NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Queue

LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

10. 2 Praktikum 1

Queue :

```java
package Pratikum1;

public class Queue13 {
    int[] data;
    int front;
    int rear;
    int size;
    int max;

    public Queue13(int n) {
        max = n;
        data = new int[max];
        size = 0;
        front = rear = -1;
    }

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

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

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

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

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

    public void Enqueue(int dt) {
        if (IsFull()) {
            System.out.println("Queue sudah penuh");
        } else {
            if (IsEmpty()) {
                front = rear = 0;
            } else {
                if (rear == max - 1) {
                    rear = 0;
                } else {
                    rear++;
                }
            }
            data[rear] = dt;
            size++;
        }
    }
```

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

NAMA	: Gilang Purnomo
NIM	: 2341720042
NO ABSEN	: 13
KELAS	: 1F
MATERI	: Queue

QueueMain :

src > Pratikum1 > QueueMain13.java > QueueMain13 > main(String[])

```java
package Pratikum1;

import java.util.Scanner;

Codeium: Refactor | Explain
public class QueueMain13 {
    Codeium: Refactor | Explain | Generate Javadoc | ×
    public static void menu() {
        System.out.println(x:"Masukkan 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:"----------------------");
    }

    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | ×
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print(s:"Masukkan kapasitas queue: ");
        int n = sc.nextInt();
        Queue13 Q = new Queue13(n);
        int pilih;

        do {
            menu();
            pilih = sc.nextInt();
            switch (pilih) {
                case 1:
                    System.out.print(s:"Masukkan data baru: ");
                    int dataMasuk = sc.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;
            }
        } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5);
    }
}
```

NAMA	: Gilang Purnomo
NIM	: 2341720042
NO ABSEN	: 13
KELAS	: 1F
MATERI	: Queue

Output :

```
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
---------------------
```

Question :

1. Pada konstruktor, mengapa nilai awal atribut front dan rear bernilai -1, sementara atribut size bernilai 0?
   Jawab : front dan rear bernilai -1 karena untuk menandai bahwa elemen masih kosong dalam index dan size bernilai 0 untuk memberitahu bahwa antrian belum memiliki elemen.

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

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

   Jawab : Jika kondisi itu terpenuhi yaitu indeks rear sama dengan jumlah max -1 maka rear ditempatkan di indexs 0.

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

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

NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Queue

Jawab : Jika kondisi terpenuhi yaitu indeks front sama dengan jumlah max – 1 maka front ditempatkan pada indexs 0.

4. Pada method **print**, mengapa pada proses perulangan variabel i tidak dimulai dari 0 **(int i=0)**, melainkan **int i=front**?
   Jawab : Karena jika dimulai dari 0 menunjukkan bahwa antrian masih kosong. Maka dari **itu int i = front**, bahwa i adalah elemen terdepan pada antrian.

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

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

   Jawab : Untuk mengatur iterasi agar melakukan iterasi antrian secara terurut dan efisien sampai batas array.

6. Tunjukkan potongan kode program yang merupakan queue overflow!
   Jawab :

```
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
   Jawab :
   queue overflow :

```
    public void Enqueue(int dt) {
        if (IsFull()) {
            System.out.println(x:"Queue sudah penuh");
            System.exit(status:0);
```

```
Masukkan kapasitas queue: 2
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: 17
Queue sudah penuh
PS E:\KULIAH 2\Pratikum Algoritma dan Struktur Data\PrakASD_1F_13>
```

NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Queue

queue underflow :

```
    public int Dequeue() {
        int dt = 0;
        if (IsEmpty()) {
            System.out.println(x:"Queue masih kosong");
            System.exit(status:0);
```

```
Masukkan kapasitas queue: 1
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
----------------------
1
Masukkan data baru: 13
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
----------------------
2
Data yang dikeluarkan: 13
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
----------------------
2
Queue masih kosong
PS E:\KULIAH 2\Pratikum Algoritma dan Struktur Data\PrakASD_1F_13>
```

10. 3 Praktikum 2

Nasabah :

src > P10 > Praktikum2 > Nasabah13.java > Nasabah13 > Nasabah13()

