PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

JOBSHEET PERTEMUAN KE-8

NAMA : ALVINO VALERIAN D.R

KELAS : 1A

NO. ABSEN : 05

NIM : 2341720027

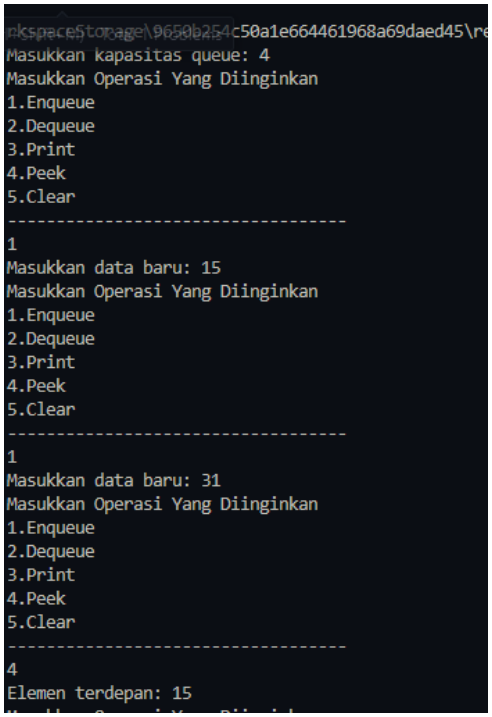
PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2024

10.2.2 Verifikasi Hasil Percobaan 1



10.2.3 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!

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

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

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

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!

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

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. nilai awal atribut front dan rear diset sebagai -1 karena pada awalnya, antrian (queue) kosong. Nilai size diatur sebagai 0 karena belum ada elemen yang dimasukkan ke dalam antrian.

2. rear == max - 1 terpenuhi, artinya rear telah mencapai batas maksimum dan elemen terakhir yang ditambahkan adalah pada indeks max - 1. Dalam penanganan ini, rear diatur kembali ke 0 ,bahwa elemen berikutnya akan dimasukkan di awal array antrian

3. front diatur kembali ke 0, menandakan bahwa elemen berikutnya yang akan diambil adalah dari indeks awal array antrian jika ada elemen di dalamnya.

4. variabel i dari nilai front, hanya mencetak elemen yang valid dalam antrian, yang sesuai dengan urutan masuknya ke dalam antrian.

5. i saat ini berada di batas maksimum array (yaitu max - 1), operasi (i+1) % max akan menghasilkan nilai 0, sehingga kembali ke awal array. Namun, jika i masih berada di dalam rentang yang valid (kurang dari max - 1), operasi modulo akan menghasilkan nilai yang sama dengan i+1, sehingga terus maju ke elemen berikutnya dalam antrian.

6.

```
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++;
    }
```

7.menambahkan `System.exit(0);`

```
public void Enqueue(int dt){
    if(isFull()){
        System.out.println("Queue sudah penuh");
        System.exit(0);
```

10.3.3 Pertanyaan2

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.!"".equals(data.norek): Memeriksa apakah atribut norek dari objek data tidak kosong (tidak merupakan string kosong).!"".equals(data.nama): Memeriksa apakah atribut nama dari objek data tidak kosong.!"".equals(data.alamat): Memeriksa apakah atribut alamat dari objek data tidak kosong.!"".equals(data.umur): Memeriksa apakah atribut umur dari objek data tidak kosong.!"".equals(data.saldo): Memeriksa apakah atribut saldo dari objek data tidak kosong.

2.

```
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");
    }
}
```

```
case 5:
                    antri.peekRear();
                    break;
        }
    }while (pilih==1 || pilih==2 || pilih==3 || pilih==4 ||
pilih==5);
```

```
3.Cek antrian terdepan
4.Cek semua antrian
5.cek antrian belakang
---------------------------------
1
No rekening: 123
Nama: vino
Alamat: malang
Umur: 21
Saldo: 2000000
Pilih Menu:
1.Antrian baru
2.Antrian keluar
3.Cek antrian terdepan
4.Cek semua antrian
5.cek antrian belakang
---------------------------------
1
No rekening: 234
Nama: nabel
Alamat: surabaya
Umur: 32
Saldo: 400000
Pilih Menu:
1.Antrian baru
2.Antrian keluar
3.Cek antrian terdepan
4.Cek semua antrian
5.cek antrian belakang
---------------------------------
5
Antrian paling belakang: 234 nabel surabaya 32 400000.0
```

