

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA STRUKTUR DATA  
QUEUE**



**Oleh:**

**FAUZIYYAH ADELIA RAMANDA**

**NIM. 2341760145**

**SIB-1F / 10**

**D-IV SISTEM INFORMASI BISNIS  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG**

## 8.2 Praktikum 1

### 8.2.1 Langkah-langkah Percobaan

Queue10.java

```
JobSheet 8 > Queue10.java > Queue10 > Dequeue()
1  public class Queue10 {
2      int [] data;
3      int front;
4      int rear;
5      int size;
6      int max;
7
8      public Queue10 (int n) {
9          max = n;
10         data = new int [max];
11         size = 0;
12         front = rear = -1;
13     }
14     public boolean isEmpty() {
15         if (size == 0) {
16             return true;
17         }else{
18             return false;
19         }
20     }
21     public boolean IsFull(){
22         if (size == max) {
23             return true;
24         } else {
25             return false;
26         }
27     }
28     public void peek () {
29         if (!isEmpty()) {
30             System.out.println("Elemen terdepan: " + data[front]);
31         } else {
32             System.out.println("Queue masih kosong");
33         }
34     }
35 }
```

```
JobSheet 8 > Queue10.java > Queue10 > Queue10(int)
1  public class Queue10 {
2
35     public void print () {
36         if (isEmpty()) {
37             System.out.println("Queue masih kosong");
38         } else {
39             int i = front;
40             while (i != rear) {
41                 System.out.print(data[i] + " ");
42                 i = (i + 1) % max;
43             }
44             System.out.println(data[i] + " ");
45             System.out.println("Jumlah elemen = " + size);
46         }
47     }
48     public void clear() {
49         if (!isEmpty()) {
50             front = rear = -1;
51             size = 0;
52             System.out.println("Queue berhasil dikosongkan");
53         }else{
54             System.out.println("Queue masih kosong");
55         }
56     }
57     public void Enqueue (int 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             }
70             data[rear] = dt;
71             size++;
72         }
73     }
74 }
```

```

74     public int Dequeue () {
75         int dt = 0;
76         if (IsEmpty()) {
77             System.out.println(x: "Queue masih kosong");
78         } else {
79             dt = data [front];
80             size--;
81             if (IsEmpty()){
82                 front = rear = -1;
83             } else {
84                 if (front == max - 1) {
85                     front = 0;
86                 } else {
87                     front++;
88                 }
89             }
90         }
91         return dt;
92     }
93
94
95 }

```

## QueueMain10.java

```

Jobsheet 8 > QueueMain10.java > QueueMain10 > main(String[])
1  import java.util.Scanner;;
2
3  public class QueueMain10 {
4      public static void menu() {
5          System.out.println(x:"Masukkan operasi yang diinginkan: ");
6          System.out.println(x:"1. Enqueue");
7          System.out.println(x:"2. Dequeue");
8          System.out.println(x:"3. Print");
9          System.out.println(x:"4. Peek");
10         System.out.println(x:"5. Clear");
11         System.out.println(x:"-----");
12     }
13     public static void main(String[] args){
14         Scanner sc = new Scanner(System.in);
15         System.out.print(s:"Masukkan kapasitas queue: ");
16         int n = sc.nextInt();
17
18         Queue10 Q = new Queue10 (n);
19
20         int pilih;
21
22         do{
23             menu();
24             pilih = sc.nextInt();
25             switch(pilih) {
26                 case 1 :
27                     System.out.print(s:"Masukkan data baru: ");
28                     int dataMasuk = sc.nextInt();
29                     Q.Enqueue(dataMasuk);
30                     break;
31                 case 2 :
32                     int dataKeluar = Q.Dequeue();
33                     if(dataKeluar != 0){
34                         System.out.println("Data yang dikeluarkan: " + dataKeluar);
35                         break;
36                     }
37                 case 3:
38                     Q.print();
39                     break;
40                 case 4:
41                     Q.peek();
42                     break;
43                 case 5:
44                     Q.clear();
45                     break;
46             }
47         } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5);
48     }
49
50 }

```

## 8.2.2 Verifikasi Hasil Percobaan

Output :

```
1110
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
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
|
```

### 8.2.3 Pertanyaan

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

Nilai awal -1 untuk front dan rear dan 0 untuk size merupakan praktik umum dalam implementasi antrian melingkar, hal ini membantu dalam menandakan antrian kosong, memudahkan perhitungan indeks, dan meningkatkan efisiensi operasi antrian.

