



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : QUEUE

LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

10.2 Praktikum 1

```
package P10.Praktikum;

public class Queue25 {
    int[] data;
    int front;
    int rear;
    int size;
    int max;

    public Queue25(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");
        }
    }

    public void print() {
        if (isEmpty()) {
            System.out.println("Queue masih kosong");
        } else {
            int i = front;
            while (i != rear) {
                System.out.println(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(x:"Queue berhasil dikosongkan");
        } else {
            System.out.println(x:"Queue masih kosong");
        }
    }

    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++;
        }
    }

    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;
    }
}

package P10.Praktikum;

import java.util.Scanner;

public class QueueMain {
    public static void menu() {
        System.out.println("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:"-----");
    }

    public static void main(String[] args) {
        Scanner sc25 = new Scanner(System.in);

        System.out.print(s:"Masukkan kapasitas queue : ");
        int n = sc25.nextInt();

        Queue25 Q = new Queue25(n);

        int pilih;

        do {
            menu();
            pilih = sc25.nextInt();
            switch (pilih) {
                case 1:
                    System.out.print(s:"Masukkan data baru : ");
                    int dataMasuk = sc25.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 : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : QUEUE

10.2.2 Verifikasi Hasil Percobaan

```
PS D:\PrakASD_1F_25> & 'C:\Program Fi
'-cp' 'C:\Users\salsabila mahda\AppData
dhat.java\jdt_ws\PrakASD_1F_25_3f0e4e3
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
```

10.2.3 Question

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

Answer:

Atribut front dan rear bernilai -1 berguna untuk menunjukkan bahwa size nya masih dalam kondisi kosong. Sedangkan size bernilai 0, karena setiap array di mulai dari index ke 0.

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

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

Answer:

Potongan kode tersebut berguna jika rear berada pada posisi max-1 atau index terakhir dari array, maka saat ada penambahan data baru maka akan diletakkan pada index ke 0.

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

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

Answer:

Potongan kode tersebut berguna jika front berada pada posisi max-1 atau index terakhir dari array, maka disaat ada penghapusan data, front akan diletakkan pada index ke 0.



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : QUEUE

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

Answer:

Karena posisi front atau data terdepan tidak selalu pada indeks ke - 0, sedangkan perulangan dimulai dari posisi front nya.

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

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

Answer:

Potongan kode tersebut bermaksud jika nilai I (front) tidak sebagai rear, maka akan dilakukan penambahan pada variable tersebut lalu akan dimodulo dengan nilai max atau kapasitas Queue tersebut. Gunanya untuk mencegah melakukan print melebihi max, sehingga dilakukan modulo max.

6. Tunjukkan potongan kode program yang merupakan queue overflow!

Answer:

```
public void Enqueue(int dt) {  
    if (IsFull()) {  
        System.out.println("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!

Answer:

```
public void Enqueue(int dt) {  
    if (IsFull()) {  
        System.out.println("Queue sudah penuh");  
        System.exit(status:0); //MODIFIKASI PERTANYAAN NO 7  
    }  
}  
  
public int Dequeue() {  
    int dt = 0;  
    if (IsEmpty()) {  
        System.out.println("Queue masih kosong");  
        System.exit(status:0); //MODIFIKASI PERTANYAAN NO 7  
    }  
}
```

```
PS D:\PrakASD_1F_25> cd .\PrakASD_1F_25\ & .\QueueMain.exe  
Masukkan kapasitas queue : 1  
Masukkan operasi yang diinginkan :  
1. Enqueue  
2. Dequeue  
3. Print  
4. Peek  
5. Clear  
-----  
1  
Masukkan data baru : 32  
Masukkan operasi yang diinginkan :  
1. Enqueue  
2. Dequeue  
3. Print  
4. Peek  
5. Clear  
-----  
1  
Masukkan data baru : 25  
Queue sudah penuh  
PS D:\PrakASD_1F_25>
```

```
PS D:\PrakASD_1F_25> d:; cd 'd:\PrakASD_1F_25\ & .\QueueMain.exe  
Masukkan kapasitas queue : 1  
Masukkan operasi yang diinginkan :  
1. Enqueue  
2. Dequeue  
3. Print  
4. Peek  
5. Clear  
-----  
2  
Queue masih kosong  
PS D:\PrakASD_1F_25>
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : QUEUE

10.3 Praktikum 2

```
package P10.Praktikum2;  
  
public class Nasabah25 {  
    String norek, nama, alamat;  
    int umur;  
    double saldo;  
  