```
1  package P10.Praktikum2;
2
   Codeium: Refactor | Explain
3  public class Nasabah13 {
4      String norek;
5      String nama;
6      String alamat;
7      int umur;
8      double saldo;
9
10
11     public Nasabah13(String norek, String nama, String alamat, int umur, double saldo) {
12         this.norek = norek;
13         this.nama = nama;
14         this.alamat = alamat;
15         this.umur = umur;
16         this.saldo = saldo;
17     }
18
19
20     public Nasabah13() {
21
22     }
23 }
24
```

NAMA	: Gilang Purnomo
NIM	: 2341720042
NO ABSEN	: 13
KELAS	: 1F
MATERI	: Queue

Queue :

src > P10 > Praktikum2 > Queue13.java > Queue13 > print()

```java
package P10.Praktikum2;

Codeium: Refactor | Explain
public class Queue13 {
    Nasabah13[] data;
    int front;
    int rear;
    int size;
    int max;

    public Queue13(int n) {
        max = n;
        data = new Nasabah13[max];
        size = 0;
        front = rear = -1;
    }

    Codeium: Refactor | Explain | Generate Javadoc | ×
    public boolean IsEmpty() {
        if (size == 0) {
            return true;
        } else {
            return false;
        }
    }

    Codeium: Refactor | Explain | Generate Javadoc | ×
    public boolean IsFull() {
        if (size == max) {
            return true;
        } else {
            return false;
        }
    }

    Codeium: Refactor | Explain | Generate Javadoc | ×
    public void peek() {
        if (!IsEmpty()) {
            System.out.println("Elemen terdepan: " + data[front].norek + " " + data[front].nama
                    + " " + data[front].alamat + " " + data[front].umur + " " + data[front].saldo);
        } else {
            System.out.println(x:"Queue masih kosong");
        }
    }
```

```java

    Codeium: Refactor | Explain | Generate Javadoc | ×
    public void print() {
        if (IsEmpty()) {
            System.out.println(x:"Queue masih kosong");
        } else {
            int i = front;
            while (i != rear) {
                System.out.println(data[i].norek + " " + data[i].nama + " " + data[i].alamat
                        + " " + data[i].umur + " " + data[i].saldo);
                i = (i + 1) % max;
            }
            System.out.println(data[i].norek + " " + data[i].nama + " " + data[i].alamat
                    + " " + data[i].umur + " " + data[i].saldo);
            System.out.println("Jumlah elemen = " + size);
        }
    }

    Codeium: Refactor | Explain | Generate Javadoc | ×
    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");
        }
    }
```

NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Queue

```java
public void Enqueue(Nasabah13 dt) {
    if (IsFull()) {
        System.out.println(x:"Queue sudah penuh");
        System.exit(status:0);
    } else {
        if (IsEmpty()) {
            front = rear = 0;
        } else {
            if (rear == max - 1) {
                rear = 0;
            } else {
                rear++;
            }
        }
        data[rear] = dt;
        size++;
    }
}

Codeium: Refactor | Explain | Generate Javadoc | ×
public Nasabah13 Dequeue() {
    Nasabah13 dt = new Nasabah13();
    if (IsEmpty()) {
        System.out.println(x:"Queue masih kosong");
        System.exit(status:0);
    } else {
        dt = data[front];
        size--;
        if (IsEmpty()) {
            front = rear = -1;
        } else {
            if (front == max -1) {
                front = 0;
            } else {
                front++;
            }
        }
    }
    return dt;
}
}
```

QueueMain :

```java
src > P10 > Praktikum2 > QueueMain13.java > QueueMain13 > menu()
package P10.Praktikum2;

import java.util.Scanner;

Codeium: Refactor | Explain
public class QueueMain13 {
    Codeium: Refactor | Explain | Generate Javadoc | ×
    public static void menu() {
        System.out.println("Pilih menu: ");
        System.out.println("1. Antrian baru");
        System.out.println("2. Antrian Keluar");
        System.out.println("3. Cek Antrian terdepan");
        System.out.println("4. Cek Semua Antrian");
        System.out.println("----------------------");
    }

    Codeium: Refactor | Explain | Generate Javadoc | ×
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Masukkan kapasitas queue: ");
        int jumlah = sc.nextInt();
        Queue13 antri = new Queue13(jumlah);
        int pilih;
```

NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Queue

