



NAMA : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
MATERI : QUEUE

Percobaan 1

```
1 package p9.praktikum1;
2
3 public class Queue15 {
4     int[] data;
5     int front;
6     int rear;
7     int size;
8     int max;
9
10    Queue15(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 terdapat: " + data[front]);
36        } else {
37            System.out.println("Queue masih kosong");
38        }
39    }
40
41    void print() {
42        if (isEmpty()) {
43            System.out.println("Queue masih kosong");
44        } else {
45            int i = front;
46            while (i != rear) {
47                System.out.println(data[i] + " ");
48                i = (i + 1) % max;
49            }
50            System.out.println(data[i] + " ");
51            System.out.println("Jumlah elemen = " + size);
52        }
53    }
54
55    void clear() {
56        if (!isEmpty()) {
57            front = rear = -1;
58            size = 0;
59            System.out.println("Queue berhasil dikosongkan");
60        } else {
61            System.out.println("Queue masih kosong");
62        }
63    }
64
65    void Enqueue(int dt) {
66        if (IsFull()) {
67            System.out.println("Queue sudah penuh");
68        } else {
69            if (isEmpty()) {
70                front = rear = 0;
71            } else {
72                if (rear == max - 1) {
73                    rear = 0;
74                } else {
75                    rear++;
76                }
77            }
78            data[rear] = dt;
79            size++;
80        }
81    }
82
83    int Dequeue() {
84        int dt = 0;
85        if (isEmpty()) {
86            System.out.println("Queue masih kosong");
87        } else {
88            dt = data[front];
89            size--;
90            if (isEmpty()) {
91                front = rear = -1;
92            } else {
93                if (front == max - 1) {
94                    front = 0;
95                } else {
96                    front++;
97                }
98            }
99        }
100        return dt;
101    }
102 }
103
104 }
```



NAMA : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
MATERI : QUEUE

```
1 package P9.Praktikum1;
2
3 import java.util.Scanner;
4
5 public class QueueMain15 {
6
7     static Scanner kiak = new Scanner(System.in);
8
9     public static void menu() {
10         System.out.println("Masukkan operasi yang diinginkan:");
11         System.out.println("1. Enqueue");
12         System.out.println("2. Dequeue");
13         System.out.println("3. Print");
14         System.out.println("4. Peek");
15         System.out.println("5. Clear");
16         System.out.println("-----");
17     }
18     public static void main(String[] args) {
19
20         System.out.print("Masukkan kapasitas queue: ");
21         int n = kiak.nextInt();
22
23         Queue15 q = new Queue15(n);
24         int pilih;
25
26         do {
27             menu();
28             pilih = kiak.nextInt();
29             switch (pilih) {
30                 case 1:
31                     System.out.print("Masukkan data baru: ");
32                     int dataMasuk = kiak.nextInt();
33                     q.Enqueue(dataMasuk);
34                     break;
35
36                 case 2:
37                     int dataKeluar = q.Dequeue();
38                     if (dataKeluar != 0) {
39                         System.out.println("Data yang dikeluarkan: " + dataKeluar);
40                     }
41                     break;
42
43                 case 3:
44                     q.print();
45                     break;
46
47                 case 4:
48                     q.peek();
49                     break;
50
51                 case 5:
52                     q.clear();
53                     break;
54
55                 default:
56                     break;
57             }
58         } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5);
59     }
60 }
61
```



NAMA : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
MATERI : QUEUE

```
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 terdapat: 15
```

Pertanyaan :

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

Jawab : Nilai -1 untuk front dan rear menandakan bahwa antrian kosong, sedangkan nilai 0 untuk size menunjukkan bahwa tidak ada elemen dalam antrian pada awalnya.

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

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

Jawab : Kode if (rear == max - 1) { rear = 0; } dimaksudkan untuk mengecek apakah pointer rear sudah mencapai batas maksimum antrian, yakni max - 1. Jika demikian, maka pointer rear akan diposisikan kembali ke indeks awal, yaitu 0.

