



NAMA : Roy Wijaya
NIM : 2341720120
KELAS : 1-G
MATERI : Queue (Jobsheet 8)

8.2 Praktikum 1

```
1 package p10;  
2  
3 import java.util.Scanner;  
4  
5 public class QueueMain27 {  
6     public static void main(String[] args) {  
7         Scanner sc = new Scanner(System.in);  
8  
9         System.out.print("Masukkan kapasitas queue: ");  
10        int n = sc.nextInt();  
11        Queue27 Q = new Queue27(n);  
12        int pilih;  
13        do {  
14            menu();  
15            pilih = sc.nextInt();  
16            switch (pilih) {  
17                case 1:  
18                    System.out.print("Masukkan data baru: ");  
19                    int dataMasuk = sc.nextInt();  
20                    Q.Enqueue(dataMasuk);  
21                    break;  
22                case 2:  
23                    int dataKeluar = Q.Dequeue();  
24                    if (dataKeluar != 0) {  
25                        System.out.println("Data yang dikeluarkan: " + dataKeluar);  
26                        break;  
27                    }  
28  
29                case 3:  
30                    Q.print();  
31                    break;  
32  
33                case 4:  
34                    Q.peek();  
35                    break;  
36  
37                case 5:  
38                    Q.clear();  
39                    break;  
40  
41            }  
42        } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5);  
43  
44    }  
45  
46    public static void menu() {  
47        System.out.println("Masukkan operasi yang diinginkan:");  
48        System.out.println("1. Enqueue");  
49        System.out.println("2. Dequeue");  
50        System.out.println("3. Print");  
51        System.out.println("4. Peek");  
52        System.out.println("5. Clear");  
53        System.out.println("-----");  
54    }  
55 }  
56
```



NAMA : Roy Wijaya
NIM : 2341720120
KELAS : 1-G
MATERI : Queue (Jobsheet 8)

```
1 package P10;  
2  
3 public class Queue27 {  
4     int[] data;  
5     int front;  
6     int rear;  
7     int size;  
8     int max;  
9  
10    public Queue27(int n) {  
11        max = n;  
12        data = new int[max];  
13        size = 0;  
14        front = rear = -1;  
15    }  
16  
17    boolean isEmpty() {  
18        if (size == 0) {  
19            return true;  
20        } else {  
21            return false;  
22        }  
23    }  
24  
25    boolean isFull() {  
26        if (size == max) {  
27            return true;  
28        } else {  
29            return false;  
30        }  
31    }  
32  
33    public void peek() {  
34        if (!isEmpty()) {  
35            System.out.println("Elemen terdepan: " + data[front]);  
36        } else {  
37            System.out.println("Queue masih kosong");  
38        }  
39    }  
40  
41  
42    public void print() {  
43        if (!isEmpty()) {  
44            System.out.println("Queue masih kosong");  
45        } else {  
46            int i = front;  
47            while (i != rear) {  
48                System.out.print(data[i] + " ");  
49                i = (i + 1) % max;  
50            }  
51            System.out.println(data[i] + " ");  
52            System.out.println("Jumlah elemen = " + size);  
53        }  
54    }  
55  
56    public void clear() {  
57        if (!isEmpty()) {  
58            front = rear = -1;  
59            size = 0;  
60            System.out.println("Queue berhasil dikosongkan");  
61        } else {  
62            System.out.println("Queue masih kosong");  
63        }  
64    }  
65  
66    public void Enqueue(int dt) {  
67        if (isFull()) {  
68            System.out.println("Queue sudah penuh");  
69        } else {  
70            if (isEmpty()) {  
71                front = rear = 0;  
72            } else {  
73                if (rear == max - 1) {  
74                    rear = 0;  
75                } else {  
76                    rear++;  
77                }  
78            }  
79            data[rear] = dt;  
80            size++;  
81        }  
82    }  
83  
84    public int Dequeue() {  
85        int dt = 0;  
86        if (!isEmpty()) {  
87            System.out.println("Queue masih kosong");  
88        } else {  
89            dt = data[front];  
90            size--;  
91            if (isEmpty()) {  
92                front = rear = -1;  
93            } else {  
94                if (front == max - 1) {  
95                    front = 0;  
96                } else {  
97                    front++;  
98                }  
99            }  
100        }  
101        return dt;  
102    }  
103 }  
104 }  
105 }
```

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
-



NAMA : Roy Wijaya
NIM : 2341720120
KELAS : 1-G
MATERI : Queue (Jobsheet 8)

8.2.3 Pertanyaan

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

Jawab:

Front dan rear bernilai -1 karena belum ada data, sedangkan size bernilai 0 karena banyak data adalah 0/ belum ada input data.

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

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

Jawab:

Dalam metode Enqueue, penggunaan kondisi if (rear == max - 1) bertujuan untuk memastikan apakah rear (ujung) dari antrian telah menyentuh batas akhir dari array yang dijadikan tempat penyimpanan elemen-elemen antrian. Bila rear sudah berada di posisi terakhir dari array, rear harus diset ulang menjadi 0, memungkinkan penambahan elemen baru ke dalam antrian.