```
        do {
            menu();
            pilih = sc.nextInt();
            sc.nextLine();
            switch (pilih) {
                case 1:
                    System.out.print(s:"No Rekening: ");
                    String norek = sc.nextLine();
                    System.out.print(s:"Nama: ");
                    String nama = sc.nextLine();
                    System.out.print(s:"Alamat: ");
                    String alamat = sc.nextLine();
                    System.out.print(s:"Umur: ");
                    int umur = sc.nextInt();
                    System.out.print(s:"Saldo: ");
                    double saldo = sc.nextDouble();
                    Nasabah13 nb = new Nasabah13(norek, nama, alamat, umur, saldo);
                    sc.nextLine();
                    antri.Enqueue(nb);
                    break;
                case 2:
                    Nasabah13 data = antri.Dequeue();
                    if (!"".equals(data.norek) && !"".equals(data.nama) && !"".equals(data.alamat)
                            && data.umur != 0 && data.saldo !=0) {
                        System.out.println("Antrian yang keluar: " + data.norek + " " + data.nama + " "
                                + data.alamat + " " + data.umur + " " + data.saldo);
                        break;
                    }
                case 3:
                    antri.peek();
                    break;
                case 4:
                    antri.print();
                    break;
            }
        } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5);
    }
}
```

Output :

```
Masukkan kapasitas queue: 8
Pilih menu:
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
----------------------
1
No Rekening: 12345
Nama: Dewi
Alamat: Malang
Umur: 23
Saldo: 1300000
Pilih menu:
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
----------------------
1
No Rekening: 32940
Nama: Susan
Alamat: Surabaya
Umur: 39
Saldo: 42000000
Pilih menu:
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
----------------------
4
12345 Dewi Malang 23 1300000.0
32940 Susan Surabaya 39 4.2E7
Jumlah elemen = 2
Pilih menu:
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
----------------------
```

NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Queue

Question :

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

```
if (!"".equals(data.norek) && !"".equals(data.nama) && !"".equals(data.alamat)
        && data.umur != 0 && data.saldo != 0) {
    System.out.println("Antrian yang keluar: " + data.norek + " " + data.nama + " "
            + data.alamat + " " + data.umur + " " + data.saldo);
    break;
}
```

Jawab : Pada If tersebut adalah jika kondisi didalam terisi oleh data dan tidak null serta umur dan saldo tidak 0 maka akan menampilkan semua data data tersebut, seperti norek, nama, Alamat, umur, dan jumlah salldo.

2. Lakukan modifikasi program dengan menambahkan method baru bernama **peekRear** pada class Queue yang digunakan untuk mengecek antrian yang berada di posisi belakang! Tambahkan pula daftar menu **5. Cek Antrian paling belakang** pada class **QueueMain** sehingga method **peekRear** dapat dipanggil!

Jawab :

```
public void peekRear() {
    if (!IsEmpty()) {
        System.out.println("Elemen belakang " + data[rear].norek + " " + data[rear].nama
                + " " + data[rear].alamat + " " + data[rear].umur + " " + data[rear].saldo);
    } else {
        System.out.println("Queue masih kosong");
    }
}
```

```
public static void menu() {
    System.out.println("Pilih menu: ");
    System.out.println("1. Antrian baru");
    System.out.println("2. Antrian Keluar");
    System.out.println("3. Cek Antrian terdepan");
    System.out.println("4. Cek Semua Antrian");
    System.out.println("5. Cek Antrian paling belakang");
    System.out.println("----------------------");
}
```

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

```
Masukkan kapasitas queue: 8
Pilih menu:
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian paling belakang
----------------------
1
No Rekening: 12345
Nama: Dewi
Alamat: Malang
Umur: 23
Saldo: 1300000
Pilih menu:
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian paling belakang
----------------------
1
No Rekening: 32940
Nama: Susen
Alamat: Surabaya
Umur: 39
Saldo: 42000000
Pilih menu:
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian paling belakang
----------------------
4
12345 Dewi Malang 23 1300000.0
32940 Susen Surabaya 39 4.2E7
Jumlah elemen = 2
```

```
Pilih menu:
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian paling belakang
----------------------
3
Elemen terdepan: 12345 Dewi Malang 23 1300000.0
Pilih menu:
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian paling belakang
----------------------
5
Elemen belakang 32940 Susen Surabaya 39 4.2E7
Pilih menu:
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian paling belakang
----------------------
```

NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Queue

10. 4 Tugas

1. Buatlah program antrian untuk mengilustasikan pesanan disebuah warung. Ketika seorang pembeli akan mengantri, maka dia harus mendaftarkan nama, dan nomor HP seperti yang digambarkan pada Class diagram berikut:

Pembeli
nama: String noHP: int
Pembeli(nama: String, noHP: int)

Class diagram Queue digambarkan sebagai berikut:

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

Keterangan:

• Method create(), isEmpty(), isFull(), enqueue(), dequeue() dan print(), kegunaannya sama seperti yang telah dibuat pada Praktikum

• Method peek(): digunakan untuk menampilkan data Pembeli yang berada di posisi antrian paling depan

• Method peekRear(): digunakan untuk menampilkan data Pembeli yang berada di posisi antrian paling belakang

• Method peekPosition(): digunakan untuk menampilkan seorang pembeli (berdasarkan nama) posisi antrian ke berapa

• Method daftarPembeli(): digunakan untuk menampilkan data seluruh pembeli

NAMA	: Gilang Purnomo
NIM	: 2341720042
NO ABSEN	: 13
KELAS	: 1F
MATERI	: Queue

Pembeli