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

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

Jawab : Ketika pointer front mencapai indeks tertinggi dalam representasi array untuk antrian, kode if (front == max - 1) { front = 0; } akan dieksekusi. Tujuannya adalah untuk mengatur kembali nilai pointer front ke indeks awal (0) setelah mencapai batas akhir array. Hal ini dilakukan agar antrian dapat terus beroperasi secara siklik, memungkinkan penghapusan elemen dari depan antrian dilanjutkan dari indeks awal array tanpa adanya batasan.



NAMA : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
MATERI : QUEUE

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

Jawab : Inisialisasi variabel i dengan nilai front bertujuan untuk memastikan bahwa proses pencetakan elemen-elemen dalam antrian dimulai dari elemen yang sebenarnya ada di depan antrian. Dengan demikian, urutan elemen yang dicetak akan sesuai dengan urutan sebenarnya dari elemen-elemen tersebut di dalam antrian, tanpa memasukkan elemen-elemen yang berada sebelum posisi front yang belum valid.

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

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

Jawab : Kode `i = (i + 1) % max;` digunakan untuk mengiterasi elemen-elemen dalam antrian secara berurutan. Operator modulus `%` memastikan bahwa nilai i akan kembali ke 0 setelah mencapai indeks maksimum (`max - 1`). Dengan cara ini, iterasi dapat berlanjut secara siklik dari indeks awal hingga indeks akhir antrian, memungkinkan akses terhadap setiap elemen dalam antrian tanpa melampaui batas array yang digunakan untuk merepresentasikannya.

6. Tunjukkan potongan kode program yang merupakan queue overflow!

Jawab :

```
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++;  
}
```

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 :

```
if (IsFull()) {  
    System.out.println(x:"Queue sudah penuh");  
    System.exit(status:0);  
} else {
```



NAMA : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
MATERI : QUEUE

```
if (IsEmpty()) {  
    System.out.println(x:"Queue masih kosong");  
    System.exit(status:0);  
} else {
```

Percobaan 2

```
1 package P9.Praktikum2;  
2  
3 public class Queue15 {  
4     Nasabah15[] data;  
5     int front;  
6     int rear;  
7     int size;  
8     int max;  
9  
10    public Queue15(int n) {  
11        max = n;  
12        data = new Nasabah15[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 print() {  
34        if (isEmpty()) {  
35            System.out.println("Queue masih kosong");  
36        } else {  
37            int i = front;  
38            while (i != rear) {  
39                System.out.println(data[i].norek + " " + data[i].nama + " " + data[i].alamat + " " + data[i].umur + " " + data[i].saldo);  
40                i = (i + 1) % max;  
41            }  
42            System.out.println(data[i].norek + " " + data[i].nama + " " + data[i].alamat + " " + data[i].umur + " " + data[i].saldo);  
43            System.out.println("Jumlah elemen = " + size);  
44        }  
45    }  
46  
47    public void clear() {  
48        if (!isEmpty()) {  
49            front = rear = -1;  
50            size = 0;  
51            System.out.println("Queue berhasil dikosongkan");  
52        } else {  
53            System.out.println("Queue masih kosong");  
54        }  
55    }  
56  
57    public void Enqueue(Nasabah15 dt) {  
58        if (isFull()) {  
59            System.out.println("Queue sudah penuh");  
60        } else {  
61            if (isEmpty()) {  
62                front = rear = 0;  
63            } else {  
64                if (rear == max - 1) {  
65                    rear = 0;  
66                } else {  
67                    rear++;  
68                }  
69                data[rear] = dt;  
70                size++;  
71            }  
72        }  
73    }  
74  
75    public Nasabah15 Dequeue() {  
76        Nasabah15 dt = new Nasabah15();  
77        if (isEmpty()) {  
78            System.out.println("Queue masih kosong ");  
79        } else {  
80            dt = data[front];  
81            size--;  
82            if (isEmpty()) {  
83                front = rear = -1;  
84            } else {  
85                if (front == max - 1) {  
86                    front = 0;  
87                } else {  
88                    front++;  
89                }  
90            }  
91        }  
92        return dt;  
93    }  
94  
95    public void peek() {  
96        if (isEmpty()) {  
97            System.out.println("Elemen terdepan: " + data[front].norek + " " + data[front].nama + " " + data[front].alamat + " " + data[front].umur + " " + data[front].saldo);  
98        } else {  
99            System.out.println("Queue masih kosong ");  
100        }  
101    }  
102  
103 }  
104  
105 }  
106  
107 }
```



