diambil dari antrian memiliki nilai yang valid untuk nomor rekening (norek), nama, alamat, umur, dan saldo. Jika salah satu dari data tersebut tidak kosong (tidak sama dengan string kosong) atau umur dan saldo tidak sama dengan 0, maka kondisi tersebut dianggap sebagai data yang valid dan akan dijalankan perintah untuk menampilkan data nasabah yang keluar dari antrian.

- 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!

```
public void peekRear() {  
    if (!IsEmpty()) {  
        System.out.println(x: "=====");  
        System.out.println(x: "            ELEMEN TERDEPAN");  
        System.out.println(x: "=====");  
        System.out.println("Nomor Rekening : " + data[rear].norek);  
        System.out.println("Nama          : " + data[rear].nama);  
        System.out.println("Alamat       : " + data[rear].alamat);  
        System.out.println("Umur        : " + data[rear].umur);  
        System.out.println("Saldo       : " + data[rear].saldo);  
    } else {  
        System.out.println(x: "Queue masih kosong");  
    }  
}
```

```
public static void menu() {  
    System.out.println(x: "=====");  
    System.out.println(x: "            MENU");  
    System.out.println(x: "=====");  
    System.out.println(x: "1. Antrian Baru");  
    System.out.println(x: "2. Antrian Keluar");  
    System.out.println(x: "3. Cek Antrian Terdepan");  
    System.out.println(x: "4. Cek Semua Antrian");  
    System.out.println(x: "5. Cek Antrian Terbelakang");  
    System.out.println(x: "6. Keluar");  
    System.out.println(x: "=====");  
}
```

```
case 5:  
    antri.peekRear();  
    break;
```

Tugas Praktikum

- 1) Buatlah program antrian untuk mengilustrasikan antrian pasien di sebuah klinik. Ketika seorang pasien akan mengantri, maka dia harus mendaftarkan nama, nomor identitas, jenis kelamin dan umur seperti yang digambarkan pada Class diagram berikut:

Pembeli
nama: String noID: int jenisKelamin: char umur: int Pasien (nama: String, noID: int, jenisKelamin: char, umur: int)

Queue
antrian: Pasien[] front: int rear: int size: int max: int Queue(n: int) isEmpty(): boolean isFull(): boolean enqueue(antri: Pasien): void dequeue(): int print(): void peek(): void peekRear(): void peekPosition(nama: String): void daftarPasien(): void

Keterangan method:

- Method create(), isEmpty(), isFull(), enqueue(), dequeue() dan print(), kegunaannya sama seperti yang telah dibuat pada Praktikum
- Method peek(): digunakan untuk menampilkan data Pasien yang berada di posisi antrian paling depan
- Method peekRear(): digunakan untuk menampilkan data Pasien yang berada di posisi antrian paling belakang
- Method peekPosition(): digunakan untuk menampilkan seorang pasien (berdasarkan nama) posisi antrian ke berapa
- Method daftarPasien(): digunakan untuk menampilkan data seluruh pasien