    Nasabah25(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;  
    }  
  
    Nasabah25(){  
    }  
}
```

```
package P10.Praktikum2;  
  
public class Queue25 {  
    Nasabah25[] data;  
    int front;  
    int rear;  
    int size;  
    int max;  
  
    public Queue25(int n){  
        max = n;  
        data = new Nasabah25[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].norek+" "+data[front].nama  
            +" "+data[front].alamat+" "+data[front].umur+" "+data[front].saldo);  
        }else{  
            System.out.println("Queue masih kosong");  
        }  
    }  
  
    public void print(){  
        if(!isEmpty()){  
            System.out.println("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);  
        }  
    }  
}
```

```
    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(Nasabah25 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++;  
        }  
    }  
  
    public Nasabah25 Dequeue() {  
        Nasabah25 dt = new Nasabah25();  
        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 : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : QUEUE

```
package P10.Praktikum;  
  
import java.util.Scanner;  
  
public class QueueMain {  
    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("-----");  
    }  
  
    RunIDebug  
    public static void main(String[] args) {  
        Scanner sc25 = new Scanner(System.in);  
  
        System.out.print("Masukkan kapasitas queue : ");  
        int jumlah = sc25.nextInt();  
        Queue<Antri> antri = new Queue<>(jumlah);  
  
        int pilih;  
  
        do {  
            menu();  
            pilih = sc25.nextInt();  
            sc25.nextLine();  
            switch (pilih) {  
                case 1:  
                    System.out.print("No Rekening: ");  
                    String norek = sc25.nextLine();  
                    System.out.print("Nama: ");  
                    String nama = sc25.nextLine();  
                    System.out.print("Alamat: ");  
                    String alamat = sc25.nextLine();  
                    System.out.print("Umur: ");  
                    int umur = sc25.nextInt();  
                    System.out.print("Saldo: ");  
                    int saldo = sc25.nextInt();  
                    Antri an = new Antri(norek, nama, alamat, umur, saldo);  
                    sc25.nextLine();  
                    antri.Enqueue(an);  
                    break;  
                case 2:  
                    Antri 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);  
    }  
}
```

10.3.2 Verifikasi Hasil Percobaan

```
Enter age (1-255): 23  
ikun2.QueueMain  
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 : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : QUEUE

10.3.3 Pertanyaan

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;
}
```

Answer:

Equal adalah membandingkan dua string, jika kedua string sama maka akan me retrun true, begitupun sebaliknya. !"".equals(data.norek) dst digunakan untuk mengecek apakah norek pada data tidak sama dengan string kosong ("") dst. Jika semua kondisi bernilai true, maka akan ditampilkan isi dari data tersebut.

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!

Answer:

```
//MODIFIKASI PERTANYAAN 2
public void peekRear(){
    if(!IsEmpty()){
        System.out.println("Elemen terdepan : "+data[rear].norek+ " "+data[rear].nama+ " "
            +data[rear].alamat+ " "+data[rear].umur+ " "+data[rear].saldo);
    }else{
        System.out.println(x:"Queue masih kosong");
    }
}

public class QueueMain {
    public static void menu() {
        System.out.println(x:"Pilih Menu : ");
        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 Paling Belakang"); //MODIFIKASI PERTANYAAN 2
        System.out.println(x:"-----");
    }
}

case 3:
    antri.peek();
    break;
case 4:
    antri.print();
    break;
//MODIFIKASI PERTANYAAN 2
case 5:
    antri.peekRear();
    break;
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : QUEUE

```
PS D:\PrakASD_1F_25>
Masukkan kapasitas queue : 5
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: Salsabila Mahda
Alamat: Bali
Umur: 19
Saldo: 3000000
Pilih Menu :
1. Antrian baru
2. Antrian keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian Paling Belakang
-----
1
No Rekening: 34567
Nama: Algi Pebrian
Alamat: Tabanan
Umur: 19
Saldo: 75000000
Pilih Menu :
1. Antrian baru
2. Antrian keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian Paling Belakang
-----
5
Elemen terdepan : 34567 Algi Pebrian Tabanan 19 7.5E7
PS D:\PrakASD_1F_25>
```

10.4 Tugas

1. Buatlah program antrian untuk mengilustrasikan 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



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : QUEUE

- 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

Answer :