```java
package P10.Tugas;

Codeium: Refactor | Explain
public class Pembeli13 {
    String nama;
    int noHP;


    public Pembeli13(String nama, int noHP) {
        this.nama = nama;
        this.noHP = noHP;
    }

    public Pembeli13() {
    }

}
```

Queue

```java
package P10.Tugas;

Codeium: Refactor | Explain
public class Queue13 {
    Pembeli13[] antrian;
    int front;
    int rear;
    int size;
    int max;

    public Queue13(int n) {
        max = n;
        antrian = new Pembeli13[max];
        size = 0;
        front = rear = -1;
    }

    Codeium: Refactor | Explain | Generate Javadoc | ×
    public boolean IsEmpty() {
        if (size == 0) {
            return true;
        } else {
            return false;
        }
    }

    Codeium: Refactor | Explain | Generate Javadoc | ×
    public boolean IsFull() {
        if (size == max) {
            return true;
        } else {
            return false;
        }
    }

    Codeium: Refactor | Explain | Generate Javadoc | ×
    public void peek() {
        if (!IsEmpty()) {
            System.out.println("Elemen terdepan: " + antrian[front].nama);
        } else {
            System.out.println(x:"Queue masih kosong");
        }
    }
```

NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Queue

```
Codeium: Refactor | Explain | Generate Javadoc | ×
41  public void peekRear() {
42      if (!IsEmpty()) {
43          System.out.println("Elemen belakang: " + antrian[rear].nama);
44      } else {
45          System.out.println(x:"Queue masih kosong");
46      }
47  }
48
Codeium: Refactor | Explain | Generate Javadoc | ×
49  public void peekPosition(String nama) {
50      if (IsEmpty()) {
51          System.out.println(x:"Queue masih kosong");
52      } else {
53          boolean antrianDitemukan = false;
54          int i = front;
55          int posisi = 1;
56          while (i != rear) {
57              if (antrian[i].nama.equalsIgnoreCase(nama)) {
58                  antrianDitemukan = true;
59                  break;
60              }
61              i = (i + 1) % max;
62              posisi++;
63          }
64          if (antrian[i].nama.equalsIgnoreCase(nama)) {
65              antrianDitemukan = true;
66          }
67          if (antrianDitemukan) {
68              System.out.println(x:"Antrian ditemukan");
69              System.out.println("Nama: " + nama);
70              System.out.println("No HP: " + antrian[i].noHP);
71              System.out.println("Posisi Antrian: " + posisi);
72          } else {
73              System.out.println("Antrian dengan nama " + nama + " tidak ditemukan.");
74          }
75      }
76  }
77
```

```
Codeium: Refactor | Explain | Generate Javadoc | ×
78  public void print() {
79      if (IsEmpty()) {
80          System.out.println(x:"Queue masih kosong");
81      } else {
82          int i = front;
83          while (i != rear) {
84              System.out.println(antrian[i].nama);
85              i = (i + 1) % max;
86          }
87          System.out.println(antrian[i].nama);
88          System.out.println("Jumlah elemen = " + size);
89      }
90  }
91
Codeium: Refactor | Explain | Generate Javadoc | ×
92  public void clear() {
93      if (!IsEmpty()) {
94          front = rear = -1;
95          size = 0;
96          System.out.println(x:"Queue berhasil dikosongkan");
97      } else {
98          System.out.println(x:"Queue masih kosong");
99      }
100 }
101
Codeium: Refactor | Explain | Generate Javadoc | ×
102 public void Enqueue(Pembeli13 antri) {
103     if (IsFull()) {
104         System.out.println(x:"Queue sudah penuh");
105     } else {
106         if (IsEmpty()) {
107             front = rear = 0;
108         } else {
109             if (rear == max - 1) {
110                 rear = 0;
111             } else {
112                 rear++;
113             }
114         }
115         antrian[rear] = antri;
116         size++;
117     }
118 }
119
```

NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Queue