Jawaban

1. a. Program

```

1 public class QueuePasiin_18 {
2
3     Pasiin dt; antrian;
4     int front;
5     int rear;
6     int size;
7     int max;
8
9     public QueuePasiin_18(int n) {
10         max = n;
11         antrian = new Pasiin_18[max];
12         size = 0;
13         front = rear = -1;
14     }
15
16     public boolean isEmpty() {
17         if (size == 0) {
18             return true;
19         } else {
20             return false;
21         }
22     }
23
24     public boolean isFull() {
25         if (size == max) {
26             return true;
27         } else {
28             return false;
29         }
30     }
31
32     public void peek() {
33         if (!isEmpty()) {
34             System.out.println("=====");
35             System.out.println("          ANTRIAN TERDEPAK");
36             System.out.println("=====");
37             System.out.println("Nama          : " + antrian[front].nama);
38             System.out.println("Nomor ID       : " + antrian[front].noID);
39             System.out.println("Jenis Kelamin  : " + antrian[front].jenisKelamin);
40             System.out.println("Umur          : " + antrian[front].umur);
41         } else {
42             System.out.println("Queue masih kosong");
43         }
44     }
45
46     public void peekRear() {
47         if (!isEmpty()) {
48             System.out.println("=====");
49             System.out.println("          ANTRIAN TERDEPAK");
50             System.out.println("=====");
51             System.out.println("Nama          : " + antrian[rear].nama);
52             System.out.println("Nomor ID       : " + antrian[rear].noID);
53             System.out.println("Jenis Kelamin  : " + antrian[rear].jenisKelamin);
54             System.out.println("Umur          : " + antrian[rear].umur);
55         } else {
56             System.out.println("Queue masih kosong");
57         }
58     }
59
60     public void peekPosition(String nama) {
61         boolean found = false;
62         if (!isEmpty()) {
63             for (int i = 0; i < antrian.length; i++) {
64                 if (antrian[i] != null && antrian[i].nama.equalsIgnoreCase(nama)) {
65                     System.out.println("=====");
66                     System.out.println("          DATA PASIIN DITEMUKAN");
67                     System.out.println("=====");
68                     System.out.println("Nama          : " + antrian[i].nama);
69                     System.out.println("Nomor ID       : " + antrian[i].noID);
70                     System.out.println("Jenis Kelamin  : " + antrian[i].jenisKelamin);
71                     System.out.println("Umur          : " + antrian[i].umur);
72                     found = true;
73                     break;
74                 }
75             }
76         }
77         if (!found) {
78             System.out.println("Data pasien dengan nama " + nama + " tidak ditemukan.");
79         }
80     }
81
82     public void displayAntrian() {
83         if (!isEmpty()) {
84             System.out.println("Queue masih kosong");
85         } else {
86             System.out.println("=====");
87             System.out.println("          DATA SEMUA ANTRIAN");
88             System.out.println("=====");
89             int i = front;
90             do {
91                 System.out.println("Nama          : " + antrian[i].nama);
92                 System.out.println("Nomor ID       : " + antrian[i].noID);
93                 System.out.println("Jenis Kelamin  : " + antrian[i].jenisKelamin);
94                 System.out.println("Umur          : " + antrian[i].umur);
95                 System.out.println("=====");
96                 i = (i + 1) % max;
97             } while (i != (rear + 1) % max);
98             System.out.println("Jumlah elemen : " + size);
99         }
100     }
101
102     public void clear() {
103         if (!isEmpty()) {
104             front = rear = -1;
105             size = 0;
106             System.out.println("Queue berhasil dihapuskan");
107         } else {
108             System.out.println("Queue masih kosong");
109         }
110     }
111
112     public void enqueue(Pasiin_18 dt) {
113         if (!isEmpty()) {
114             System.out.println("Queue sudah penuh");
115         } else {
116             if (!isFull()) {
117                 front = rear = 0;
118             } else {
119                 if (rear == max - 1) {
120                     rear = 0;
121                 } else {
122                     rear++;
123                 }
124             }
125             antrian[rear] = dt;
126             size++;
127         }
128     }
129
130     public Pasiin_18 dequeue() {
131         Pasiin_18 dt = new Pasiin_18();
132         if (!isEmpty()) {
133             System.out.println("Queue masih kosong");
134         } else {
135             dt = antrian[front];
136             size--;
137             if (!isEmpty()) {
138                 front = rear = -1;
139             } else {
140                 if (front == max - 1) {
141                     front = 0;
142                 } else {
143                     front++;
144                 }
145             }
146         }
147         return dt;
148     }
149 }
150
151
152
153
154