```
P10 > Tugas > J Pembeli25.java > Pembeli25
1 package P10.Tugas;
2
3 public class Pembeli25 {
4     String nama;
5     int noHP;
6
7     Pembeli25(String nama, int noHP) {
8         this.nama = nama;
9         this.noHP = noHP;
10    }
11
12    Pembeli25(){
13
14    }
15
16 }
```

```
P10 > Tugas > J Queue.java > Queue > DequeDeq
1 package P10.Tugas;
2
3 public class Queue {
4     Pembeli25[] antrian;
5     int front, rear, size, max;
6
7     public Queue(int n) {
8         max = n;
9         antrian = new Pembeli25[max];
10        size = 0;
11        front = rear = -1;
12    }
13
14    public boolean isEmpty(){
15        if(size == 0){
16            return true;
17        }else{
18            return false;
19        }
20    }
21
22    public boolean IsFull() {
23        if (size == max) {
24            return true;
25        } else {
26            return false;
27        }
28    }
29
30    public void enqueue(Pembeli25 antri){
31        if(IsFull()){
32            System.out.println("Queue sudah penuh");
33        }else{
34            if(IsEmpty()){
35                front = rear = 0;
36            }else{
37                if (rear == max -1){
38                    rear = 0;
39                }else{
40                    rear++;
41                }
42            }
43            antrian[rear] = antri;
44            size++;
45        }
46    }
47 }
```

```
P10 > Tugas > J Queue.java > Queue > DequeDeq
48 public void peek(){
49     if(IsEmpty()){
50         System.out.println("Elemen terdepan : " + antrian[front].nama + " " + antrian[front].noHP);
51     }else{
52         System.out.println("Queue masih kosong");
53     }
54 }
55
56 public void peekRear(){
57     if(IsEmpty()){
58         System.out.println("Elemen yang belakang : " + antrian[rear].nama + " " + antrian[rear].noHP);
59     }else{
60         System.out.println("Queue masih kosong");
61     }
62 }
63
64 public void print(){
65     if(IsEmpty()){
66         System.out.println("Queue masih kosong");
67     }else{
68         int i = front;
69         while(i != rear){
70             System.out.println(antrian[i].nama + " | " + antrian[i].noHP);
71             i = (i + 1) % max;
72         }
73         System.out.println(antrian[i].nama + " | " + antrian[i].noHP);
74         System.out.println("Jumlah elemen = " + size);
75     }
76 }
77
78 public Pembeli25 Dequeue() {
79     Pembeli25 dt = new Pembeli25();
80     if (isEmpty()) {
81         System.out.println("Queue masih kosong");
82     } else {
83         dt = antrian[front];
84         size--;
85         if (isEmpty()) {
86             front = rear = -1;
87         } else {
88             if (front == max -1) {
89                 front = 0;
90             } else {
91                 front++;
92             }
93         }
94     }
95     return dt;
96 }
97 }
```




NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : QUEUE

```
P10 > Tugas > J Queue.java > Queue > enqueue(Pembeli25)
3 public class Queue {
98
99     public void peekPosition(String nama){
100         int i = front;
101         boolean j = true;
102         while (j) {
103             if (nama.equals(this.antrian[i].nama)) {
104                 System.out.println("data " + antrian[i].nama + " ada di antrian ke-" + i);
105                 j = false;
106             }
107             i++;
108         }
109     }
110
111     public void daftarPembeli(){
112         if (isEmpty()){
113             System.out.println("Queue masih kosong");
114         } else {
115             int i = front;
116             while (i != rear){
117                 System.out.println(antrian[i].nama + " | " + antrian[i].noHP);
118                 i = (i + 1) % max;
119             }
120             System.out.println(antrian[i].nama + " | " + antrian[i].noHP);
121             System.out.println("Jumlah elemen = " + size);
122         }
123     }
124
125
126
127 }
```

