

# ALGORITMA STRUKTUR DATA

## Praktikum – Queue

Lavina 2341760062

### Praktikum 1

Queue.java

```
1  package Praktikum1;
2
3  class Queue {
4      int[] data;
5      int front, rear, size, max;
6
7      public Queue(int n) {
8          max = n;
9          data = new int[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 peek() {
31         if (!isEmpty()) {
32             System.out.println("Elemen terdepan: " + data[front]);
33         } else {
34             System.out.println("Queue masih kosong");
35         }
36     }
37
38     public void print() {
39         if (isEmpty()) {
40             System.out.println("Queue masih kosong");
41         } else {
42             int i = front;
43             while (i != rear) {
44                 System.out.print(data[i] + " ");
45                 i = (i + 1) % max;
46             }
47             System.out.println(data[i] + " ");
48             System.out.println("Jumlah elemen = " + size);
49         }
50     }
51
52     public void clear() {
53         if (!isEmpty()) {
54             front = rear = -1;
55             size = 0;
56             System.out.println("Queue berhasil dikosongkan");
57         } else {
58             System.out.println("Queue masih kosong");
59         }
60     }
61
62     public void Enqueue(int dt) {
63         if (isFull()) {
64             System.out.println("Queue sudah penuh");
65         } else {
66             if (isEmpty()) {
67                 front = rear = 0;
```

```

68         } else {
69             if (rear == max - 1) {
70                 rear = 0;
71             } else {
72                 rear++;
73             }
74         }
75         data[rear] = dt;
76         size++;
77     }
78 }
79
80 public int Dequeue() {
81     int dt = 0;
82     if (isEmpty()) {
83         System.out.println("Queue masih kosong");
84     } else {
85         dt = data[front];
86         size--;
87         if (isEmpty()) {
88             front = rear = -1;
89         } else {
90             if (front == max - 1) {
91                 front = 0;
92             } else {
93                 front++;
94             }
95         }
96     }
97     return dt;
98 }
99 }

```

## QueueMain.java

```

1  package Praktikum1;
2
3  import java.util.Scanner;
4
5  public class QueueMain {
6      public static void menu() {
7          System.out.println(
8              "Masukkan operasi yang diinginkan:\n1. Enqueue\n2. Dequeue\n3. Print\n4. Peek\n5. Clear\n-----");
9      }
10
11     public static void main(String[] args) {
12         Scanner sc = new Scanner(System.in);
13

```

```

14     System.out.print("Masukkan kapasitas queue: ");
15     int n = sc.nextInt();
16     Queue q = new Queue(n);
17
18     int pilih;
19     do {
20         menu();
21         pilih = sc.nextInt();
22         switch (pilih) {
23             case 1:
24                 System.out.print("Masukkan data baru: ");
25                 int dataMasuk = sc.nextInt();
26                 q.Enqueue(dataMasuk);
27                 break;
28             case 2:
29                 int dataKeluar = q.Dequeue();
30                 if (dataKeluar != 0) {
31                     System.out.println("Data yang dikeluarkan: " + dataKeluar);
32                 }
33                 break;
34             case 3:
35                 q.print();
36                 break;
37             case 4:
38                 q.peek();
39                 break;
40             case 5:
41                 q.clear();
42                 break;
43         }
44     } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5);
45 }
46 }
47

```

Output :

```

Masukkan kapasitas queue: 6
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: 23
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
3
15 23
Jumlah elemen = 2

```

```

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
23
Jumlah elemen = 1

```

## Pertanyaan

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

**Jawab :** Atribut size dibuat 0 karena digunakan untuk mengecek jumlah data yang ada didalam queue yang mana jika 0 maka artinya queue kosong, sedangkan pada rear, -1 mewakili index dari data yang ada didalam queue, jika -1 maka queue kosong, jika 0 maka ada 1 data pada index ke-0.

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

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

**Jawab :** Maksudnya jika index rear = max - 1 berarti data terakhir(rear) berada diujung queue, jika demikian index rear diubah ke 0 sehingga data yang baru akan masuk ke index 0.

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

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

**Jawab :** Maksudnya jika index front = max - 1 berarti data front berada diujung queue, jika true maka index front dipindah ke index 0, karena data yang sebelumnya ada difront sudah hilang.