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

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

Jawab: Dalam metode Dequeue, penggunaan kondisi if (front == max - 1) bertujuan untuk memverifikasi apakah penunjuk front pada antrian sudah berada di ujung terakhir array yang dipergunakan untuk menampung elemen-elemen antrian. Apabila front sudah berada pada posisi akhir, maka front diatur ulang menjadi 0.

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

Jawab:

Variabel i tidak dimulai dari 0 melainkan dimulai dari front karena konsep method tersebut menampilkan data antrian dari antrian terdepan sampai antrian paling belakang, dimana antrian depan tidak selalu dimulai dari 0 dan antrian belakang tidak selalu dimulai dari max-1.



NAMA : Roy Wijaya
NIM : 2341720120
KELAS : 1-G
MATERI : Queue (Jobsheet 8)

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

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

Jawab:

Kode tersebut digunakan untuk mengatur variabel i agar maju kembali ke depan ketika i sudah berada pada indeks tertinggi array

6. Tunjukkan potongan kode program yang merupakan queue overflow!

Jawab:

```
if (isFull()) {  
    System.out.println(x:"Queue sudah penuh");  
} else {
```

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:

```
1 public void Enqueue(int dt) {  
2     if (isFull()) {  
3         System.out.println("Queue sudah penuh");  
4         System.exit(0);  
5     } else {  
6         if (isEmpty()) {  
7             front = rear = 0;  
8         } else {  
9             if (rear == max - 1) {  
10                rear = 0;  
11            } else {  
12                rear++;  
13            }  
14        }  
15        data[rear] = dt;  
16        size++;  
17    }  
18 }
```

```
1 public int Dequeue() {  
2     int dt = 0;  
3     if (isEmpty()) {  
4         System.out.println("Queue masih kosong ");  
5         System.exit(0);  
6     } else {  
7         dt = data[front];  
8         size--;  
9         if (isEmpty()) {  
10            front = rear = -1;  
11        } else {  
12            if (front == max - 1) {  
13                front = 0;  
14            } else {  
15                front++;  
16            }  
17        }  
18    }  
19    return dt;  
20 }  
21 }  
22 }
```



NAMA : Roy Wijaya
NIM : 2341720120
KELAS : 1-G
MATERI : Queue (Jobsheet 8)

8.3 Praktikum 2

Masukkan kapasitas queue: 8

Pilih menu:

1. Antrian Baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian

1

Masukkan No Rekening: 12345

Masukkan nama: Dewi

Masukkan alamat: Malang

Masukkan umur: 23

Masukkan saldo: 1300000

Pilih menu:

1. Antrian Baru
2. Antrian Keluar
3. Cek Antrian Terdepan
4. Cek Semua Antrian

1

Masukkan No Rekening: 32940

Masukkan nama: Susan

Masukkan alamat: Surabaya

Masukkan umur: 39

Masukkan 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

```
1 package #10.Praktikum2;
2
3 public class Queue27 {
4     Nasabah27[] data;
5     int front;
6     int rear;
7     int size;
8     int max;
9
10    public Queue27(int n) {
11        max = n;
12        data = new Nasabah27[max];
13        size = 0;
14        front = rear = -1;
15    }
16
17    boolean isEmpty() {
18        if (size == 0) {
19            return true;
20        } else {
21            return false;
22        }
23    }
24
25    boolean isFull() {
26        if (size == max) {
27            return true;
28        } else {
29            return false;
30        }
31    }
32
33    public void peek() {
34        if (!isEmpty()) {
35            System.out.println("Elemen terdepan: " + data[front].norek + " " + data[front].nama + " "
36                               + data[front].alamat + " " + data[front].umur + " " + data[front].saldo);
37        } else {
38            System.out.println("Queue masih kosong.");
39        }
40    }
41
42    public void print() {
43        if (isEmpty()) {
44            System.out.println("Queue masih kosong");
45        } else {
46            int i = front;
47            while (i != rear) {
48                System.out.println(data[i].norek + " " + data[i].nama + " " + data[i].alamat + " " + data[i].umur + " "
49                                   + data[i].saldo);
50                i = (i + 1) % max;
51            }
52            System.out.println(data[i].norek + " " + data[i].nama + " " + data[i].alamat + " " + data[i].umur + " "
53                               + data[i].saldo);
54            System.out.println("Jumlah elemen = " + size);
55        }
56    }
57
58    public void clear() {
59        if (!isEmpty()) {
60            front = rear = -1;
61            size = 0;
62            System.out.println("Queue berhasil dikosongkan");
63        } else {
64            System.out.println("Queue masih kosong");
65        }
66    }
67
68    public void enqueue(Nasabah27 dt) {
69        if (isFull()) {
70            System.out.println("Queue sudah penuh");
71        } else {
72            if (isEmpty()) {
73                front = rear = 0;
74            } else {
75                if (rear == max - 1) {
76                    rear = 0;
77                } else {
78                    rear++;
79                }
80            }
81            data[rear] = dt;
82            size++;
83        }
84    }
85
86    public Nasabah27 dequeue() {
87        Nasabah27 dt = new Nasabah27();
88        if (isEmpty()) {
89            System.out.println("Queue masih kosong");
90        } else {
91            dt = data[front];
92            size--;
93            if (isEmpty()) {
94                front = rear = -1;
95            } else {
96                if (front == max - 1) {
97                    front = 0;
98                } else {
99                    front++;
100                }
101            }
102        }
103        return dt;
104    }
105
106    }
107
108 }
109
110
```