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

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

Potongan kode yang ditunjukkan berfungsi untuk memperbarui nilai rear setelah elemen baru ditambahkan ke antrian.

- Jika elemen baru ditambah ke elemen akhir, nilai rear diubah menjadi 0 untuk memulai Kembali dari elemen pertama
- Jika elemen baru ditambahkan ke indeks lain, nilai rear incremented untuk menunjuk ke elemen yang baru ditambahkan

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

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

- `front == max - 1` : potongan kode ini mengecek apakah posisi front saat ini berada pada indeks terakhir dari array data. Jika ini sudah terpenuhi, artinya elemen yang akan di dequeue adalah elemen terakhir dalam array data
- Jika kondisi tersebut terpenuhi, maka front direset ke 0

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

Variable I dimulai dari front karena disini ingin mencetak elemen-elemen dalam urutan yang benar sesuai dengan struktur queue. Dengan memulai dari front kita mencetak elemen dari front queue sehingga rear queue bisa sesuai dengan urutan mereka dalam array data circular

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

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

Pada method print, variable, I tidak memulai dari 0, melainkan dari front, karena kita ingin mencetak elemen queue yang sesungguhnya, yang mungkin dimulai dari indeks selain 0 dalam array circular

6. Tunjukkan potongan kode program yang merupakan queue overflow!

```
if (isFull()) {  
    System.out.println("Queue sudah penuh");  
} else {  
    if (isEmpty()) {  
        front = rear = 0;  
    } else {  
        if (rear == max - 1) {  
            rear = 0;  
        } else {  
            rear++;  
        }  
    }  
    data[rear] = dt;  
    size++;  
}
```

## 8.3 Praktikum 2

### 8.3.1 Langkah-langkah Percobaan

Nasabah10.java

```
Jobsheet 8 > J Nasabah10.java > Nasabah10 > Nasabah10(String nama, String alamat, int umur, double saldo)
1 public class Nasabah10 {
2     String norek, nama, alamat;
3     int umur;
4     double saldo;
5     Nasabah10 (String norek, String nama, String alamat, int umur, double saldo) {
6         this.norek = norek;
7         this.nama = nama;
8         this.alamat = alamat;
9         this.umur = umur;
10        this.saldo = saldo;
11    }
12    Nasabah10 [] data;
13    int front;
14    int rear;
15    int size;
16    int max;
17
18    public Nasabah10 (int n) {
19        max = n;
20        data = new Nasabah10 [max];
21        size = 0;
22        front = rear = -1;
23    }
24    public boolean IsEmpty() {
25        if (size == 0) {
26            return true;
27        } else {
28            return false;
29        }
30    }
31    public boolean IsFull() {
32        if (size == max) {
33            return true;
34        } else {
35            return false;
36        }
37    }
38 }
```

```
39 public void Enqueue (Nasabah10 dt) {
40     if (IsFull()) {
41         System.out.println("Queue sudah penuh");
42     } else {
43         if (IsEmpty()) {
44             front = rear = 0;
45         } else {
46             if (rear == max - 1) {
47                 rear = 0;
48             } else {
49                 rear++;
50             }
51         }
52         data[rear] = dt;
53         size++;
54     }
55 }
56 Nasabah10(){
57 }
58 public Nasabah10 Dequeue () {
59     Nasabah10 dt = new Nasabah10();
60     if (IsEmpty()) {
61         System.out.println("Queue masih kosong");
62     } else {
63         dt = data [front];
64         size--;
65         if (IsEmpty()){
66             front = rear = -1;
67         } else {
68             if (front == max - 1) {
69                 front = 0;
70             } else {
71                 front++;
72             }
73         }
74     }
75     return dt;
76 }
77 }
```

```

76     public void peek() {
77         if(!isEmpty()){
78             System.out.println("Elemen terdepan: " + data[front].norek + " " + data[front].nama + " " + data[front].alamat + " " + data[front].umur + " " + data[front].saldo);
79         } else {
80             System.out.println("Queue masih kosong");
81         }
82     }
83
84     public void print(){
85         if(!isEmpty()){
86             System.out.println("Queue masih kosong");
87         } else {
88             int i = front;
89             while (i != rear) {
90                 System.out.println(data[i].norek + " " + data[i].nama + " " + data[i].alamat + " " + data[i].umur + " " + data[i].saldo);
91                 i = (i+1) % max;
92             }
93             System.out.println(data[i].norek + " " + data[i].nama + " " + data[i].alamat + " " + data[i].umur + " " + data[i].saldo);
94             System.out.println("Jumlah elemen: " + size);
95         }
96     }
97 }
98