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

**Jawab :** Karena prinsip dari queue adalah FIFO yang mana data mana yang pertama masuk (front) itulah data yang akan keluar(diprint) terlebih dahulu.

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

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

**Jawab :** Maksud dari kode tersebut adalah untuk memberi nilai variabel i dengan i + 1 yang lalu dimodulo dengan nilai max, tujuannya untuk melakukan loop sampai dimana i = rear, berikut contoh perhitungannya :

```
while (i != rear) {
```

```
    // Iterasi 1
```

```
    // 1 != 3
```

```
// Iterasi 2
// 2 != 3
// Iterasi 3
// 3 != 3 -> false, loop berhenti
```

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

```
// Iterasi 1
// i = (1 + 1) % 6
// i = 2 % 6
// i = 2
```

```
// Iterasi 2
// i = (2 + 1) % 6
// i = 3 % 6
// i = 3
```

```
}
```

6. Tunjukkan potongan kode program yang merupakan queue overflow!

**Jawab :** Queue overflow ditunjukkan oleh,

```
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!

```
public void print() {
    if (isEmpty()) {
        System.out.println(x:"Queue masih kosong");
        System.exit(-1);
    }
}
```

```
public void Enqueue(int dt) {
    if (isFull()) {
        System.out.println(x:"Queue sudah penuh");
        System.exit(-1);
    }
}
```

## Praktikum 2

### Queue.java

```
1 package Praktikum2;
2
3 class Queue {
4     Nasabah[] data;
5     int front, rear, size, max;
6
7     public Queue(int n) {
8         max = n;
9         data = new Nasabah[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 peek() {
31        if (!isEmpty()) {
32            System.out.println("Elemen terdepan: " + data[front].norek + " " + data[front].nama + " " + data[front].alamat
33                + " " + data[front].umur + " " + data[front].saldo);
34        } else {
35            System.out.println("Queue masih kosong");
36        }
37    }
38
39    public void print() {
40        if (isEmpty()) {
41            System.out.println("Queue masih kosong");
42            System.exit(-1);
43        } else {
44            int i = front;
45            while (i != rear) {
46                System.out.println(data[i].norek + " " + data[i].nama + " " + data[i].alamat
47                    + " " + data[i].umur + " " + data[i].saldo);
48                i = (i + 1) % max;
49            }
50            System.out.println(data[i].norek + " " + data[i].nama + " " + data[i].alamat
51                + " " + data[i].umur + " " + data[i].saldo);
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(Nasabah dt) {
67        if (isFull()) {
68            System.out.println("Queue sudah penuh");
69            System.exit(-1);
70        } else {
71            if (isEmpty()) {
72                front = rear = 0;
```

```

73     } else {
74         if (rear == max - 1) {
75             rear = 0;
76         } else {
77             rear++;
78         }
79     }
80     data[rear] = dt;
81     size++;
82 }
83 }
84
85 public Nasabah Dequeue() {
86     Nasabah dt = new Nasabah();
87     if (isEmpty()) {
88         System.out.println("Queue masih kosong");
89     } else {
90         dt = data[front];
91         size--;
92         if (isEmpty()) {
93             front = rear = -1;
94         } else {
95             if (front == max - 1) {
96                 front = 0;
97             } else {
98                 front++;
99             }
100         }
101     }
102     return dt;
103 }
104 }

```

## QueueMain.java

