



NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : Queue (Jobsheet 8)

1. Praktikum 1

```
src > P9 > Praktikum1 > Queue.java > Queue > Dequeue()
1 package P9.Praktikum1;
2
3 public class Queue {
4     int[] data;
5     int front;
6     int rear;
7     int size;
8     int max;
9
10    public Queue(int n) {
11        max = n;
12        data = new int[max];
13        size = 0;
14        front = rear = -1;
15    }
16
17    public boolean isEmpty() {
18        if (size == 0) {
19            return true;
20        } else {
21            return false;
22        }
23    }
24
25    public 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(x:"Queue masih kosong.");
38        }
39    }
40
41    public void print() {
42        if (!isEmpty()) {
43            int i = front;
44            do {
45                System.out.print(data[i] + " ");
46                i = (i + 1) % max;
47            } while (i != (rear + 1) % max);
48            System.out.println();
49            System.out.println("Jumlah elemen = " + size);
50        } else {
51            System.out.println(x:"Queue masih kosong.");
52        }
53    }
54 }
```

```
src > P9 > Praktikum1 > QueueMain
3 public class Queue {
40
47    public void clear() {
48        if (!isEmpty()) {
49            front = rear = -1;
50            size = 0;
51            System.out.println(x:"Queue berhasil dikosongkan.");
52        } else {
53            System.out.println(x:"Queue sudah kosong.");
54        }
55    }
56
57    public void Enqueue(int dt) {
58        if (!IsFull()) {
59            if (isEmpty()) {
60                front = rear = 0;
61            } else {
62                rear = (rear + 1) % max;
63            }
64            data[rear] = dt;
65            size++;
66        } else {
67            System.out.println(x:"Queue sudah penuh.");
68        }
69    }
70
71    public int Dequeue() {
72        int dt = 0;
73        if (!isEmpty()) {
74            dt = data[front];
75            size--;
76            if (size == 0) {
77                front = rear = -1;
78            } else {
79                front = (front + 1) % max;
80            }
81        } else {
82            System.out.println(x:"Queue masih kosong.");
83        }
84        return dt;
85    }
86 }
```



NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : Queue (Jobsheet 8)

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

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS D:\college\semester 2\Algoritma dan Struktur Data_16_01> & "C:\Program  
Files\Java\jdk-20\bin\java.exe" -D%ShowCodeDetails%=-D%ShowCodeDetails%=-D%  
ShowCodeDetails%=-D%ShowCodeDetails%=-D%ShowCodeDetails% -cp "D:\col  
lege\semester 2\Algoritma dan Struktur Data_16_01\bin" P9.Praktikum1.QueueMain  
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.  
-----  
2  
data yang dikeluarkan 15
```