```
P10 > Tugas > J QueueMain.java > QueueMain > menu()
1 package P10.tugas;
2 import java.util.Scanner;
3
4 public class QueueMain {
5     public static void menu(){
6         System.out.println("Pilih Menu ");
7         System.out.println("1. Antrian baru");
8         System.out.println("2. Antrian Keluar");
9         System.out.println("3. Cek Antrian Terdepan");
10        System.out.println("4. Cek Antrian Terbelakang");
11        System.out.println("5. Cek Semua Antrian");
12        System.out.println("6. Cek Posisi Pembeli Berdasarkan Nama");
13        System.out.println("7. Menampilkan data seluruh pembeli");
14        System.out.println("-----");
15    }
16
17    public static void main(String[] args) {
18        Scanner sc25 = new Scanner(System.in);
19        System.out.print("Masukkan kapasitas queue : ");
20        int jumlah = sc25.nextInt();
21        Queue antri = new Queue(jumlah);
22
23        int pilih;
24        do {
25            menu();
26            pilih = sc25.nextInt();
27            sc25.nextLine();
28
29            switch (pilih) {
30                case 1:
31                    System.out.print("Nama Pembeli : ");
32                    String nama = sc25.nextLine();
33                    System.out.print("Nomor HP Pembeli : ");
34                    int noHP = sc25.nextInt();
35                    Pembeli25 nb = new Pembeli25(nama, noHP);
36                    antri.enqueue(nb);
37                    break;
38                case 2:
39                    Pembeli25 data = antri.Dequeue();
40                    if (!"".equals(data.nama) && !"".equals(data.noHP)) {
41                        System.out.println("Antrian yang keluar : " + data.nama + " " + data.noHP);
42                    }
43                    break;
44                case 3:
45                    antri.peak();
46                    break;
47                case 4:
48                    antri.peakRear();
49                    break;
50                case 5:
51                    antri.print();
52                    break;
53                case 6:
54                    System.out.print("Masukkan nama Pembeli : ");
55                    String namaPembeli = sc25.nextLine();
56                    antri.peakPosition(namaPembeli);
57                    break;
58                case 7:
59                    antri.daftarPembeli();
60                    break;
61            }
62        } while (pilih >= 1 && pilih <= 8);
63    }
64 }
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : QUEUE

```
PS D:\PrakASD_1F_25> & 'C:\Program Files\Java
.Tugas.QueueMain'
Masukkan kapasitas queue : 7
Pilih Menu
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Antrian Terbelakang
5. Cek Semua Antrian
6. Cek Posisi Pembeli Berdasarkan Nama
7. Menampilkan data seluruh pembeli
-----
1
Nama Pembeli : Algi
Nomor HP Pembeli : 123456789
Pilih Menu
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Antrian Terbelakang
5. Cek Semua Antrian
6. Cek Posisi Pembeli Berdasarkan Nama
7. Menampilkan data seluruh pembeli
-----
1
Nama Pembeli : Salsa
Nomor HP Pembeli : 134567829
Pilih Menu
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Antrian Terbelakang
5. Cek Semua Antrian
6. Cek Posisi Pembeli Berdasarkan Nama
7. Menampilkan data seluruh pembeli
-----
1
Nama Pembeli : Abi
Nomor HP Pembeli : 213452617
Pilih Menu
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Antrian Terbelakang
5. Cek Semua Antrian
6. Cek Posisi Pembeli Berdasarkan Nama
7. Menampilkan data seluruh pembeli
-----
2
Antrian yang keluar : Algi 123456789

Pilih Menu
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Antrian Terbelakang
5. Cek Semua Antrian
6. Cek Posisi Pembeli Berdasarkan Nama
7. Menampilkan data seluruh pembeli
-----
3
Elemen terdepan : Salsa 134567829
Pilih Menu
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Antrian Terbelakang
5. Cek Semua Antrian
6. Cek Posisi Pembeli Berdasarkan Nama
7. Menampilkan data seluruh pembeli
-----
4
Elemen yang belakang : Abi 213452617
Pilih Menu
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Antrian Terbelakang
5. Cek Semua Antrian
6. Cek Posisi Pembeli Berdasarkan Nama
7. Menampilkan data seluruh pembeli
-----
5
Salsa | 134567829
Abi | 213452617
Jumlah elemen = 2
Pilih Menu
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Antrian Terbelakang
5. Cek Semua Antrian
6. Cek Posisi Pembeli Berdasarkan Nama
7. Menampilkan data seluruh pembeli
-----
6
Masukkan nama Pembeli : Abi
data Abi ada di antrian ke-2
```

```
Pilih Menu
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Antrian Terbelakang
5. Cek Semua Antrian
6. Cek Posisi Pembeli Berdasarkan Nama
7. Menampilkan data seluruh pembeli
-----
7
Salsa | 134567829
Abi | 213452617
Jumlah elemen = 2
Pilih Menu
1. Antrian baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Antrian Terbelakang
5. Cek Semua Antrian
6. Cek Posisi Pembeli Berdasarkan Nama
7. Menampilkan data seluruh pembeli
-----
0
PS D:\PrakASD_1F_25>
```