```

1  package Praktikum2;
2
3  import java.util.Scanner;
4
5  public class QueueMain {
6      public static void menu() {
7          System.out.println(
8              "Pilih menu:\n1. Antrian baru\n2. Antrian keluar\n3. Cek antrian terdepan\n4. Cek semua antrian\n-----");
9      }
10
11     public static void main(String[] args) {
12         Scanner sc = new Scanner(System.in);
13
14         System.out.print("Masukkan kapasitas queue: ");
15         int jumlah = sc.nextInt();
16         Queue antri = new Queue(jumlah);
17
18         int pilih;
19         do {
20             menu();
21             pilih = sc.nextInt();
22             sc.nextLine();
23
24             switch (pilih) {
25                 case 1:
26                     System.out.print("No Rekening: ");
27                     String norek = sc.nextLine();
28                     System.out.print("Nama: ");
29                     String nama = sc.nextLine();
30                     System.out.print("Alamat: ");
31                     String alamat = sc.nextLine();
32                     System.out.print("Umur: ");
33                     int umur = sc.nextInt();
34                     System.out.print("Saldo: ");
35                     double saldo = sc.nextDouble();
36
37                     Nasabah nb = new Nasabah(norek, nama, alamat, umur, saldo);
38                     sc.nextLine();
39                     antri.Enqueue(nb);
40                     break;
41                 case 2:
42                     Nasabah data = antri.Dequeue();
43                     if (!"".equals(data.norek) && !"".equals(data.nama) && !"".equals(data.alamat) && data.umur != 0
44                         && data.saldo != 0) {
45                         System.out.println("Antrian yang keluar: " + data.norek + " " + data.nama + " " + data.alamat + " "
46                             + 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

```

Output :

```

Masukkan kapasitas queue: 4
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek antrian terdepan
4. Cek semua antrian
-----
1
No Rekening: 1200046675
Nama: Lavina
Alamat: Losawi, Malang
Umur: 19
Saldo: 10000000
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek antrian terdepan
4. Cek semua antrian
-----
1
No Rekening: 1200197833
Nama: Aden
Alamat: Rungkut, Surabaya
Umur: 17
Saldo: 8000000

Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek antrian terdepan
4. Cek semua antrian
-----
4
1200046675 Lavina Losawi, Malang 19 1.0E7
1200197833 Aden Rungkut, Surabaya 17 8000000.0
Jumlah elemen = 2
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek antrian terdepan
4. Cek semua antrian
-----
3
Elemen terdepan: 1200046675 Lavina Losawi, Malang 19 1.0E7
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek antrian terdepan
4. Cek semua antrian
-----
2
Antrian yang keluar: 1200046675 Lavina Losawi, Malang 19 1.0E7

Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek antrian terdepan
4. Cek semua antrian
-----
4
1200197833 Aden Rungkut, Surabaya 17 8000000.0
Jumlah elemen = 1
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek antrian terdepan
4. Cek semua antrian
-----

```

### 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 :** Pertama kode tersebut mengecek apakah data pada queue kosong dengan menggunakan operator `!""` atau tidak sama dengan kosong, jika true maka data nasabah akan ditampilkan.

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!

Method `peekRear()` :

```

public void peekRear() {
    if (!isEmpty()) {
        System.out.println("Antrian belakang: " + data[rear].norek + " " + data[rear].nama + " " + data[rear].alamat
            + " " + data[rear].umur + " " + data[rear].saldo);
    } else {
        System.out.println(x:"Queue masih kosong");
    }
}

```

Output :

```

1
No Rekening: 123
Nama: mima
Alamat: losawi
Umur: 19
Saldo: 1000
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek antrian terdepan
4. Cek semua antrian
5. Cek antrian paling belakang
-----
1
No Rekening: 456
Nama: aden
Alamat: malang
Umur: 17
Saldo: 5000
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek antrian terdepan
4. Cek semua antrian
5. Cek antrian paling belakang
-----
5
Antrian belakang: 456 aden malang 17 5000.0

```

## Tugas

1. Buatlah program antrian untuk mengilustrasikan antrian pasien di sebuah klinik. Ketika seorang pasien akan mengantri, maka dia harus mendaftarkan nama, nomor identitas, jenis kelamin dan umur seperti yang digambarkan pada Class diagram berikut:

Pembeli
nama: String noID: int jenisKelamin: char umur: int
Pasien (nama: String, noID: int, jenisKelamin: char, umur: int)

Class diagram Queue digambarkan sebagai berikut:

Queue
antrian: Pasien[] front: int rear: int size: int max: int
Queue(n: int) isEmpty(): boolean isFull(): boolean enqueue(antri: Pasien): void dequeue(): int print(): void peek(): void peekRear(): void peekPosition(nama: String): void daftarPasien(): void

Keterangan method:

- Method create(), isEmpty(), isFull(), enqueue(), dequeue() dan print(), kegunaannya sama seperti yang telah dibuat pada Praktikum
- Method peek(): digunakan untuk menampilkan data Pasien yang berada di posisi antrian paling depan
- Method peekRear(): digunakan untuk menampilkan data Pasien yang berada di posisi antrian paling belakang
- Method peekPosition(): digunakan untuk menampilkan seorang pasien (berdasarkan nama) posisi antrian ke berapa
- Method daftarPasien(): digunakan untuk menampilkan data seluruh pasien

## Pasien.java