NAMA : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
MATERI : QUEUE

```
1 package P9.Praktikum2;
2
3 import java.util.Scanner;
4
5 public class QueueMain15 {
6
7     static Scanner kiak = new Scanner(System.in);
8     public static void main(String[] args) {
9         System.out.print("Masukkan kapasitas queue: ");
10        int max = kiak.nextInt();
11        Queue15 antri = new Queue15(max);
12        int pilih;
13
14        do {
15            menu();
16            pilih = kiak.nextInt();
17
18            switch (pilih) {
19                case 1:
20                    kiak.nextLine();
21                    System.out.print("Masukkan No Rekening: ");
22                    String norek = kiak.nextLine();
23                    System.out.print("Masukkan nama: ");
24                    String nama = kiak.nextLine();
25                    System.out.print("Masukkan alamat: ");
26                    String alamat = kiak.nextLine();
27                    System.out.print("Masukkan umur: ");
28                    int umur = kiak.nextInt();
29                    System.out.print("Masukkan saldo: ");
30                    double saldo = kiak.nextDouble();
31                    Nasabah15 nb = new Nasabah15(norek, nama, alamat, umur, saldo);
32                    kiak.nextLine();
33                    antri.Enqueue(nb);
34                    break;
35                case 2:
36                    Nasabah15 data = antri.Dequeue();
37                    if (!"".equals(data.norek) && !"".equals(data.nama) && !"".equals(data.alamat) && data.umur != 0
38                        && data.saldo != 0) {
39                        System.out.println("Antrian yang keluar: " + data.norek + " " + data.nama + " " + data.alamat
40                            + " " + data.umur + " " + data.saldo);
41                    }
42                    break;
43                case 3:
44                    antri.peek();
45                    break;
46                case 4:
47                    antri.print();
48                    break;
49            }
50        } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4);
51    }
52
53    public static void menu() {
54        System.out.println("Pilih menu: ");
55        System.out.println("1. Antrian Baru");
56        System.out.println("2. Antrian Keluar");
57        System.out.println("3. Cek Antrian Terdepan");
58        System.out.println("4. Cek Semua Antrian");
59        System.out.println("5. Cek Antrian Paling Belakang");
60        System.out.println("-----");
61    }
62 }
63
64
```



NAMA : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
MATERI : QUEUE

```
1 package p9.Praktikum2;
2
3 public class Nasabah15 {
4     String norek;
5     String nama;
6     String alamat;
7     int umur;
8     double saldo;
9
10    public Nasabah15() {
11
12    }
13
14    Nasabah15(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    }
24 }
25
```

```
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
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
5. Cek Antrian Paling Belakang
-----
1
```



NAMA : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
MATERI : QUEUE

```
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
5. Cek Antrian Paling Belakang
-----
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
5. Cek Antrian Paling Belakang
-----
```

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;
}
```

Jawab : Kode tersebut memverifikasi bahwa setiap properti dari objek data memiliki nilai yang valid dan bukan kosong. Selain itu, ia juga memastikan bahwa nilai properti umur dan saldo tidak bernilai 0. Apabila semua persyaratan ini terpenuhi, maka informasi mengenai data yang keluar dari antrian akan ditampilkan, dan perulangan akan dihentikan dengan perintah break.

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 terdepan: " + data[rear].norek + " " + data[rear].nama + " " + data[rear].alamat + " " + data[rear].umur + " " + data[rear].saldo);
    } else {
        System.out.println(x:"Queue masih kosong ");
    }
}
```

```
break;
case 5:
    antri.peekRear();
    break;
}
} while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5);
}
```