```

NasabahMain10.java

```

1  import java.util.Scanner;
2  public class QueueMain10 {
3      public static void menu() {
4          System.out.println("Pilihan menu ");
5          System.out.println("2. Antrian baru");
6          System.out.println("3. Cek Antrian terdepan");
7          System.out.println("4. Cek Semua Antrian");
8          System.out.println("-----");
9      }
10     public static void main(String[] args){
11         Scanner sc = new Scanner(System.in);
12         System.out.print("Masukkan kapasitas queue: ");
13         int jumlah = sc.nextInt();
14         Nasabah10 antri = new Nasabah10(jumlah);
15
16         int pilih;
17

```

```

18         do{
19             menu();
20             pilih = sc.nextInt();
21             sc.nextLine();
22             switch (pilih) {
23                 case 1:
24                     System.out.print("No Rekening: ");
25                     String norek = sc.nextLine();
26                     System.out.print("Nama: ");
27                     String nama = sc.nextLine();
28                     System.out.print("Alamat: ");
29                     String alamat = sc.nextLine();
30                     System.out.print("Umur: ");
31                     int umur = sc.nextInt();
32                     System.out.print("Saldo: ");
33                     double saldo = sc.nextDouble();
34                     Nasabah10 nb = new Nasabah10(norek, nama, alamat, umur, saldo);
35                     sc.nextLine();
36                     antri.Enqueue(nb);
37                     break;
38

```

```

39         case 2:
40             Masabuhlo data = antri.Dequeue();
41             if(!"".equals(data.norek) && !"".equals(data.nama) && !"".equals(data.alamat) && data.umur != 0 && data.saldo != 0){
42                 System.out.println("Antrian yang keluar: " + data.norek + " " + data.nama + " " + data.alamat + " " + data.umur + " " + data.saldo);
43                 break;
44             }
45             case 3:
46                 antri.peek();
47                 break;
48             case 4:
49                 antri.print();
50                 break;
51         }
52     } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 );
53 }
54 }

```

## 8.3.2 Verifikasi Hasil Percobaan

### Output

```

Masukkan kapasitas queue: 4
Pilihan menu
2. Antrian baru
3. Cek Antrian terdepan
4. Cek Semua Antrian
-----
1
No Rekening: 1200046675
Nama: Arif
Alamat: Sukun, Malang
Umur: 25
Saldo: 1200000
Pilihan menu
2. Antrian baru
3. Cek Antrian terdepan
4. Cek Semua Antrian
-----
1
No Rekening: 1200198733
Nama: Dewi
Alamat: Rungkut, Surabaya
Umur: 30
Saldo: 8600000
Pilihan menu
2. Antrian baru
3. Cek Antrian terdepan
4. Cek Semua Antrian
-----
4
1200046675 Arif Sukun, Malang 25 1200000.0
1200198733 Dewi Rungkut, Surabaya 30 8600000.0
Jumlah elemen = 2
Pilihan menu
2. Antrian baru
3. Cek Antrian terdepan
4. Cek Semua Antrian
-----
2
Antrian yang keluar: 1200046675 Arif Sukun, Malang 25 1200000.0
Pilihan menu
2. Antrian baru
3. Cek Antrian terdepan
4. Cek Semua Antrian
-----
3
Elemen terdepan: 1200198733 Dewi Rungkut, Surabaya 30 8600000.0
Pilihan menu
2. Antrian baru
3. Cek Antrian terdepan
4. Cek Semua Antrian
-----
2
Antrian yang keluar: 1200198733 Dewi Rungkut, Surabaya 30 8600000.0
Pilihan menu
2. Antrian baru
3. Cek Antrian terdepan
4. Cek Semua Antrian
-----

```



### 8.3.3 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;
}
```

Digunakan untuk mengecek apakah data yang di dequeue dari antrian memiliki nilai yang valid sebelum dicetak. Jika semua atribut data seperti nomor rek, nama, alamat, umur, dan saldo tidak kosong, maka data tersebut dianggap valid dan dicetak sebagai antrian yang keluar

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!

### 8.4 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 samaseperti yang telah dibuat pada Praktikum
- Method peek(): digunakan untuk menampilkan data Pasien yang berada di posisi antrian palingdepan
- Method peekRear(): digunakan untuk menampilkan data Pasien yang berada di posisi antrianpaling belakang
- Method peekPosition(): digunakan untuk menampilkan seorang pasien (berdasarkan nama)posisi antrian ke berapa
- Method daftarPasien(): digunakan untuk menampilkan data seluruh pasien

Pasien10.java