```
1  package Tugas;
2
3  public class Pasien {
4      String nama;
5      int noID, umur;
6      char jenisKelamin;
7
8      Pasien() {
9
10     }
11
12     Pasien(String nama, int noID, char jenisKelamin, int umur) {
13         this.nama = nama;
14         this.noID = noID;
15         this.jenisKelamin = jenisKelamin;
16         this.umur = umur;
17     }
18 }
```

## Queue.java

```
1  package Tugas;
2
3  public class Queue {
4      Pasien[] antrian;
5      int front, rear, size, max;
6
7      public Queue(int n) {
8          max = n;
9          antrian = new Pasien[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 peek() {
31         if (!isEmpty()) {
32             System.out.println(
33                 "Elemen terdepan: " + antrian[front].nama + " " + antrian[front].noID + " " + antrian[front].jenisKelamin
34                 + " " + antrian[front].umur);
35         } else {
36             System.out.println("Queue masih kosong");
37         }
38     }
39 }
```

```

40 public void peekRear() {
41     if (!isEmpty()) {
42         System.out
43             .println("Antrian belakang: " + antrian[rear].nama + " " + antrian[rear].noID + " "
44                 + antrian[rear].jenisKelamin
45                 + " " + antrian[rear].umur);
46     } else {
47         System.out.println("Queue masih kosong");
48     }
49 }
50
51 public void peekPosition(String nama) {
52     int position = -1;
53     for (int i = 0; i < size; i++) {
54         if (nama.equals(antrian[i].nama)) {
55             position = i;
56         }
57     }
58     System.out.println("Pasien: " + nama + " antrian nomor: " + (position + 1));
59 }
60
61 public void print() {
62     if (isEmpty()) {
63         System.out.println("Queue masih kosong");
64         System.exit(-1);
65     } else {
66         int i = front;
67         while (i != rear) {
68             System.out.println(antrian[i].nama + " " + antrian[i].noID + " " + antrian[i].jenisKelamin
69                 + " " + antrian[i].umur);
70             i = (i + 1) % max;
71         }
72         System.out.println(antrian[i].nama + " " + antrian[i].noID + " " + antrian[i].jenisKelamin
73             + " " + antrian[i].umur);
74         System.out.println("Jumlah elemen = " + size);
75     }
76 }
77
78 public void daftarPasien() {
79     for (int i = 0; i < antrian.length; i++) {
80         if (antrian[i] != null) {
81             System.out
82                 .println(antrian[i].nama + " " + antrian[i].noID + " " + antrian[i].jenisKelamin + " " + antrian[i].umur);
83         } else {
84             System.out.println("-");
85         }
86     }
87 }
88
89 public void Enqueue(Pasien dt) {
90     if (isFull()) {
91         System.out.println("Queue sudah penuh");
92         System.exit(-1);
93     } else {
94         if (isEmpty()) {
95             front = rear = 0;
96         } else {
97             if (rear == max - 1) {
98                 rear = 0;
99             } else {
100                 rear++;
101             }
102         }
103         antrian[rear] = dt;
104         size++;
105     }
106 }
107
108 public Pasien Dequeue() {
109     Pasien dt = new Pasien();
110     if (isEmpty()) {
111         System.out.println("Queue masih kosong");
112     } else {
113         dt = antrian[front];
114         size--;
115         if (isEmpty()) {

```

```

116         front = rear = -1;
117     } else {
118         if (front == max - 1) {
119             front = 0;
120         } else {
121             front++;
122         }
123     }
124 }
125 return dt;
126 }
127
128 }
129

```

## QueueMain.java

