

Laporan Hasil Praktikum

Algoritma Sistem Dasar

QUEUE

Jobsheet 10



MUHAMMAD PEARL OCSHADA

244107020064

TI 1-E

Program Studi Teknik Informatika

Jurusan Teknologi Informasi

POLINEMA

2024

Jobsheet X

Queue

Percobaan1

- Source Code queue.java

```
public class Queue {  
    int[] data;  
    int front;  
    int rear;  
    int size;  
    int max;  
  
    public Queue(int n){  
        max = n;  
        data = new int[max];  
        size = 0;  
        front = rear = -1;  
    }  
  
    public boolean isFull(){  
        if (size == max) {  
            return true;  
        } else {  
            return false;  
        }  
    }  
  
    public boolean isEmpty(){  
        if (size == 0) {  
            return true;  
        } else {  
            return false;  
        }  
    }  
  
    public void peek(){  
        if (!isEmpty()) {  
            System.out.println("Elemen terdepan: "+data[front]);  
        }  
    }  
}
```

```

public void print(){
    if (!isEmpty()) {
        int i = front;
        while (i!= rear){
            System.out.print(data[i] + " ");
            i = (i+1)%max;
        }
        System.out.println(data[i] + " ");
        System.out.println("Jumlah element = " + 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(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 sudah kosong!");
        } else{
            dt = data[front];
            size--;
            if (isEmpty()) {
                front = rear = -1;
            } else {
                if (front == max -1) {
                    front = 0;
                } else {
                    front++;
                }
            }
        }
        return dt;
    }
}

```

- Source Code queueMain.java

```

import java.util.Scanner;

public class queueMain {
    public static void menu() {
        System.out.println("Masukkan Operasi yang diinginkan: ");
        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("Masukkan kapasitas Queue: ");
    int n = sc.nextInt();
    Queue Q = new Queue(n);
    int pilih;
    do {
        menu();
        System.out.print("Pilih : ");
        pilih = sc.nextInt();
        switch (pilih) {
            case 1:
                System.out.print("Masukkan 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;
            default:
                System.out.println("Pilih menu dengan benar");
                break;
        }
    } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5);
}

```

- Pertanyaan

1. Pada konstruktor nilai awal atribut front and rear bernilai -1 dikarenakan pada Queue bernilai kosong yang memiliki indexs -1 untuk nilai sementara atau menandai bahwa Queue masi kosong, dan jika nanti Queue memiliki data maka front and rear akan berpindah. Kenapa Size bernilai 0, dikarenakan pada Array Queue belum memiliki element data yang tersimpan saat pertama kali dibuat.
2. Kode tersebut digunakan untuk mengatur agar index rear Kembali ke indeks 0 jika sudah mencapai akhir array Queue (max -1).
3. Kode tersebut digunakan untuk mengatur agar index front Kembali ke indeks 0 jika data sudah mencapai akhir array Queue(max-1), maka saat element tersebut dihapus maka front Kembali ke array index 0 agar bisa membaca data diposisi awal jika ada.
4. Dikarenakan element pertama belum pasti dari indeks 0. Pada Queue element pertama yang valid berada pada posisi front, jadi perulangan untuk mencetak isi Queue harus dimulai dari front, bukan dari 0.
5. Kode ini digunakan untuk mengatur indeks i agar melingkar ke awal array jika sudah mencapai akhir (max - 1). Dengan % max, nilai indeks tidak akan melebihi panjang array dan tetap dalam rentang valid.
6. Kode yang menunjukkan Queue overflow pada saat if kondisi (isFull), dikarenakan pada code tersebut mencetak jika Queue sudah penuh.

```
if (isFull()) {  
    System.out.println("Queue sudah penuh!");  
    System.exit(0);  
}
```

7. Memodifikasi code program dengan menambahkan system.exit(0) untuk menghentikan program Queue underflow atau overflow.

```

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

```

Percobaan2

- Source code mahasiswa20.java

```
package p2Jobsheet10;

public class mahasiswa20 {
    String nim;
    String nama;
    String prodi;
    String kelas;

    public mahasiswa20(String nim, String nama, String prodi, String kelas){
        this.nim = nim;
        this.nama = nama;
        this.prodi = prodi;
        this.kelas = kelas;
    }

    public void tampilData(){
        System.out.println(nim + " - "+nama+" - "+prodi+" - "+kelas+" - ");
    }
}
```