```




```
1  public class Pasien_18 {  
2  
3      String nama;  
4      int noID;  
5      char jenisKelamin;  
6      int umur;  
7  
8      Pasien_18(String nama, int noID, char jenisKelamin, int umur) {  
9          this.nama = nama;  
10         this.noID = noID;  
11         this.jenisKelamin = jenisKelamin;  
12         this.umur = umur;  
13     }  
14  
15     public Pasien_18() {  
16  
17     }  
18  
19 }  
20
```

```

1  import java.util.Scanner;
2
3  public class PasienMain_18 {
4      public static void main(String[] args) {
5          Scanner input18 = new Scanner(System.in);
6          Scanner inputString18 = new Scanner(System.in);
7
8          System.out.print("Masukan kapasitas antrian : ");
9          int jumlah = input18.nextInt();
10
11         QueuePasien_18 antri = new QueuePasien_18(jumlah);
12
13         int pilihan = 0;
14         do {
15             menu();
16             pilihan = input18.nextInt();
17             switch (pilihan) {
18                 case 1:
19                     System.out.println("=====");
20                     System.out.println("          MASUKKAN DATA DIRI");
21                     System.out.println("=====");
22                     System.out.print("Nama Pasien      : ");
23                     String nama = inputString18.nextLine();
24                     System.out.print("Nomor ID        : ");
25                     int noID = input18.nextInt();
26                     System.out.print("Jenis Kelamin (P/L) : ");
27                     char jenisKelamin = input18.next().charAt(0);
28                     System.out.print("Umur            : ");
29                     int umur = input18.nextInt();
30                     input18.nextLine();
31                     Pasien_18 nb = new Pasien_18(nama, noID, jenisKelamin, umur);
32                     antri.Enqueue(nb);
33                     break;
34                 case 2:
35                     Pasien_18 data = antri.Dequeue();
36                     if (!"".equalsIgnoreCase(data.nama) && data.noID != 0
37                         && !Character.isWhitespace(data.jenisKelamin)
38                         && (data.jenisKelamin == 'L' || data.jenisKelamin == 'P')
39                         && data.umur != 0) {
40                         System.out.println("=====");
41                         System.out.println("          ANTRIAN YANG KELUAR");
42                         System.out.println("=====");
43                         System.out.println("Nama           : " + data.nama);
44                         System.out.println("Nomor ID       : " + data.noID);
45                         System.out.println("Jenis Kelamin  : " + data.jenisKelamin);
46                         System.out.println("Umur           : " + data.umur);
47                     }
48                     break;
49                 case 3:
50                     antri.peek();
51                     break;
52                 case 4:
53                     antri.peekRear();
54                     break;
55                 case 5:
56                     System.out.println("=====");
57                     System.out.print("Masukkan nama pasien : ");
58                     String cariNama = inputString18.nextLine();
59                     System.out.println("=====");
60                     antri.peekPosition(cariNama);
61                     break;
62                 case 6:
63                     antri.daftarPasien();
64                     break;
65                 default:
66                     break;
67             }
68         } while (pilihan != 7);
69
70         input18.close();
71         inputString18.close();
72     }
73
74     public static void menu() {
75         System.out.println("=====");
76         System.out.println("          MENU");
77         System.out.println("=====");
78         System.out.println("1. Pasien Baru");
79         System.out.println("2. Pasien Keluar");
80         System.out.println("3. Cek Pasien Terdepan");
81         System.out.println("4. Cek Pasien Terbelakang");
82         System.out.println("5. Cari Pasien Dengan Nama");
83         System.out.println("6. Cek Semua Daftar Pasien");
84         System.out.println("7. Keluar");
85         System.out.println("=====");
86     }
87 }
88
89
90
91
92

