

**PRAKTIKUM ALGORITMA DAN STRUKTUR DATA**  
**JOBSHEET 8**



**NAMA : DIMAS ADI BAYU SAMUDRA**

**KELAS : 1A**

**NO. ABSEN : 08**

**NIM : 2341720169**

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**  
**2024**

# Praktikum

## Percobaan 1

### Kode Queue

```
public class Queue08 {
    int [] data ;
    int front, rear, size, max;

    public Queue08(int n){
        max = n;
        data = new int[max];
        size = 0 ;
        front = rear = -1;
    }

    public boolean IsEmpty(){
        if (size == 0) {
            return true;
        } else {
            return false;
        }
    }

    public boolean isFull(){
        if (size == max) {
            return true;
        } else {
            return false;
        }
    }

    public void peek(){
        if (!IsEmpty()) {
            System.out.println("Elemen terdepan : " + data[front]);
        } else {
            System.out.println("Queue masih kosong");
        }
    }

    public void print(){
        if (!IsEmpty()) {
            System.out.println("Queue masih kosong");
        } else {
            int i = front;
            while (i != rear) {
                System.out.println(data[i] + " ");
                i = (i + 1) % max;
            }
            System.out.println(data[i] + " ");
            System.out.println("jumlah elemen = " + size);
        }
    }
}
```

```

public void clear () {
    if (!IsEmpty()) {
        front = rear = -1;
        size = 0;
        System.out.println("Queue Berhasil dikosongkan");
    } else {
        System.out.println("Queue masih kosong ");
    }
}

public void Enqueue(int dt) {
    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++;
    }
}

public int Dequeue() {
    int dt = 0;
    if (IsEmpty()) {
        System.out.println("Queue masih kosong ");
    } else {
        dt = data[front];
        size--;
        if (IsEmpty()) {
            front = rear - 1;
        } else {
            if (front == max - 1) {
                front = 0;
            } else {
                front++;
            }
        }
    }
    return dt;
}
}

```

Kode Main

```
import java.util.Scanner;
```

```

public class main08 {
    public static void menu(){
        System.out.println("Masukan Operasi yang diiinginkan : ");
        System.out.println("1. Enqueue");
        System.out.println("2. Dequeue");
        System.out.println("3. Print");
        System.out.println("4. Peek");
        System.out.println("5. Clear");
        System.out.println("-----");
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Masukan kapasitas queue : ");
        int n = sc.nextInt();

        Queue08 Q = new Queue08(n);
        int pilih;
        do {
            menu();
            pilih = sc.nextInt();
            switch (pilih) {
                case 1:
                    System.out.print("Masukan Data baru : ");
                    int dataMasuk = sc.nextInt();
                    Q.Enqueue(dataMasuk);
                    break;
                case 2:
                    int dataKeluar = Q.Dequeue();
                    if (dataKeluar != 0) {
                        System.out.println("Data yang dikeluarkan : " +
dataKeluar);
                        break;
                    }
                case 3:
                    Q.print();
                    break;
                case 4:
                    Q.peek();
                    break;
                case 5:
                    Q.clear();
                    break;
            }
        } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 ||
pilih == 5 );

    }
}

```

## Hasil run program queue

```
Masukan kapasitas queue : 4
Masukan Operasi yang diiinginkan :
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
1
Masukan Data baru : 15
Masukan Operasi yang diiinginkan :
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
1
Masukan Data baru : 31
Masukan Operasi yang diiinginkan :
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
4
Elemen terdepan : 15
Masukan Operasi yang diiinginkan :
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
```

## Pertanyaan

1. Pada konstruktor, mengapa nilai awal atribut front dan rear bernilai -1, sementara atribut size bernilai 0?
2. Pada method Enqueue, jelaskan maksud dan kegunaan dari potongan kode berikut!
3. Pada method Dequeue, jelaskan maksud dan kegunaan dari potongan kode berikut!
4. Pada method print, mengapa pada proses perulangan variabel i tidak dimulai dari 0 (int i=0), melainkan int i=front?
5. Perhatikan kembali method print, jelaskan maksud dari potongan kode berikut!
6. Tunjukkan potongan kode program yang merupakan queue overflow!
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!