```
Codeium: Refactor | Explain | Generate Javadoc | ×
120     public int Dequeue() {
121         int antri = 0;
122         if (IsEmpty()) {
123             System.out.println(x:"Queue masih kosong");
124         } else {
125             antri = front;
126             size--;
127             if (IsEmpty()) {
128                 front = rear = -1;
129             } else {
130                 if (front == max -1) {
131                     front = 0;
132                 } else {
133                     front++;
134                 }
135             }
136         }
137         return antri;
138     }
139
Codeium: Refactor | Explain | Generate Javadoc | ×
140     public void daftarPembeli () {
141         if (IsEmpty()) {
142             System.out.println(x:"Queue masih kosong");
143         } else {
144             int i = front;
145             while (i != rear) {
146                 System.out.println(antrian[i].nama + " " + antrian[i].noHP);
147                 i = (i + 1) % max;
148             }
149             System.out.println(antrian[i].nama + " " + antrian[i].noHP);
150         }
151     }
152 }
153
```

QueueMain

```
src > P10 > Tugas > QueueMain13.java > QueueMain13 > main(String[])
1   package P10.Tugas;
2
3   import java.util.Scanner;
4
    Codeium: Refactor | Explain
5   public class QueueMain13 {
        Codeium: Refactor | Explain | Generate Javadoc | ×
6       public static void menu() {
7           System.out.println(x:"Pilih menu: ");
8           System.out.println(x:"1. Pembeli baru");
9           System.out.println(x:"2. Pembeli Keluar");
10          System.out.println(x:"3. Cek semua Pembeli");
11          System.out.println(x:"4. Cek Pembeli terdepan");
12          System.out.println(x:"5. Cek Pembeli paling belakang");
13          System.out.println(x:"6. Cari Pembeli");
14          System.out.println(x:"7. Data seluruh Pembeli");
15          System.out.println(x:"----------------------");
16      }
17
        Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | ×
18      public static void main(String[] args) {
19          Scanner sc = new Scanner(System.in);
20
21          System.out.print(s:"Masukkan kapasitas queue: ");
22          int jumlah = sc.nextInt();
23          Queue13 pembeli = new Queue13(jumlah);
24          int pilih;
```

NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Queue

```
25
26          do {
27              menu();
28              pilih = sc.nextInt();
29              sc.nextLine();
30              switch (pilih) {
31                  case 1:
32                      System.out.print(s:"Masukkan Nama pembeli: ");
33                      String nama = sc.nextLine();
34                      System.out.print(s:"Masukkan No HP pembeli: ");
35                      int noHP = sc.nextInt();
36                      Pembeli13 pbl = new Pembeli13(nama, noHP);
37                      pembeli.Enqueue(pbl);
38                      sc.nextLine();
39                      break;
40                  case 2:
41                      int pblKeluar = pembeli.Dequeue();
42                      if (pblKeluar != 0) {
43                          System.out.println("Antrian yang keluar: " + pblKeluar);
44                          break;
45                      }
46                  case 3:
47                      pembeli.print();
48                      break;
49                  case 4:
50                      pembeli.peek();
51                      break;
52                  case 5:
53                      pembeli.peekRear();
54                      break;
55                  case 6:
56                      System.out.print(s:"Masukkan nama Pembeli: ");
57                      nama = sc.nextLine();
58                      pembeli.peekPosition(nama);
59                      break;
60                  case 7:
61                      System.out.println(x:"Data Seluruh Pembeli");
62                      pembeli.daftarPembeli();
63                      break;
64              }
65          } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5 || pilih == 6 || pilih == 7);
66      }
67  }
68
```

NAMA	: Gilang Purnomo
NIM	: 2341720042
NO ABSEN	: 13
KELAS	: 1F
MATERI	: Queue

Output :

```
Masukkan kapasitas queue: 4
Pilih menu:
1. Pembeli baru
2. Pembeli Keluar
3. Cek semua Pembeli
4. Cek Pembeli terdepan
5. Cek Pembeli paling belakang
6. Cari Pembeli
7. Data seluruh Pembeli
----------------------
1
Masukkan Nama pembeli: Gilang
Masukkan No HP pembeli: 858532561
Pilih menu:
1. Pembeli baru
2. Pembeli Keluar
3. Cek semua Pembeli
4. Cek Pembeli terdepan
5. Cek Pembeli paling belakang
6. Cari Pembeli
7. Data seluruh Pembeli
----------------------
1
Masukkan Nama pembeli: Mita
Masukkan No HP pembeli: 857061051
Pilih menu:
1. Pembeli baru
2. Pembeli Keluar
3. Cek semua Pembeli
4. Cek Pembeli terdepan
5. Cek Pembeli paling belakang
6. Cari Pembeli
7. Data seluruh Pembeli
----------------------
1
Masukkan Nama pembeli: Obi
Masukkan No HP pembeli: 452189573
```