```

1 package tugas;
2
3 import java.util.Scanner;
4
5 public class QueueMain {
6     public static void menu() {
7         System.out.println(
8             "Pilih menu:\n1. Antrian baru\n2. Antrian keluar\n3. Cek antrian terdepan\n4. Cek semua antrian\n5. Cek antrian paling belakang\n6. Cek nomor antrian\n7. Daftar pasien\n-----");
9     }
10
11     public static void main(String[] args) {
12         Scanner sc = new Scanner(System.in);
13
14         System.out.print("Masukkan kapasitas antrian: ");
15         int jumlah = sc.nextInt();
16         Queue antri = new Queue(jumlah);
17
18         int pilih;
19         do {
20             menu();
21             pilih = sc.nextInt();
22             sc.nextLine();
23
24             switch (pilih) {
25                 case 1:
26                     System.out.print("No ID: ");
27                     int noID = sc.nextInt();
28                     sc.nextLine();
29                     System.out.print("Nama: ");
30                     String nama = sc.nextLine();
31                     System.out.print("Jenis kelamin: ");
32                     char jenisKelamin = sc.next().charAt(0);
33                     System.out.print("Umur: ");
34                     int umur = sc.nextInt();
35
36                     Pasien p = new Pasien(nama, noID, jenisKelamin, umur);
37                     sc.nextLine();
38                     antri.Enqueue(p);
39                     break;
40                 case 2:
41                     Pasien data = antri.Dequeue();
42                     if (data.noID != 0 && !"".equals(data.nama) && data.jenisKelamin != ' ' && data.umur != 0) {
43                         System.out.println("Antrian yang keluar: " + data.noID + " " + data.nama + " " + data.jenisKelamin + " "
44                             + data.umur);
45                     }
46                     break;
47                 case 3:
48                     antri.peek();
49                     break;
50                 case 4:
51                     antri.print();
52                     break;
53                 case 5:
54                     antri.peekRear();
55                     break;
56                 case 6:
57                     System.out.print("Masukkan nama pasien yang dicari: ");
58                     String nm = sc.nextLine();
59                     antri.peekPosition(nm);
60                     break;
61                 case 7:
62                     antri.daftarPasien();
63                     break;
64             }
65         } while (pilih < 8);
66     }
67 }
68

```

Output :

```
Masukkan kapasitas antrian: 4
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek antrian terdepan
4. Cek semua antrian
5. Cek antrian paling belakang
6. Cek nomor antrian
7. Daftar pasien
```

```
-----
1
No ID: 123
Nama: lavina
Jenis kelamin: p
Umur: 19
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek antrian terdepan
4. Cek semua antrian
5. Cek antrian paling belakang
6. Cek nomor antrian
7. Daftar pasien
```

```
-----
1
No ID: 456
Nama: aden
Jenis kelamin: l
Umur: 17
```

```
Pilih menu:
1. Antrian baru
Run and Debug (Ctrl+Shift+D)
-----
4. Cek semua antrian
5. Cek antrian paling belakang
6. Cek nomor antrian
7. Daftar pasien
-----
5
Antrian belakang: aden 456 l 17
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek antrian terdepan
4. Cek semua antrian
5. Cek antrian paling belakang
6. Cek nomor antrian
7. Daftar pasien
-----
6
Masukkan nama pasien yang dicari: aden
Pasien: aden antrian nomor: 2
```

```
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek antrian terdepan
4. Cek semua antrian
5. Cek antrian paling belakang
6. Cek nomor antrian
7. Daftar pasien
```

```
-----
3
Elemen terdepan: lavina 123 p 19
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek antrian terdepan
4. Cek semua antrian
5. Cek antrian paling belakang
6. Cek nomor antrian
7. Daftar pasien
-----
4
lavina 123 p 19
aden 456 l 17
Jumlah elemen = 2
```

```
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek antrian terdepan
4. Cek semua antrian
5. Cek antrian paling belakang
6. Cek nomor antrian
7. Daftar pasien
-----
7
lavina 123 p 19
aden 456 l 17
-
-
Pilih menu:
1. Antrian baru
2. Antrian keluar
3. Cek antrian terdepan
4. Cek semua antrian
5. Cek antrian paling belakang
6. Cek nomor antrian
7. Daftar pasien
-----
2
Antrian yang keluar: 123 lavina p 19
```