NAMA : Roy Wijaya
NIM : 2341720120
KELAS : 1-G
MATERI : Queue (Jobsheet 8)

```
1 package p10.praktikum2;  
2  
3 public class Nasabah27 {  
4     String norek;  
5     String nama;  
6     String alamat;  
7     int umur;  
8     double saldo;  
9  
10    public Nasabah27() {  
11  
12    }  
13  
14    Nasabah27(String norek, String nama, String alamat, int umur, double saldo) {  
15        this.norek = norek;  
16        this.nama = nama;  
17        this.alamat = alamat;  
18        this.umur = umur;  
19        this.saldo = saldo;  
20    }  
21 }  
22 }  
23
```




NAMA : Roy Wijaya
NIM : 2341720120
KELAS : 1-G
MATERI : Queue (Jobsheet 8)

```
1 package P10.Praktikum2;
2
3 import java.util.Scanner;
4
5 public class QueueMain {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         System.out.print("Masukkan kapasitas queue: ");
9         int max = sc.nextInt();
10        Queue27 antri = new Queue27(max);
11        int pilih;
12
13        do {
14            menu();
15            pilih = sc.nextInt();
16
17            switch (pilih) {
18                case 1:
19                    sc.nextLine();
20                    System.out.print("Masukkan No Rekening: ");
21                    String norek = sc.nextLine();
22                    System.out.print("Masukkan nama: ");
23                    String nama = sc.nextLine();
24                    System.out.print("Masukkan alamat: ");
25                    String alamat = sc.nextLine();
26                    System.out.print("Masukkan umur: ");
27                    int umur = sc.nextInt();
28                    System.out.print("Masukkan saldo: ");
29                    double saldo = sc.nextDouble();
30                    Nasabah27 nb = new Nasabah27(norek, nama, alamat, umur, saldo);
31                    sc.nextLine();
32                    antri.Enqueue(nb);
33                    break;
34                case 2:
35                    Nasabah27 data = antri.Dequeue();
36                    if (!"".equals(data.norek) && !"".equals(data.nama) && !"".equals(data.alamat) && data.umur != 0
37                        && data.saldo != 0) {
38                        System.out.println("Antrian yang keluar: " + data.norek + " " + data.nama + " " + data.alamat
39                            + " " + data.umur + " " + data.saldo);
40                    }
41                    break;
42                case 3:
43                    antri.peek();
44                    break;
45                case 4:
46                    antri.print();
47                    break;
48            }
49        } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4);
50    }
51
52    public static void menu() {
53        System.out.println("Pilih menu: ");
54        System.out.println("1. Antrian Baru");
55        System.out.println("2. Antrian Keluar");
56        System.out.println("3. Cek Antrian Terdepan");
57        System.out.println("4. Cek Semua Antrian");
58        System.out.println("-----");
59    }
60 }
61
62
```



NAMA : Roy Wijaya
NIM : 2341720120
KELAS : 1-G
MATERI : Queue (Jobsheet 8)

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:

!"".equals(data.norek): Memastikan bahwa nomor rekening (norek) tidak kosong.

!"".equals(data.nama): Memastikan bahwa nama nasabah tidak kosong.

!"".equals(data.alamat): Memastikan bahwa alamat nasabah tidak kosong.

data.umur != 0: Memastikan bahwa umur nasabah tidak nol.

data.saldo != 0: Memastikan bahwa saldo nasabah tidak nol.

Lalu jika kondisi tersebut memenuhi maka program akan mencetak informasi nasabah 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!

Jawab:

```
1 public void peekRear() {
2     if (!isEmpty()) {
3         System.out.println("Elemen terdepan: " + data[rear].norek + " " + data[rear].nama + " "
4             + data[rear].alamat + " " + data[rear].umur + " " + data[rear].saldo);
5     } else {
6         System.out.println("Queue masih kosong ");
7     }
8 }
```




NAMA : Roy Wijaya
NIM : 2341720120
KELAS : 1-G
MATERI : Queue (Jobsheet 8)

```
1         break;
2     case 4:
3         antri.print();
4         break;
5     case 5:
6         antri.peakRear();
7         break;
8
9     }
10 } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5);
11 }
12
13 public static void menu() {
14     System.out.println("Pilih menu: ");
15     System.out.println("1. Antrian Baru");
16     System.out.println("2. Antrian Keluar");
17     System.out.println("3. Cek Antrian Terdepan");
18     System.out.println("4. Cek Semua Antrian");
19     System.out.println("5. Cek Antrian Paling Belakang");
20     System.out.println("-----");
21 }
```

LINK GITHUB: https://github.com/RoyW12/AlgoritmaStrukturData_1G_28