



NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET VIII - QUEUE

PRAKTIKUM 8 - QUEUE

Praktikum 1

Waktu percobaan : 45 menit

```
1 package Jobsheet_8.Praktikum1;
2
3 public class Queue07 {
4     int[] data;
5     int front;
6     int rear;
7     int size;
8     int max;
9
10    public Queue07(int n) {
11        max = n;
12        data = new int[max];
13        size = 0;
14        front = rear = -1;
15    }
16    public boolean IsEmpty() {
17        if (size == 0) {
18            return true;
19        } else {
20            return false;
21        }
22    }
23    public boolean IsFull() {
24        if (size == max) {
25            return true;
26        } else {
27            return false;
28        }
29    }
30    public void peek() {
31        if (!IsEmpty()) {
32            System.out.println("Elemen terdepan : " + data[front]);
33        } else {
34            System.out.println("Queue masih kosong");
35        }
36    }
37    public void print() {
38        if (IsEmpty()) {
39            System.out.println("Queue masih kosong");
40        } else {
41            int i = front;
42            while (i != rear) {
43                System.out.println(data[i] + " ");
44                i = (i + 1) % max;
45            }
46            System.out.println(data[i] + " ");
47            System.out.println("Jumlah elemen = " + size);
48        }
49    }
50 }
```

```
1 public void clear() {
2     if (!IsEmpty()) {
3         front = rear = -1;
4         size = 0;
5         System.out.println("Queue berhasil dikosongkan");
6     } else {
7         System.out.println("Queue masih kosong");
8     }
9 }
10 public void Enqueue(int dt) {
11     if (IsFull()) {
12         System.out.println("Queue sudah penuh");
13     } else {
14         if (IsEmpty()) {
15             front = rear = 0;
16         } else {
17             if (rear == max - 1) {
18                 rear = 0;
19             } else {
20                 rear++;
21             }
22         }
23         data[rear] = dt;
24         size++;
25     }
26 }
27 public int Dequeue() {
28     int dt = 0;
29     if (IsEmpty()) {
30         System.out.println("Queue masih kosong");
31     } else {
32         dt = data[front];
33         size--;
34         if (IsEmpty()) {
35             front = rear = -1;
36         } else {
37             if (front == max - 1) {
38                 front = 0;
39             } else {
40                 front++;
41             }
42         }
43     }
44     return dt;
45 }
46 }
47 }
```

```
1 package Jobsheet_8.Praktikum1;
2 import java.util.Scanner;
3
4 public class QueueMain07 {
5     public static void menu() {
6         System.out.println();
7         System.out.println("Masukkan operasi yang diinginkan : ");
8         System.out.println("1. Enqueue");
9         System.out.println("2. Dequeue");
10        System.out.println("3. Print");
11        System.out.println("4. Peek");
12        System.out.println("5. Clear");
13        System.out.println("-----");
14    }
15 }
```



NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET VIII - QUEUE

```
15 public static void main(String[] args) {  
16     Scanner ayleen07 = new Scanner(System.in);  
17     int pilih;  
18  
19     System.out.print("Masukkan kapasitas queue : ");  
20     int n = ayleen07.nextInt();  
21     Queue07 Q = new Queue07(n);  
22  
23     do {  
24         menu();  
25         pilih = ayleen07.nextInt();  
26         switch (pilih) {  
27             case 1:  
28                 System.out.print("Masukkan data baru : ");  
29                 int dataMasuk = ayleen07.nextInt();  
30                 Q.Enqueue(dataMasuk);  
31                 break;  
32             case 2:  
33                 int dataKeluar = Q.Dequeue();  
34                 if (dataKeluar != 0) {  
35                     System.out.println("Data yang di keluarkan : " + dataKeluar);  
36                     break;  
37                 }  
38             case 3:  
39                 Q.print();  
40                 break;  
41             case 4:  
42                 Q.peek();  
43                 break;  
44             case 5:  
45                 Q.clear();  
46                 break;  
47         }  
48     } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5);  
49 }  
50 }  
51 }
```

Hasil run program

```
7/  
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
```



NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET VIII - QUEUE

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 bernilai - 1 karena tidak menunjuk ke data manapun. Saat equeue pertama kali dibuat, maka belum ada elemen dalam array queue sehingga tidak bisa menentukan elemen pertama dan terakhir. Front dan rear digunakan untuk pointer urutan elemen pertama dan terakhir dimana elemen pertama tidak selalu pada indeks 0
- Atribut size bernilai 0 karena queue masih kosong. Atribut size digunakan untuk menyimpan jumlah elemen yang saat ini ada dalam queue. Saat queue pertama kali dibuat, maka masih belum ada elemen array pada queue, sehingga size queue masih bernilai 0.