## Jawaban

1. Dikarenakan Index yang di gunakan di mulai dari 0 sehingga rear di bernilai -1, dan size di beri nilai dari 0 di karenakan size adalah total data yang diinput.
2. Jika nilai rear sama dengan max -1 maka nilai rear menjadi 0.
3. jika nilai front sama dengan max -1 maka nilai front menjadi 0.
4. dikarenakan front tidak selalu pada index 0.
5. nilai i akan di ganti dengan  $i + 1$  modulus max
6. queue overflow

```
public void Enqueue(int dt){
    if (isFull()) {
        System.out.println("Queue sudah kosong");
    } else {
        if (IsEmpty()) {
            front = rear = 0;
        } else {
            if (rear == max - 1) {
                rear = 0;
            } else {
                rear++;
            }
        }
        data[rear] = dt;
        size++;
    }
}
```

7. kode programm yang di tambahkan adalah System.exit(0);

```
public void enqueue(int dt) {
    if (isFull()) {
        System.out.println("Queue overflow! Program dihentikan.");
        System.exit(0);
    } else {
        if (isEmpty()) {
            front = rear = 0;
        } else {
            if (rear == max - 1) {
                rear = 0;
            } else {
                rear++;
            }
        }
        data[rear] = dt;
        size++;
    }
}

public int dequeue() {
```

```

        int dt = 0;
        if (isEmpty()) {
            System.out.println("Queue underflow! Program dihentikan.");
            System.exit(0);
        } else {
            dt = data[front];
            size--;
            if (isEmpty()) {
                front = rear = -1;
            } else {
                if (front == max - 1) {
                    front = 0;
                } else {
                    front++;
                }
            }
        }
        return dt;
    }
}

```

## Percobaan 2

### Kode program Nasabah08

```

package praktikum2;

public class Nasabah08 {
    String norek,nama,alamat;
    int umur;
    double saldo;

    public Nasabah08(String norek, String nama, String alamat, int umur ,
double saldo){
        this.norek = norek;
        this.nama = nama;
        this.alamat = alamat;
        this.umur = umur;
        this.saldo = saldo;
    }
}

```

### Kode program QueueNasabah08

```

package praktikum2;

public class QueueNasabah08 {

    Nasabah08[] data;
    int front,rear,size,max;
    public QueueNasabah08(int n) {
        max = n;
        data = new Nasabah08[max];
        size = 0;
        front = rear = -1;
    }
}

```

```

    }

    public boolean isEmpty() {
        return size == 0;
    }

    public boolean isFull() {
        return size == max;
    }

    public void peek() {
        if (!isEmpty()) {
            System.out.println("Elemen terdepan : " + data[front].norek + "
" + data[front].nama + " " + data[front].alamat + " " + data[front].umur + "
" + data[front].saldo );
        } else {
            System.out.println("Queue masih kosong");
        }
    }

    public void print() {
        if (!isEmpty()) {
            int i = front;
            while (i != rear) {
                System.out.println(data[i].norek + " " + data[i].nama + " "
+ data[i].alamat + " " + data[i].umur + " " + data[i].saldo);
                i = (i + 1) % max;
            }
            System.out.println(data[i].norek + " " + data[i].nama + " " +
data[i].alamat + " " + data[i].umur + " " + data[i].saldo);
            System.out.println("Jumlah elemen = " + size);
        } else {
            System.out.println("Queue masih kosong");
        }
    }

    public void clear() {
        if (!isEmpty()) {
            front = rear = -1;
            size = 0;
            System.out.println("Queue berhasil dikosongkan");
        } else {
            System.out.println("Queue masih kosong");
        }
    }

    public void enqueue(Nasabah08 dt) {
        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++;
}

}

public Nasabah08 dequeue() {
    Nasabah08 dt = new Nasabah08(null, null, null, max, front);
    if (isEmpty()) {
        System.out.println("Queue masih kosong");
    } else {
        dt = data[front];
        size--;
        if (isEmpty()) {
            front = rear = -1;
        } else {
            if (front == max - 1) {
                front = 0;
            } else {
                front++;
            }
        }
    }
    return dt;
}
}

```

Kode program mainNasabah08

```

package praktikum2;

import java.util.Queue;
import java.util.Scanner;

public class QueueMain08 {
    public static void menu(){
        System.out.println("Pilih Menu : ");
        System.out.println("1. Antrian baru");
        System.out.println("2. Antrian keluar");
        System.out.println("3. Cek Antrian terdepan");
        System.out.println("4. Cek Semua Antrian");
        System.out.println("-----");
    }
    public static void main(String[] args) {
        int pilih;
        Scanner sc = new Scanner(System.in);

        System.out.print("Masukan kapasitas Queue : ");
        int jumlah = sc.nextInt();
    }
}