- Source code antrianLayanan.java

```
package p2Jobsheet10;

public class antrianLayanan {
    mahasiswa20[] data;
    int front;
    int rear;
    int size;
    int max;

    public antrianLayanan(int max){
        this.max = max;
        this.data = new mahasiswa20[max];
        this.size = 0;
        this.front = 0;
        this.rear = -1;
    }
}
```



```

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

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

public void liatTerdepan(){
    if (!isEmpty()) {
        System.out.println("Mahasiswa terdepan: "+data[front]);
        System.out.println("NIM - NAMA - PRODI - KELAS");
        data[front].tampilData();
    } else {
        System.out.println("Antrian Kosong");
    }
}

public void print(){
    if (isEmpty()) {
        System.out.println("Antrian Kosong");
        return;
    }

    System.out.println("Daftar Mahasiswa dalam antrian Antrian: ");
    System.out.println("NIM - NAMA - PRODI - KELAS");
    for (int i = 0; i < size; i++) {
        int index = (front+i)%max;
        System.out.print((i+1)+" . ");
        data[index].tampilData();
    }
}

public int getJmlAntri(){
    return size;
}

```

```

public void tambahAntrian(mahasiswa20 mhs){
    if (isFull()) {
        System.out.println("Data sudah penuh!");
        return;
    }
    rear = (rear +1)% max;
    data[rear] = mhs;
    size++;
    System.out.println(mhs.nama + " Berhasil masuk ke antrian");
}
public mahasiswa20 layaniMhs(){
    if (isEmpty()) {
        System.out.println("Queue sudah kosong!");
        return null;
    }
    mahasiswa20 mhs = data[front];
    front = (front+1) % max;
    size--;
    return mhs;
}
}

```

- Source code layananAkademikSiakad.java

```

package p2Jobsheet10;
import java.util.Scanner;
public class layananAkademikSiakad {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        antrianLayanan antrian = new antrianLayanan(5);
        int pilih;
        do {
            System.out.println("\n === Menu Antrian Layanan Akademik === ");
            System.out.println("1. Tambah Antrian Mahasiswa");
            System.out.println("2. Layani Mahasiswa");
            System.out.println("3. Lihat Mahasiswa Terdepan");
            System.out.println("4. Lihat Semua Antrian");
            System.out.println("5. Jumlah Mahasiswa dalam antrian");

```

```
System.out.println("5. Jumlah Mahasiswa dalam antrian");

    System.out.println("0. Keluar");
    System.out.print("Pilih menu: ");
    pilih = sc.nextInt();
    sc.nextLine();
    switch (pilih) {
        case 1:
            System.out.print("NIM    :");
            String nim = sc.nextLine();
            System.out.print("NAMA  : ");
            String nama = sc.nextLine();
            System.out.print("PRODI :");
            String prodi = sc.nextLine();
            System.out.print("KELAS :");
            String kelas = sc.nextLine();
            mahasiswa20 mhs = new mahasiswa20(nim, nama, prodi, kelas);
            antrian.tambahAntrian(mhs);
            break;
        case 2:
            mahasiswa20 dilayani = antrian.layaniMhs();
            if (dilayani != null) {
                System.out.print("Melayani Mahasiswa :");
                dilayani.tampilData();
            }
            break;
        case 3:
            antrian.liatTerdepan();
            break;
        case 4:
            antrian.print();
            break;
        case 5:
            antrian.getJmlAntri();
            break;
        case 0:
            System.out.println(" === Thanks You === ");
        default:
            System.out.println("Pilih menu dengan benar");
            break;
    }
```

```
    }  
    } while (pilih != 0 || pilih > 5);  
}  
}
```

- Pertanyaan

Memodifikasi Code dengan menambahkan method baru bernama lihatAkhir pada class antrianLayanan, menambahkan pula daftar menu **6. Cek antrian paling belakang** pada class layananAkademikSiakad.
potongan code antrianLayanan:

```
public void liatBelakang() {  
    if (isEmpty()) {  
        System.out.println("Antrian Kosong");  
    } else {  
        System.out.println("Mahasiswa Belakang Sendiri: "+data[rear]);  
        System.out.println("NIM - NAMA - PRODI - KELAS");  
        data[rear].tampilData();  
    }  
}
```