```
Masukkan operasi yang diinginkan :  
1. Enqueue.  
2. Dequeue.  
3. Print.  
4. Peek.  
5. Clear.  
-----  
3  
31  
Jumlah elemen = 1  
Masukkan operasi yang diinginkan :  
1. Enqueue.  
2. Dequeue.  
3. Print.  
4. Peek.  
5. Clear.  
-----  
4  
Elemen terdepan : 31  
Masukkan operasi yang diinginkan :  
1. Enqueue.  
2. Dequeue.  
3. Print.  
4. Peek.  
5. Clear.  
-----  
5  
Queue berhasil dikosongkan.
```



NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : Queue (Jobsheet 8)

Pertanyaan!

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

Jawab : Nilai awal atribut front dan rear nilainya -1 karena untuk menunjukkan queue awalnya tidak memiliki elemen. Berarti, saat queue dibuat, belum ada elemen yang dimasukkan, sehingga tidak ada indeks yang valid untuk elemen pertama dan terakhir dalam queue. Sedangkan nilai awal atribut size diatur menjadi 0 karena untuk melacak jumlah elemen yang saat ini ada dalam queue. Pada awalnya, saat queue baru dibuat, tidak ada elemen yang dimasukkan, sehingga ukuran queue adalah 0.

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

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

Jawab : Potongan kode tersebut bertanggung jawab untuk menangani kasus di mana rear telah mencapai indeks maksimum dari array yang digunakan untuk menyimpan elemen dalam queue. Potongan kode tersebut memastikan bahwa queue dapat terus berputar dan tidak terbatas oleh batas indeks maksimum array yang digunakan untuk menyimpan elemen-elemen queue.

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

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

Jawab : Potongan kode tersebut untuk menangani front telah mencapai indeks maksimum dari array yang digunakan untuk menyimpan elemen dalam queue.

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

Jawab : Karena queue yang diimplementasikan menggunakan konsep circular queue. Dalam circular queue, elemen-elemen queue mungkin tidak selalu berurutan dari indeks 0 hingga max - 1. Jadi, front sebagai awal perulangan memungkinkan kita untuk mencetak elemen-elemen queue dengan benar dan memperlakukan queue seperti circular queue yang diharapkan.

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

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

Jawab : Potongan kode tersebut digunakan untuk menggeser variabel i ke indeks berikutnya dalam array yang digunakan untuk menyimpan elemen-elemen queue. Dengan demikian, potongan kode tersebut memastikan bahwa perulangan mencetak elemen-elemen queue akan berlanjut secara benar dari indeks pertama hingga terakhir.

6. Tunjukkan potongan kode program yang merupakan queue overflow!

Jawab :

```
case 1:  
System.out.println("Masukkan data baru : ");  
int dataMasuk = sc.nextInt();
```



NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : Queue (Jobsheet 8)

```
Q.Enqueue(dataMasuk);  
break;
```

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 :

```
src > PS > Praktikum1 > Queue.java > Queue > Dequeue  
3 public class Queue {  
64  
65     public void Enqueue(int dt) {  
66         if (!IsFull()) {  
67             if (IsEmpty()) {  
68                 front = rear = 0;  
69             } else {  
70                 if (rear == max - 1) {  
71                     rear = 0;  
72                 } else {  
73                     rear++;  
74                 }  
75             }  
76             data[rear] = dt;  
77             size++;  
78         } else {  
79             System.out.println(x:"Queue sudah penuh. Program dihentikan.");  
80             System.exit(status:0);  
81         }  
82     }  
83  
84     public int Dequeue() {  
85         int dt = 0;  
86         if (!IsEmpty()) {  
87             dt = data[front];  
88             size--;  
89             if (size == 0) {  
90                 front = rear = -1;  
91             } else {  
92                 if (front == max - 1) {  
93                     front = 0;  
94                 } else {  
95                     front++;  
96                 }  
97             }  
98         } else {  
99             System.out.println(x:"Queue masih kosong. Program dihentikan.");  
100             System.exit(status:0);  
101         }  
102         return dt;  
103     }  
104 }  
105  
106  
  
System.out.println(x:"Masukkan kapasitas queue : ");  
int n = sc.nextInt();  
  
Queue Q = new Queue(n);  
  
do {  
    menu();  
    pilih = sc.nextInt();  
    switch (pilih) {  
        case 1:  
            if (!Q.IsFull()) {  
                System.out.println(x:"Masukkan data baru : ");  
                int dataMasuk = sc.nextInt();  
                Q.Enqueue(dataMasuk);  
            } else {  
                System.out.println(x:"Queue sudah penuh. Tidak dapat menambahkan elemen baru.");  
            }  
            break;  
        case 2:  
            int dataKeluar = Q.Dequeue();  
            System.out.println(x:"Data yang dikeluarkan adalah : " + dataKeluar);  
            break;  
    }  
}
```



NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : Queue (Jobsheet 8)

2. Praktikum 2