```

```

QueueNasabah08 antri = new QueueNasabah08(jumlah);

do {
    menu();
    pilih = sc.nextInt();
    switch (pilih) {
        case 1:
            System.out.print("No Rekening : ");
            String norek = sc.next();
            System.out.print("Nama : ");
            String nama = sc.next();
            System.out.print("Alamat : ");
            String alamat = sc.next();
            System.out.print("Umur : ");
            int umur = sc.nextInt();
            System.out.print("Saldo : ");
            double saldo = sc.nextDouble();
            Nasabah08 nb = new Nasabah08(norek, nama, alamat, umur,
saldo);

            sc.nextLine();
            antri.enqueue(nb);
            break;
        case 2:
            Nasabah08 data = antri.dequeue();
            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;
            }
        case 3:
            antri.peek();
            break;
        case 4:
            antri.print();
            break;
    }
} while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 );
}

```

Hasil Run

```

Masukan 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
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;
}

```

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!

Jawaban

1. untuk memastikan bahwa data nasabah tidak kosong sehingga bisa mengeluarkan dari antrian
2. Kode tambahan dan modifikasi

Kode queueNasabah

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

Kode mainQueueNasabah

```
System.out.println("5. Cek Antrian paling belakang");
case 5:
    antri.peekRear();
```

Hasil run

```
Masukan kapasitas Queue : 8
Pilih Menu :
1. Antrian baru
2. Antrian keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian paling belakang
-----
1
No Rekening : asd
Nama : asd
Alamat : asda
Umur : 1
Saldo : 12
Pilih Menu :
1. Antrian baru
2. Antrian keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian paling belakang
-----
1
No Rekening : sdf
Nama : sdf
Alamat : sdf
Umur : 2
Saldo : 22
Pilih Menu :
1. Antrian baru
2. Antrian keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian paling belakang
-----
5
Elemen terbelakang : sdf sdf sdf 2 22.0
```

## Tugas Praktikum

### Kode Pembeli

```
package praktikum3;

public class Pembeli08 {
    String nama;
    int noHp;

    public Pembeli08(String nama, int noHp){
        this.nama = nama;
        this.noHp = noHp;
    }
}
```

### Kode Queue08

```
package praktikum3;

public class Queue08 {
    Pembeli08[] antrian;
    int front, rear, size, max;

    public Queue08(int n){
        max = n;
        antrian = new Pembeli08[max];
        size = 0;
        front = rear = -1;
    }

    public boolean isEmpty(){
        return size == 0;
    }

    public boolean isFull() {
        return size == max;
    }

    public void enqueue(Pembeli08 pb) {
        if (isFull()) {
            System.out.println("Queue sudah penuh");
        } else {
            if (isEmpty()) {
                front = rear = 0;
            } else {
                if (rear == max - 1) {
                    rear = 0;
                } else {
                    rear++;
                }
            }
            antrian[rear] = pb;
            size++;
        }
    }
}
```

```

    }
}

public Pembeli08 dequeue() {
    Pembeli08 atr = new Pembeli08(null, front);
    if (isEmpty()) {
        System.out.println("Queue masih kosong");
    } else {
        atr = antrian[front];
        size--;
        if (isEmpty()) {
            front = rear = -1;
        } else {
            if (front == max - 1) {
                front = 0;
            } else {
                front++;
            }
        }
    }
    return atr;
}

public void print() {
    if (!isEmpty()) {
        int i = front;
        while (i != rear) {
            System.out.println(antrian[i].nama + " " + antrian[i].noHp);
            i = (i + 1) % max;
        }
        System.out.println(antrian[i].nama + " " + antrian[i].noHp );
        System.out.println("Jumlah elemen = " + size);
    } else {
        System.out.println("Queue masih kosong");
    }
}

public void peek() {
    if (!isEmpty()) {
        System.out.println("Elemen terdepan : " + antrian[front].nama +
" " + antrian[front].noHp );
    } else {
        System.out.println("Queue masih kosong");
    }
}

public void peekRear() {
    if (!isEmpty()) {
        System.out.println("Elemen terdepan : " + antrian[rear].nama + "
" + antrian[rear].noHp );
    } else {
        System.out.println("Queue masih kosong");
    }
}