Potongan code layanan Akademik Siakad:

```
System.out.println("3. Lihat Mahasiswa Terdepan");

        System.out.println("4. Lihat Semua Antrian");

        System.out.println("5. Jumlah Mahasiswa dalam antrian");

        System.out.println("6. Lihat Mahasiswa Belakang");

        System.out.println("0. Keluar");

        System.out.print("Pilih menu: ");

        pilih = sc.nextInt();
```

```
                case 5:

                        antrian.getJmlAntri();

                        break;

                case 6:

                        antrian.liatBelakang();

                        break;

                case 0:

                        System.out.println(" === Thanks You === ");

                default:

                        System.out.println("Pilih menu dengan benar");

                        break;

                }

        }

        } while (pilih != 0 || pilih > 6);

    }

}
```

- Tugas

Source code class krs20:

```

package tugasJobshet10;

public class krs20 {
    String nama;
    String nim;
    String kelas;

    public krs20(String nama, String nim, String kelas){
        this.nama = nama;
        this.nim = nim;
        this.kelas = kelas;
    }

    public void tampilData(){
        System.out.println(nama+" | "+nim+" | "+kelas+" | ");
    }
}

```

Source code class antrian:

```

package tugasJobshet10;
import p2Jobsheet10.mahasiswa20;
public class antrian {
    krs20[] data;
    int front;
    int rear;
    int size;
    int max;
    int totalProsesKRS = 0;
    public antrian(int max){
        this.max = max;
        this.data = new krs20[max];
        this.front = 0;
        this.size = 0;
        this.rear = -1;
    }
    public boolean isFull(){
        if (size == max) {
            return true;
        } else {
            return false;
        }
    }
}

```

```

    }

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

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

    public void tambahAntrian(krs20 krs){
        if (isFull()) {
            System.out.println("Data sudah penuh!");
            return;
        }
        rear = (rear +1)% max;
        data[rear] = krs;
        size++;
        System.out.println(krs.nama + " Berhasil masuk ke antrian");
    }

    public void layaniKrs20(){
        if (size < 2) {
            System.out.println("Antrian Kurang dari 2, tidak bisa memproses");
            return;
        }
        System.out.println("Memproses KRS untuk 2 Mahasiswa");
        for (int i = 0; i < 2; i++) {
            krs20 krs = data[front];
            krs.tampilData();
            front = (front+1) % max;
            size--;
        }
    }

```

```

        totalProsesKRS++;
    }
}

public void print(){
    if (isEmpty()) {
        System.out.println("Antrian Kosong");
        return;
    }

    System.out.println("Daftar Mahasiswa dalam antrian: ");
    System.out.println("NIM | NAMA | KELAS");
    for (int i = 0; i < size; i++) {
        int index = (front+i)%max;
        System.out.print((i+1)+" . ");
        data[index].tampilData();
    }
}

public void peekDepan(){
    if (size < 2) {
        System.out.println("Antrian kurang dari 2, tidak dapat diproses");
        return;
    }

    System.out.println("Daftar KRS dalam antrian: ");
    System.out.println("NIM | NAMA | KELAS");
    for (int i = 0; i < 2; i++) {
        int idx = (front+i) % max;
        data[idx].tampilData();
    }
}

public void peekBelakang(){
    if (isEmpty()) {
        System.out.println("Antrian Kosong");
    } else {
        System.out.println("Mahasiswa Belakang Sendiri: "+data[rear]);
        System.out.println("NIM - NAMA - PRODI - KELAS");
        data[rear].tampilData();
    }
}
}

```



```

    public int getJmlProsesKRS() {
        return totalProsesKRS;
    }

    public int getJmlAntri() {
        return size;
    }

    public int getSisaMhs() {
        return 30 - totalProsesKRS;
    }
}