```
1  package Tugas14;
2
3  public class Pasien10 {
4      String nama;
5      int noID;
6      char jenisKelamin;
7      int umur;
8
9      public Pasien10 (String nama, int noID, char jenisKelamin, int umur){
10         this.nama = nama;
11         this.noID = noID;
12         this.jenisKelamin = jenisKelamin;
13         this.umur = umur;
14     }
15 }
16
```

## Queue10.java

```

1  package Tugas14;
2
3  public class Queue10 {
4      Pasien10[] antrian;
5      int front;
6      int rear;
7      int size;
8      int max;
9
10     public Queue10(int n) {
11         max = n;
12         antrian = new Pasien10[max];
13         size = 0;
14         front = rear = -1;
15     }
16
17     public boolean isEmpty(){
18         return size == 0;
19     }
20
21     public boolean isFull(){
22         return size == max;
23     }
24
25     public void enqueue(Pasien10 pasien){
26         if(isFull()){
27             System.out.println("Queue sudah penuh");
28         } else {
29             if (isEmpty()){
30                 front = rear = 0;
31             } else {
32                 rear = (rear + 1) % max;
33             }
34             antrian [rear] = pasien;
35             size++;
36         }
37     }
38 }

```

```

39
40     public Pasien10 dequeue(){
41         Pasien10 pasien = null;
42         if(isEmpty()){
43             System.out.println("Queue masih kosong");
44         } else {
45             pasien = antrian[front];
46             size--;
47             if(isEmpty()){
48                 front = rear = -1;
49             } else {
50                 front = (front + 1) % max;
51             }
52         }
53         return pasien;
54     }
55
56     public void print() {
57         if(isEmpty()){
58             System.out.println("Queue masih kosong");
59         } else {
60             for (i = front;
61                  i = (i + 1) % max;
62                  i = (i + 1) % max;
63                  while (i != (rear + 1) % max);
64             )
65                 System.out.println("Nama: " + antrian[i].nama + ", No. ID: " + antrian[i].noID + ", Jenis Kelamin: " + antrian[i].jenisKelamin + ", Umur: " + antrian[i].umur);
66         }
67     }
68 }

```

```

61     public void peek(){
62         if(!isEmpty()){
63             System.out.println("Pasien terdepan: " + antrian[front].nama);
64         } else{
65             System.out.println(x:"Queue masih kosong");
66         }
67     }
68     public void peekRear(){
69         if(!isEmpty()){
70             System.out.println("Pasien paling belakang: " + antrian[rear].nama);
71         } else{
72             System.out.println(x:"Queue masih kosong");
73         }
74     }
75     public void peekPosition(String nama) {
76         if(!isEmpty()){
77             int i = front;
78             int position = 1;
79             do{
80                 if(antrian[i].nama.equals(nama)){
81                     System.out.println("Posisi" + nama + "dalam antrian: " + position);
82                     return;
83                 }
84                 position++;
85                 i = (i+1) % max;
86             } while (i != (rear + 1) % max);
87             System.out.println("Pasien" + nama + "tidak ditemukan dalam antrian");
88         } else {
89             System.out.println(x:"Queue masih kosong");
90         }
91     }

```

```

92     public void daftarPasien(){
93         if(!isEmpty()){
94             System.out.println(x:"Daftar Pasien dalam Antrian:");
95             int i = front;
96             do {
97                 System.out.println("Nama: " + antrian[i].nama + ", No.ID: " + antrian[i].noID + ", Jenis Kelamin: " + antrian[i].jenisKelamin + ", Umur: " + antrian[i].umur);
98                 i = (i + 1) % max;
99             } while (i != (rear + 1) % max);
100         } else {
101             System.out.println(x:"Queue masih kosong");
102         }
103     }
104 }
105

```

## QueueMain10.java