```
Pilih menu:
1. Pembeli baru
2. Pembeli Keluar
3. Cek semua Pembeli
4. Cek Pembeli terdepan
5. Cek Pembeli paling belakang
6. Cari Pembeli
7. Data seluruh Pembeli
----------------------
1
Masukkan Nama pembeli: brian
Masukkan No HP pembeli: 876900452
Pilih menu:
1. Pembeli baru
2. Pembeli Keluar
3. Cek semua Pembeli
4. Cek Pembeli terdepan
5. Cek Pembeli paling belakang
6. Cari Pembeli
7. Data seluruh Pembeli
----------------------
3
Gilang
Mita
Obi
brian
Jumlah elemen = 4
Pilih menu:
1. Pembeli baru
2. Pembeli Keluar
3. Cek semua Pembeli
4. Cek Pembeli terdepan
5. Cek Pembeli paling belakang
6. Cari Pembeli
7. Data seluruh Pembeli
----------------------
4
Elemen terdepan: Gilang
```

NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Queue

```
Pilih menu:
1. Pembeli baru
2. Pembeli Keluar
3. Cek semua Pembeli
4. Cek Pembeli terdepan
5. Cek Pembeli paling belakang
6. Cari Pembeli
7. Data seluruh Pembeli
----------------------
6
Masukkan nama Pembeli: Mita
Antrian ditemukan
Nama: Mita
No HP: 857061051
Posisi Antrian: 2
Pilih menu:
1. Pembeli baru
2. Pembeli Keluar
3. Cek semua Pembeli
4. Cek Pembeli terdepan
5. Cek Pembeli paling belakang
6. Cari Pembeli
7. Data seluruh Pembeli
----------------------
6
Masukkan nama Pembeli: obi
Antrian ditemukan
Nama: obi
No HP: 452189573
Posisi Antrian: 3
Pilih menu:
1. Pembeli baru
2. Pembeli Keluar
3. Cek semua Pembeli
4. Cek Pembeli terdepan
5. Cek Pembeli paling belakang
6. Cari Pembeli
7. Data seluruh Pembeli
----------------------
6
Masukkan nama Pembeli: Sifa
Antrian dengan nama Sifa tidak ditemukan.
```

```
Pilih menu:
1. Pembeli baru
2. Pembeli Keluar
3. Cek semua Pembeli
4. Cek Pembeli terdepan
5. Cek Pembeli paling belakang
6. Cari Pembeli
7. Data seluruh Pembeli
----------------------
7
Data Seluruh Pembeli
Gilang 858532561
Mita 857061051
Obi 452189573
brian 876900452
Pilih menu:
1. Pembeli baru
2. Pembeli Keluar
3. Cek semua Pembeli
4. Cek Pembeli terdepan
5. Cek Pembeli paling belakang
6. Cari Pembeli
7. Data seluruh Pembeli
----------------------
2
Mita
Obi
brian
Jumlah elemen = 3
Pilih menu:
1. Pembeli baru
2. Pembeli Keluar
3. Cek semua Pembeli
4. Cek Pembeli terdepan
5. Cek Pembeli paling belakang
6. Cari Pembeli
7. Data seluruh Pembeli
----------------------
3
Mita
Obi
brian
Jumlah elemen = 3
```

```
Pilih menu:
1. Pembeli baru
2. Pembeli Keluar
3. Cek semua Pembeli
4. Cek Pembeli terdepan
5. Cek Pembeli paling belakang
6. Cari Pembeli
7. Data seluruh Pembeli
----------------------
3
Mita
Obi
brian
Jumlah elemen = 3
```

```
Pilih menu:
1. Pembeli baru
2. Pembeli Keluar
3. Cek semua Pembeli
4. Cek Pembeli terdepan
5. Cek Pembeli paling belakang
6. Cari Pembeli
7. Data seluruh Pembeli
----------------------
7
Data Seluruh Pembeli
Mita 857061051
Obi 452189573
brian 876900452
```