TUGAS

```
package tugas;


public class Pembeli05 {
        String nama;
        int nomorHP;

        public Pembeli05(){
        }
        Pembeli05 (String nama, int nomorHP){
            this.nama = nama;
            this.nomorHP = nomorHP;

        }
        Pembeli05 [] antrian;
        int front;
        int rear;
        int size;
        int max;

        public Pembeli05 (int n) {
            max = n;
            antrian = new Pembeli05 [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 Enqueue(Pembeli05 dt) {
        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] = dt;
            size++;
        }
    }
    public Pembeli05 Dequeue() {
        Pembeli05 dt = new Pembeli05();
        if (IsEmpty()) {
            System.out.println("Queue Masih Kosong");
        } else {
            dt = antrian[front];
            size--;
```

```
                if (IsEmpty()) {
                    front = rear = -1;
                } else {
                    if (front == max -1){
                        front = 0;
                    } else {
                        front++;
                    }
                }
            }
            return dt;
        }
        public void print() {
            if (IsEmpty()) {
                System.out.println("Queue Masih Kosong");
            } else {
                int i = front;
                while (i != rear) {
                    System.out.println(antrian[i].nama + " " +
antrian[i].nomorHP);
                    i = (i + 1) % max;
                }
                System.out.println(antrian[i].nama + " " +
antrian[i].nomorHP);
                System.out.println("Jumlah Elemen = " + size );
            }
        }
        public void peek() {
            if(!IsEmpty()){
                System.out.println("Pelanggan Terdepan : " +
antrian[front].nama);
            } else {
                System.out.println("Antrian Masih Kosong");
            }
```

```
        }
        public void peekRear() {
            if(!IsEmpty()) {
                System.out.println("Atrian Pelanggan Posisi Belakang
: " + antrian[rear].nama);
            } else {
                System.out.println("Antrian Masih Kosong");
            }
        }
        public void peekPosition(String nama) {
            int posisi = -1;
            for (int i = 0; i < size; i++) {
                if (antrian[i].nama.equalsIgnoreCase(nama)){
                    posisi = i + 1;
                    break;
                }
            }
            if (posisi != -1) {
                System.out.println("Pelanggan " + nama + " Berada Di
Posisi ke-" + posisi);
            } else {
                System.out.println("Pelanggan " + nama + " Tidak
Ditemukan Dalam Antrian");
            }
        }
        public void daftarPelanggan() {
            if (IsEmpty()) {
                System.out.println("Antrian Masih Kosong");
            } else {
                System.out.println("Daftar Pelanggan Dalam Antrian :
");
                int i = front;
                do {
                    System.out.println(antrian[i].nama);
```

```
                    i = (i + 1) % max;
                } while (i != rear);
                System.out.println(antrian[i].nama);
            }
        }
    }
```

```
package tugas;

import java.util.Scanner;

public class Queue05 {
    public static void menu() {
        System.out.println();
        System.out.println("Pilih Menu");
        System.out.println("1. Antrian pelanggan baru");
        System.out.println("2. Antrian pelanggan keluar");
        System.out.println("3. Cek Pelanggan terdepan");
        System.out.println("4. Cek Semua pelanggan");
        System.out.println("5. Cek Antrian pelanggan Di posisi
belakang");
        System.out.println("6. Cek Posisi pelanggan");
        System.out.println("7. Tampilkan daftar pelanggan");

System.out.println("==========================================");
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Masukkan kapasitas Queue05 : ");
        int jumlah = sc.nextInt();
        Pembeli05 antri = new Pembeli05(jumlah);
```

```
        int pilih;


        do {
            menu();
            pilih = sc.nextInt();
            sc.nextLine();
            switch (pilih) {
            case 1:
                System.out.print("Nama      : ");
                String nama = sc.nextLine();
                System.out.print("Nomor HP  : ");
                int nomorHP = sc.nextInt();
                Pembeli05 p = new Pembeli05(nama, nomorHP);
                sc.nextLine();
                antri.Enqueue(p);
                break;
            case 2:
                Pembeli05 data = antri.Dequeue();
                if (!"".equals(data.nama) &&
!"".equals(data.nomorHP)) {
                    System.out.println("Antrian Yang Keluar : " +
data.nama + " " + data.nomorHP);
                    break;
                }
            case 3:
                antri.peek();
                break;
            case 4:
                antri.print();
                break;
            case 5:
                antri.peekRear();
```

```
                break;
            case 6:
                System.out.print("Nama Pelanggan : ");
                String namaPelanggan = sc.nextLine();
                antri.peekPosition(namaPelanggan);
                break;
            case 7:
                antri.daftarPelanggan();
                break;
            }
        } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 
4 || pilih == 5 || pilih == 6 || pilih == 7);
    }
}
```