```

1  package Tugas14;
2  import java.util.Scanner;
3  public class QueueMain10 {
4      Run | Debug
5      public static void main(String[] args){
6          Scanner sc = new Scanner(System.in);
7          System.out.print(s:"Masukkan kapasitas queue: ");
8          int jumlah = sc.nextInt();
9          Queue10 antri = new Queue10 (jumlah);
10         int pilih;
11
12         do{
13             menu();
14             pilih = sc.nextInt();
15             sc.nextLine();
16             switch (pilih){
17                 case 1:
18                     System.out.println(x:"Nama: ");
19                     String nama = sc.nextLine();
20                     System.out.print(s:"No.ID: ");
21                     int noID = sc.nextInt();
22                     sc.nextLine();
23                     System.out.print(s:"Jenis kelamin (L/P): ");
24                     char jenisKelamin = sc.nextLine().charAt(index:0);
25                     System.out.print(s:"Umur: ");
26                     int umur = sc.nextInt();
27                     Pasien10 pasien = new Pasien10 (nama, noID, jenisKelamin, umur);
28                     antri.enqueue(pasien);
29                     break;
30
31                 case 2:
32                     Pasien10 keluar = antri.dequeue();
33                     if(keluar != null){
34                         System.out.println("Pasien yang keluar: " + keluar.nama);
35                     }
36                     break;
37                 case 3:
38                     antri.peek();
39                     break;
40                 case 4:
41                     antri.peekRear();
42                     break;
43                 case 5:
44                     System.out.print(s:"Masukkan nama Pasien: ");
45                     String namaCari = sc.nextLine();
46                     antri.peekPosition(namaCari);
47                     break;
48                 case 6:
49                     antri.daftarPasien();
50                     break;
51             }
52         } while (pilih >= 1 && pilih <= 6);
53     }
54
55     public static void menu(){
56         System.out.println(x:"Pilih menu: ");
57         System.out.println(x:"1. Antri Pasien");
58         System.out.println(x:"2. Panggil Pasien");
59         System.out.println(x:"3. Cek Pasien terdepan");
60         System.out.println(x:"4. Cek Pasien paling belakang");
61         System.out.println(x:"5. Cek posisi Pasien berdasarkan nama");
62         System.out.println(x:"6. Daftar Pasien dalam antrian");
63         System.out.println(x:"7. Keluar");
64         System.out.println(x:"-----");
65     }
66 }

```

Output :

```
Sheet 2_000112 (bin) - Page 17/44
Masukkan kapasitas queue: 2
Pilih menu:
1. Antri Pasien
2. Panggil Pasien
3. Cek Pasien terdepan
4. Cek Pasien paling belakang
5. Cek posisi Pasien berdasarkan nama
6. Daftar Pasien dalam antrian
7. Keluar
-----
1
Nama:
Ujii
No.ID: 2
Jenis kelamin (L/P): P
Umur: 19
Pilih menu:
1. Antri Pasien
2. Panggil Pasien
3. Cek Pasien terdepan
4. Cek Pasien paling belakang
5. Cek posisi Pasien berdasarkan nama
6. Daftar Pasien dalam antrian
7. Keluar
-----
1
Nama:
Ujii
No.ID: 10
Jenis kelamin (L/P): P
Umur: 19
Pilih menu:
1. Antri Pasien
2. Panggil Pasien
3. Cek Pasien terdepan
4. Cek Pasien paling belakang
5. Cek posisi Pasien berdasarkan nama
6. Daftar Pasien dalam antrian
7. Keluar
-----
3
```

Pasien terdepan: Ujii

Pilih menu:

1. Antri Pasien
2. Panggil Pasien
3. Cek Pasien terdepan
4. Cek Pasien paling belakang
5. Cek posisi Pasien berdasarkan nama
6. Daftar Pasien dalam antrian
7. Keluar

-----

4

Pasien paling belakang: Ujii

Pilih menu:

1. Antri Pasien
2. Panggil Pasien
3. Cek Pasien terdepan
4. Cek Pasien paling belakang
5. Cek posisi Pasien berdasarkan nama
6. Daftar Pasien dalam antrian
7. Keluar

-----

5

Masukkan nama Pasien: Ujii

PosisiUjiidalam antrian: 1

Pilih menu:

1. Antri Pasien
2. Panggil Pasien
3. Cek Pasien terdepan
4. Cek Pasien paling belakang
5. Cek posisi Pasien berdasarkan nama
6. Daftar Pasien dalam antrian
7. Keluar

-----

6

Daftar Pasien dalam Antrian:

Nama: Ujii, No.ID: 2, Jenis Kelamin: P, Umur: 19

Nama: Ujii, No.ID: 10, Jenis Kelamin: P, Umur: 19

Pilih menu:

1. Antri Pasien
2. Panggil Pasien
3. Cek Pasien terdepan
4. Cek Pasien paling belakang
5. Cek posisi Pasien berdasarkan nama
6. Daftar Pasien dalam antrian
7. Keluar

-----

7