```
src > P9 > Praktikum2 > QueueMain
1 package P9.Praktikum2;
2
3 public class Nasabah {
4     String norek, nama, alamat;
5     int umur;
6     double saldo;
7
8     Nasabah (String norek, String nama, String alamat, int umur, double saldo) {
9         this.norek = norek;
10        this.nama = nama;
11        this.alamat = alamat;
12        this.umur = umur;
13        this.saldo = saldo;
14    }
15
16 }
```

```
J Queue.java - Prak... QueueMain
P9 > Praktikum2 > Queue.java > Queue > peek()
1 package P9.Praktikum2;
2
3 public class Queue {
4     Nasabah[] data;
5     int front;
6     int rear;
7     int size;
8     int max;
9
10    public Queue(int n) {
11        max = n;
12        data = new Nasabah[max];
13        size = 0;
14        front = rear = -1;
15    }
16
17    public boolean IsEmpty() {
18        if (size == 0) {
19            return true;
20        } else {
21            return false;
22        }
23    }
24
25    public boolean IsFull() {
26        if (size == max) {
27            return true;
28        } else {
29            return false;
30        }
31    }
32
33    public void Enqueue(Nasabah dt) {
34        if (!IsFull()) {
35            if (IsEmpty()) {
36                front = rear = 0;
37            } else {
38                if (rear == max - 1) {
39                    rear = 0;
40                } else {
41                    rear++;
42                }
43            }
44            data[rear] = dt;
45            size++;
46        } else {
47            System.out.println("Queue sudah penuh. Program dihentikan.");
48            System.exit(status:0);
49        }
50    }
51 }
```



NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : Queue (Jobsheet 8)

```
1 // PB - Praktikum 2 - 1 Queue.java - 1 Queue - 1 Queue (Jobsheet 8)
2
3 public class Queue {
4
5     public void enqueue() {
6         Marabahaya dt = null;
7         if (!isEmpty()) {
8             dt = data[front];
9             rear--;
10             if (rear == 0) {
11                 front = rear = -1;
12             } else {
13                 if (front == max - 1) {
14                     front = 0;
15                 } else {
16                     front++;
17                 }
18             }
19         } else {
20             System.out.println("Queue masih kosong. Program dihentikan.");
21             System.exit(0);
22         }
23         return dt;
24     }
25
26     public void print() {
27         if (!isEmpty()) {
28             System.out.println("Queue masih kosong.");
29         } else {
30             int i = front;
31             while (i != rear) {
32                 System.out.println(data[i].nama + " " + data[i].alamat + " " + data[i].umur + " " + data[i].saldo);
33                 i = (i + 1) % max;
34             }
35             System.out.println(data[i].nama + " " + data[i].alamat + " " + data[i].umur + " " + data[i].saldo);
36             System.out.println("Jumlah elemen = " + size);
37         }
38     }
39
40     public void peek() {
41         if (!isEmpty()) {
42             System.out.println("Elemen terdapat : " + data[front]);
43         } else {
44             System.out.println("Queue masih kosong.");
45         }
46     }
47 }
48
49 }
```

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

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



NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : Queue (Jobsheet 8)

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);  
}
```

Jawab : Kode program di atas digunakan untuk memeriksa apakah data antrian yang keluar memiliki nilai yang valid sebelum mencetaknya. Keseluruhan `if` statement tersebut akan mengevaluasi apakah semua atribut dari objek `data` memiliki nilai yang valid. Jika semua syarat terpenuhi, maka pernyataan `println` akan dijalankan untuk mencetak data antrian yang keluar.