```

Source code class krsDemo20:

```

package tugasJobshet10;

import java.util.Scanner;

public class krsDemo20 {

    public static void menu() {
        System.out.println("\n === Menu Antrian KRS === ");
        System.out.println("1. Tambahkan Antrian KRS Mahasiswa");
        System.out.println("2. Layani KRS Mahasiswa");
        System.out.println("3. Lihat 2 Mahasiswa Terdepan");
        System.out.println("4. Jumlah Mahasiswa dalam Antrian");
        System.out.println("5. Lihat Semua Antrian");
        System.out.println("6. Lihat Mahasiswa Terakhir");
        System.out.println("7. Cek Jumlah antrian");
        System.out.println("8. Cek Sisa Mahasiswa belum KRS");
        System.out.println("9. Kosongkan Antrian");
        System.out.println("0. Keluar");
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        antrian antri = new antrian(10);

        int pilih;

        do {
            menu();

            System.out.print("Pilih Menu: ");

            pilih = sc.nextInt();

            sc.nextLine();

            switch (pilih) {

```

case 1:

```
        System.out.print("NIM :");  
        String nim = sc.nextLine();  
        System.out.print("NAMA :");  
        String nama = sc.nextLine();  
        System.out.print("Kelas :");  
        String kelas = sc.nextLine();  
        krs20 krs = new krs20(nama, nim, kelas);  
        antri.tambahAntrian(krs);  
        break;
```

case 2:

```
        antri.layaniKrs20();  
        break;
```

case 3:

```
        antri.peekDepan();  
        break;
```

case 4:

```
        System.out.println("Jumlah Mahasiswa dalam Antrian: " +  
antri.getJmlAntri());  
        break;
```

case 5:

```
        antri.print();  
        break;
```

case 6:

```
        antri.peekBelakang();  
        break;
```

case 7:

```
        System.out.println("Cek Jumlah antrian: "+antri.getJmlProsesKRS());  
        break;
```

case 8:

```
        System.out.println("Sisa Mahasiswa belum proses KRS:  
"+antri.getSisaMhs());  
        break;
```

case 9:

```
        antri.clear();  
        break;
```

case 0:

```
        System.out.println("Thanks You");  
        break;
```

```

        default:
            break;

    }

    } while (pilih != 0 || pilih > 10);

}
}

```

Compile & Run Program

```

=== Menu Antrian KRS ===
1. Tambahkan Antrian KRS Mahasiswa
2. Layani KRS Mahasiswa
3. Lihat 2 Mahasiswa Terdepan
4. Jumlah Mahasiswa dalam Antrian
5. Lihat Semua Antrian
6. Lihat Mahasiswa Terakhir
7. Cek Jumlah antrian
8. Cek Sisa Mahasiswa belum KRS
9. Kosongkan Antrian
0. Keluar
Pilih Menu: 1
NIM :123
NAMA :asdf
Kelas :1E
asdf Berhasil masuk ke antrian

```

```

Pilih Menu: 1
NIM :123
NAMA :eart
Kelas :1G
eart Berhasil masuk ke antrian

```

```

Pilih Menu: 3
Daftar KRS dalam antrian:
NIM | NAMA | KELAS
asdf | 123 | 1E |
eart | 123 | 1G |

```

```

Pilih Menu: 4
Jumlah Mahasiswa dalam Antrian: 2

```

```

Pilih Menu: 5
Daftar Mahasiswa dalam antrian:
NIM | NAMA | KELAS
1. asdf | 123 | 1E |
2. eart | 123 | 1G |

```

```

Pilih Menu: 6
Mahasiswa Belakang Sendiri: tugasJobshet10.krs20@7f31245a
NIM - NAMA - PRODI - KELAS
eart | 123 | 1G |

```

```

Pilih Menu: 7
Cek Jumlah antrian: 0

```

```

Pilih Menu: 8
Sisa Mahasiswa belum proses KRS: 30

```

```

Pilih Menu: 2
Memproses KRS untuk 2 Mahasiswa
asdf | 123 | 1E |
eart | 123 | 1G |

```

```

Pilih Menu: 7
Cek Jumlah antrian: 2

```

```

Pilih Menu: 8
Sisa Mahasiswa belum proses KRS: 28

```

```

Pilih Menu: 9
Antrian kosong

```