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

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

Jawab : Potongan kode tersebut dalam method Enqueue digunakan untuk penambahan elemen ke dalam queue ketika rear sudah mencapai batas maksimum (indeks terakhir queue). Saat melakukan penambahan elemen dan pointer rear berada pada indeks paling terakhir dalam array queue maka pointer rear akan ditempatkan ke indeks pertama yaitu 0 dan memasukkan elemen baru pada indeks 0. Sehingga jika rear sudah berada di batas maksimum, maka rear akan diatur kembali ke posisi indeks 0, sehingga elemen baru disisipkan di awal array queue.

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

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

Jawab : Potongan kode tersebut dalam method Dequeue digunakan untuk menangani ketika data front berada pada indeks akhir array (max - 1), maka posisi front akan digeser ke indeks pertama yaitu 0 sehingga pengambilan elemen dilakukan pada array queue indeks terakhir.

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

Jawab : Karena proses perulangan yang dilakukan pada method print digunakan untuk menampilkan data sesuai elemen pertama dan terakhir. Dimana posisi elemen pertama(front)



NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET VIII - QUEUE

tidak harus dimulai dari indeks 0 dan elemen terakhir tidak selalu pada indeks max-1. Array queue bisa saja memiliki elemen yang tidak dimulai dari indeks 0. Sehingga $i = \text{front}$ digunakan untuk memastikan posisi front sehingga proses dimulai dari indeks yang sesuai dengan elemen pertama.

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

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

Jawab : Potongan kode program tersebut pada method print berfungsi untuk menggeser indeks i . Jika i sudah mencapai indeks akhir array ($\text{max} - 1$), maka dengan operasi modulus $\% \text{max}$, nilai i akan kembali ke indeks 0 agar indeks dapat berputar kembali ke awal array ketika mencapai akhir.

6. Tunjukkan potongan kode program yang merupakan queue overflow!

Jawab : Potongan kode queue overflow ditunjukkan pada kode berikut :

```
1 public void Enqueue(int dt) {  
2     if (IsFull()) {  
3         System.out.println("Queue sudah penuh");  
4     } 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); // Pertanyaan no 7  
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 }
```

```
19 public int Dequeue() {  
20     int dt = 0;  
21     if (IsEmpty()) {  
22         System.out.println("Queue masih kosong");  
23         System.exit(0); // Pertanyaan no 7  
24     } else {  
25         dt = data[front];  
26         size--;  
27         if (IsEmpty()) {  
28             front = rear = -1;  
29         } else {  
30             if (front == max - 1) {  
31                 front = 0;  
32             } else {  
33                 front++;  
34             }  
35         }  
36     }  
37     return dt;  
38 }
```



NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET VIII - QUEUE

Praktikum 2

Waktu percobaan : 45 menit

```
1 package Jobsheet_8.Praktikum2;
2
3 public class Queue07 {
4     Nasabah07[] data;
5     int front;
6     int rear;
7     int size;
8     int max;
9
10    public Queue07(int n) {
11        max = n;
12        data = new Nasabah07[max];
13        size = 0;
14        front = rear = -1;
15    }
16    public boolean isEmpty() {
17        if (size == 0) {
18            return true;
19        } else {
20            return false;
21        }
22    }
23    public boolean IsFull() {
24        if (size == max) {
25            return true;
26        } else {
27            return false;
28        }
29    }
30    public void peek() {
31        if (!isEmpty()) {
32            System.out.println("Elemen terdepan : " + data[front].norek + " " +
33                data[front].nama + " " + data[front].alamat + " " + data[front].umur
34                + " " + data[front].saldo);
35        } else {
36            System.out.println("Queue masih kosong");
37        }
38    }
39    public void print() {
40        if (isEmpty()) {
41            System.out.println("Queue masih kosong");
42        } else {
43            int i = front;
44            while (i != rear) {
45                System.out.println(data[i].norek + " " + data[i].nama + " " +
46                    data[i].alamat + " " + data[i].umur + " " + data[i].saldo);
47                i = (i + 1) % max;
48            }
49            System.out.println(data[i].norek + " " + data[i].nama + " " +
50                data[i].alamat + " " + data[i].umur + " " + data[i].saldo);
51        }
52    }
53 }
```

```
1     System.out.println("Jumlah elemen = " + size);
2 }
3
4 public void clear() {
5     if (!isEmpty()) {
6         front = rear = -1;
7         size = 0;
8         System.out.println("Queue berhasil dikosongkan");
9     } else {
10        System.out.println("Queue masih kosong");
11    }
12 }
13 public void Enqueue(Nasabah07 dt) {
14     if (IsFull()) {
15         System.out.println("Queue sudah penuh");
16     } else {
17         if (isEmpty()) {
18             front = rear = 0;
19         } else {
20             if (rear == max - 1) {
21                 rear = 0;
22             } else {
23                 rear++;
24             }
25             data[rear] = dt;
26             size++;
27         }
28     }
29 }
30 public Nasabah07 Dequeue() {
31     Nasabah07 dt = new Nasabah07();
32     if (isEmpty()) {
33         System.out.println("Queue masih kosong");
34     } else {
35         dt = data[front];
36         size--;
37         if (isEmpty()) {
38             front = rear = -1;
39         } else {
40             if (front == max - 1) {
41                 front = 0;
42             } else {
43                 front++;
44             }
45         }
46     }
47     return dt;
48 }
49 }
50 }
```

```
1 package Jobsheet_8.Praktikum2;
2
3 public class Nasabah07 {
4     public String norek, nama, alamat;
5     public int umur;
6     public double saldo;
7
8     Nasabah07() {
9
10    }
11    public Nasabah07(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 }
```



NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET VIII - QUEUE

```
1 package Jobsheet_8.Praktikum2;
2 import java.util.Scanner;
3
4 public class QueueMain07 {
5     public static void menu() {
6         System.out.println();
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("-----");
13    }
14    public static void main(String[] args) {
15        Scanner ayleen07 = new Scanner(System.in);
16        int pilih;
17
18        System.out.print("Masukkan kapasitas queue : ");
19        int jumlah = ayleen07.nextInt();
20        Queue07 antri = new Queue07(jumlah);
21
22        do {
23            menu();
24            pilih = ayleen07.nextInt();
25            ayleen07.nextLine();
26            switch (pilih) {
27                case 1:
28                    System.out.print("No Rekening : ");
29                    String norek = ayleen07.nextLine();
30                    System.out.print("Nama : ");
31                    String nama = ayleen07.nextLine();
32                    System.out.print("Alamat : ");
33                    String alamat = ayleen07.nextLine();
34                    System.out.print("Umur : ");
35                    int umur = ayleen07.nextInt();
36                    System.out.print("Saldo : ");
37                    double saldo = ayleen07.nextDouble();
38                    Nasabah07 nb = new Nasabah07(norek, nama, alamat, umur, saldo);
39                    ayleen07.nextLine();
40                    antri.Enqueue(nb);
41                    break;
42                case 2:
43                    Nasabah07 data = antri.Dequeue();
44                    if (!"".equals(data.norek) && !"".equals(data.nama) && !"".equals(
45                        (data.alamat) && data.umur != 0 && data.saldo != 0) {
46                        System.out.println("Antrian yang keluar : " + data.norek + " "
47                            + data.nama + " " + data.alamat + " " + data.umur + " " + data.saldo);
48                        break;
49                    }
50                case 3:
51                    antri.peek();
52                    break;
53                case 4:
54                    antri.print();
55                    break;
56            }
57        } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4);
58    }
59 }
60 }
```



NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET VIII - QUEUE

Hasil run program

```
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
```

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 : Kondisi ini dibuat untuk memastikan bahwa data tersebut memiliki nilai yang valid dan telah diisi lengkap dengan informasi rekening, nama, alamat, umur, dan saldo.

Potongan kode tersebut juga digunakan untuk memastikan bahwa data yang diambil dari operasi Dequeue tidak kosong atau null. Jika salah satu kondisi tidak terpenuhi, program tidak akan mencetak data karena dianggap data tersebut kosong atau null.



NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET VIII - QUEUE

2. Lakukan modifikasi program dengan menambahkan method baru bernama **peekRear** pada classQueue 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 :

Class Queue07.java

```
1 public void peekRear() {  
2     if (!IsEmpty()) {  
3         System.out.println("Elemen terbelakang : " +  
4             data[rear].norek + " " + data[rear].nama + " " +  
5             data[rear].alamat + " " + data[rear].umur + " " +  
6             data[rear].saldo);  
7     } else {  
8         System.out.println("Queue masih kosong");  
9     }  
10 }
```

Class QueueMain07.java

```
1 System.out.println("Masukkan operasi yang diinginkan : ");  
2 System.out.println("1. Enqueue");  
3 System.out.println("2. Dequeue");  
4 System.out.println("3. Print");  
5 System.out.println("4. Peek");  
6 System.out.println("5. Clear");  
7 System.out.println("-----");
```

```
1     case 4:  
2         Q.peek();  
3         break;  
4     case 5:  
5         Q.clear();  
6         break;  
7     }  
8  
9 } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5);
```




NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET VIII - QUEUE

Hasil run program

```
7'  
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  
-----  
4. Cek Semua Antrian  
5. Cek Antrian paling belakang  
-----  
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  
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  
-----  
5  
Elemen terbelakang : 32940 Susan Surabaya 39 4.2E7
```

Link GitHub Pribadi :

https://github.com/ayleenrq/Prak_AlgoritmaStrukturData_07.git