```

1. b. Output

```
Masukan kapasitas antrian : 4
=====
                        MENU
=====
1. Pasien Baru
2. Pasien Keluar
3. Cek Pasien Terdepan
4. Cek Pasien Terbelakang
5. Cari Pasien Dengan Nama
6. Cek Semua Daftar Pasien
7. Keluar
=====
1
=====
                        MASUKKAN DATA DIRI
=====
Nama Pasien      : Imam
Nomor ID         : 1234
Jenis Kelamin (P/L) : L
Umur             : 20
=====
                        MENU
=====
1. Pasien Baru
2. Pasien Keluar
3. Cek Pasien Terdepan
4. Cek Pasien Terbelakang
5. Cari Pasien Dengan Nama
6. Cek Semua Daftar Pasien
7. Keluar
=====
1
=====
                        MASUKKAN DATA DIRI
=====
Nama Pasien      : Darto
Nomor ID         : 1235
Jenis Kelamin (P/L) : L
Umur             : 23
```

```
=====
                        MENU
=====
1. Pasien Baru
2. Pasien Keluar
3. Cek Pasien Terdepan
4. Cek Pasien Terbelakang
5. Cari Pasien Dengan Nama
6. Cek Semua Daftar Pasien
7. Keluar
=====
1
=====
                        MASUKKAN DATA DIRI
=====
Nama Pasien           : Indah
Nomor ID              : 1236
Jenis Kelamin (P/L)   : P
Umur                  : 19
=====
                        MENU
=====
1. Pasien Baru
2. Pasien Keluar
3. Cek Pasien Terdepan
4. Cek Pasien Terbelakang
5. Cari Pasien Dengan Nama
6. Cek Semua Daftar Pasien
7. Keluar
=====
1
=====
                        MASUKKAN DATA DIRI
=====
Nama Pasien           : Mawar
Nomor ID              : 1237
Jenis Kelamin (P/L)   : P
Umur                  : 19
=====
```

```
=====
                        MENU
=====
1. Pasien Baru
2. Pasien Keluar
3. Cek Pasien Terdepan
4. Cek Pasien Terbelakang
5. Cari Pasien Dengan Nama
6. Cek Semua Daftar Pasien
7. Keluar
=====
3
=====
                        ANTRIAN TERDEPAN
=====
Nama           : Imam
Nomor ID       : 1234
Jenis Kelamin  : L
Umur          : 20
=====
                        MENU
=====
1. Pasien Baru
2. Pasien Keluar
3. Cek Pasien Terdepan
4. Cek Pasien Terbelakang
5. Cari Pasien Dengan Nama
6. Cek Semua Daftar Pasien
7. Keluar
=====
4
=====
                        ANTRIAN TERDEPAN
=====
Nama           : Mawar
Nomor ID       : 1237
Jenis Kelamin  : P
Umur          : 19
```

```

=====
                        MENU
=====
1. Pasien Baru
2. Pasien Keluar
3. Cek Pasien Terdepan
4. Cek Pasien Terbelakang
5. Cari Pasien Dengan Nama
6. Cek Semua Daftar Pasien
7. Keluar
=====
2
=====
                        ANTRIAN YANG KELUAR
=====
Nama           : Imam
Nomor ID       : 1234
Jenis Kelamin  : L
Umur           : 20
=====
                        MENU
=====
1. Pasien Baru
2. Pasien Keluar
3. Cek Pasien Terdepan
4. Cek Pasien Terbelakang
5. Cari Pasien Dengan Nama
6. Cek Semua Daftar Pasien
7. Keluar
=====
5
=====
Masukkan nama pasien   : Indah
=====
=====
                        DATA PASIEN DITEMUKAN
=====
Nama           : Indah
Nomor ID       : 1236
Jenis Kelamin  : P
Umur           : 19

```



```

=====
                        MENU
=====
1. Pasien Baru
2. Pasien Keluar
3. Cek Pasien Terdepan
4. Cek Pasien Terbelakang
5. Cari Pasien Dengan Nama
6. Cek Semua Daftar Pasien
7. Keluar
=====
6
=====
                        DATA SEMUA ANTRIAN
=====
Nama          : Darto
Nomor ID      : 1235
Jenis Kelamin : L
Umur         : 23
=====
Nama          : Indah
Nomor ID      : 1236
Jenis Kelamin : P
Umur         : 19
=====
Nama          : Mawar
Nomor ID      : 1237
Jenis Kelamin : P
Umur         : 19
=====
Jumlah elemen = 3
=====
                        MENU
=====
1. Pasien Baru
2. Pasien Keluar
3. Cek Pasien Terdepan
4. Cek Pasien Terbelakang
5. Cari Pasien Dengan Nama
6. Cek Semua Daftar Pasien
7. Keluar
=====
7

```

Repository : <https://github.com/zidnafaz/Praktikum-Algoritma-Struktur-Data>