Sebagai berikut :

1. `!"".equals(data.norek)` memeriksa apakah atribut `norek` dari objek `data` tidak kosong. Jika `norek` tidak kosong, maka kondisi ini bernilai `true`.
 2. `!"".equals(data.nama)` memeriksa apakah atribut `nama` dari objek `data` tidak kosong.
 3. `!"".equals(data.alamat)` memeriksa apakah atribut `alamat` dari objek `data` tidak kosong.
 4. `data.umur != 0` apakah atribut `umur` dari objek `data` tidak sama dengan 0. Sebuah nilai umur yang valid dianggap bukan 0.
 5. `data.saldo != 0` memeriksa apakah atribut `saldo` dari objek `data` tidak sama dengan 0.
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("Antrian paling belakang : " + data[rear].norek + " " + data[rear].nama + " " + data[rear].alamat + " " + data[rear].umur + " " + data[rear].saldo);  
    } else {  
        System.out.println("Queue masih kosong.");  
    }  
}
```



NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : Queue (Jobsheet 8)

```
1 package P3.Praktikum3;
2
3 import java.util.Scanner;
4
5 public class QueueMain {
6     public static void menu() {
7         System.out.println("Pilih menu : ");
8         System.out.println("1. Antrian Baru.");
9         System.out.println("2. Antrian Keluar.");
10        System.out.println("3. Cek Antrian Terdepan.");
11        System.out.println("4. Cek Semua Antrian.");
12        System.out.println("5. Cek Antrian Paling Belakang.");
13        System.out.println(".....");
14    }
15
16    Run [Debug]
17    public static void main(String[] args) {
18        Scanner sc = new Scanner(System.in);
19
20        System.out.println("Masukkan kapasitas queue : ");
21        int jumlah = sc.nextInt();
22        Queue antri = new Queue(jumlah);
23
24        int pilih;
25        do {
26            menu();
27            pilih = sc.nextInt();
28            sc.nextLine();
29            switch (pilih) {
30                case 1:
31                    System.out.println("No Rekening : ");
32                    String norek = sc.nextLine();
33                    System.out.println("Nama : ");
34                    String nama = sc.nextLine();
35                    System.out.println("Alamat : ");
36                    String alamat = sc.nextLine();
37                    System.out.println("Umur : ");
38                    int umur = sc.nextInt();
39                    System.out.println("Saldo : ");
40                    double saldo = sc.nextDouble();
41                    Nasabah nb = new Nasabah(norek, nama, alamat, umur, saldo);
42                    antri.enqueue(nb);
43                    break;
44                case 2:
45                    Nasabah data = antri.dequeue();
46                    if (!"".equals(data.norek) && !"".equals(data.nama) && !"".equals(data.alamat) && data.umur != 0 && data.saldo != 0) {
47                        System.out.println("Antrian yang keluar : " + data.norek + " " + data.nama + " " + data.alamat + " " + data.umur + " " + data.saldo);
48                    }
49                    break;
50                case 3:
51                    antri.peek();
52                    break;
53                case 4:
54                    antri.print();
55                    break;
56                case 5:
57                    antri.peekRear();
58                    break;
59            }
60        } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4);
61    }
62 }
```

```
2. Antrian Keluar.
3. Cek Antrian Terdepan.
4. Cek Semua Antrian
5. Cek Antrian Paling Belakang
-----
1
No Rekening :
123123
Nama :
Kiak
Alamat :
Candi
Umur :
34
Saldo :
23000000
Pilih menu :
1. Antrian Baru.
2. Antrian Keluar.
3. Cek Antrian Terdepan.
4. Cek Semua Antrian
5. Cek Antrian Paling Belakang
-----
1
No Rekening :
321321
Nama :
Soultan
Alamat :
Kalimantan
Umur :
55
Saldo :
39000000
Pilih menu :
1. Antrian Baru.
2. Antrian Keluar.
3. Cek Antrian Terdepan.
4. Cek Semua Antrian
5. Cek Antrian Paling Belakang
-----
3
Antrian paling depan : 123123 Kiak Candi 34 2.3E7
Pilih menu :
1. Antrian Baru.
2. Antrian Keluar.
3. Cek Antrian Terdepan.
4. Cek Semua Antrian
5. Cek Antrian Paling Belakang
-----
5
Antrian paling belakang : 321321 Soultan Kalimantan 55 39000000
PS D:\College\connection-2\VA\coding\dan-Schedule\Date-16-01\
```