```

```

    }

    public void peekPosition(String nama) {
        if (isEmpty()) {
            for (int i = 0; i < antrian.length; i++) {
                if (antrian[i].nama.equals(nama)) {
                    System.out.println("anda urutan " + i);
                }
            }
        }
    }

    public void daftarPembeli(){
        if (!isEmpty()) {
            int i = front;
            while (i != rear) {
                System.out.println(antrian[i].nama + " " + antrian[i].noHp);
                i = (i + 1) % max;
            }
            System.out.println(antrian[i].nama + " " + antrian[i].noHp );
            System.out.println("Jumlah elemen = " + size);
        } else {
            System.out.println("Queue masih kosong");
        }
    }
}

```

#### Kode mainPembeli

```

package praktikum3;

import java.util.Scanner;

public class mainPembeli {
    public static void menu(){
        System.out.println("Pilih Menu : ");
        System.out.println("1. Antrian baru");
        System.out.println("2. Antrian keluar");
        System.out.println("3. Cek Antrian terdepan");
        System.out.println("4. Cek Semua Antrian");
        System.out.println("5. Cek Antrian paling belakang");
        System.out.println("6. cari nama ");
        System.out.println("7. Daftar pembeli");
        System.out.println("-----");
    }

    public static void main(String[] args) {
        int pilih;
        Scanner sc = new Scanner(System.in);

        System.out.print("Masukan kapasitas Queue : ");
        int jumlah = sc.nextInt();
        Queue08 antri = new Queue08(jumlah);

        do {

```

```

menu();
pilih = sc.nextInt();
switch (pilih) {
    case 1:
        System.out.print("Nama : ");
        String nama = sc.next();
        System.out.print("no Hp : ");
        int noHp = sc.nextInt();
        Pembeli08 pb = new Pembeli08(nama,noHp);
        sc.nextLine();
        antri.enqueue(pb);
        break;
    case 2:
        Pembeli08 antrian = antri.dequeue();
        if (!"".equals(antrian.nama) && antrian.noHp != 0 ) {
            System.out.println("Antrian yang keluar : " +
antrian.nama + " " + antrian.noHp );
            break;
        }
    case 3:
        antri.peek();
        break;
    case 4:
        antri.print();
        break;
    case 5:
        antri.peekRear();
        break;
    case 6:
        System.out.print("Masukan nama yang ingin anda cari
antreannya : ");
        String cariNama = sc.next();
        antri.peekPosition(cariNama);
        break;
    case 7:
        antri.daftarPembeli();
}
} while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 ||
pilih == 5 || pilih == 6 || pilih == 7);
}
}

```



Hasil run

```
Masukan kapasitas Queue : 5
Pilih Menu :
1. Antrian baru
2. Antrian keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian paling belakang
6. cari nama
7. Daftar pembeli
-----
1
Nama : samid
no Hp : 12345
Pilih Menu :
1. Antrian baru
2. Antrian keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian paling belakang
6. cari nama
7. Daftar pembeli
-----
1
Nama : dimas
no Hp : 44223
Pilih Menu :
1. Antrian baru
2. Antrian keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian paling belakang
6. cari nama
7. Daftar pembeli
-----
1
Nama : didi
no Hp : 44552
```

```
Pilih Menu :
1. Antrian baru
2. Antrian keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian paling belakang
6. cari nama
7. Daftar pembeli
-----
2
Antrian yang keluar : samid 12345
Pilih Menu :
1. Antrian baru
2. Antrian keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian paling belakang
6. cari nama
7. Daftar pembeli
-----
3
Elemen terdepan : dimas 44223
Pilih Menu :
1. Antrian baru
2. Antrian keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian paling belakang
6. cari nama
7. Daftar pembeli
-----
4
dimas 44223
didi 44552
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. cari nama
7. Daftar pembeli
-----
5
Elemen terdepan : didi 44552
Pilih Menu :
1. Antrian baru
2. Antrian keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian paling belakang
6. cari nama
7. Daftar pembeli
-----
6
Masukan nama yang ingin anda cari antreannya : didi
Pilih Menu :
1. Antrian baru
2. Antrian keluar
3. Cek Antrian terdepan
4. Cek Semua Antrian
5. Cek Antrian paling belakang
6. cari nama
7. Daftar pembeli
-----
7
dimas 44223
didi 44552
Jumlah elemen = 2
```