

NAMA: WAHDA ADELLA PUTRI FEBRIANA

KELAS: 1B/24

NIM: 244107020156

Praktikum 1

1. Buat file queue.java beserta isinya

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

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

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

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

    void peek() {
        if (!isEmpty()) {
            System.out.println("Elemen Terdepan: " + data[front]);
        } else {
            System.out.println("Queue Masih Kosong");
        }
    }

    void print() {
        if (isEmpty()) {
            System.out.println("Queue Masih Kosong");
        } else {
            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);
        }
    }

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

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

    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;
    }
}
```

2. Buat file queueMain.java beserta isinya

```
public class QueueMain {
    public static void menu() {
        System.out.println("Masukkan oprasi 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 input = new Scanner(System.in);

        System.out.print("Masukkan kapasitas queue: ");
        int n = input.nextInt();
        Queue Q = new Queue(n);

        int pilih = 0;

        do {
            menu();
            pilih = input.nextInt();
            switch (pilih) {
                case 1:
                    System.out.print("Masukkan data baru: ");
                    int dataMasuk = input.nextInt();
                    Q.Enqueue(dataMasuk);
                    break;
                case 2:
                    int dataKeluar = Q.Dequeue();
                    if (dataKeluar != 0) {
                        System.out.println("Data yang di keluarkan : " + dataKeluar);
                    }
                    break;
                case 3:
                    Q.print();
                    break;
                case 4:
                    Q.peek();
                    break;
                case 5:
                    Q.clear();
                    break;
                default:
                    break;
            }
        } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5);

        input.close();
    }
}
```

3. Hasil run

```

Masukkan kapasitas queue: 4
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: 31
Masukkan operasi yang diinginkan :
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
4
Elemen Terdepan: 15

```

Pertanyaan

1. Karena front dan rear hitungnya dari index ke 0, sedangkan size adalah jumlah data yang ada
2. Jika index rear sama dengan max -1 maka saat data baru ditambahkan maka rear (data paling belakang) berada pada index ke 0
3. Jika index front sama dengan max -1 maka saat data akan dihapus, maka front (data terdepan) berada di index ke 0
4. Karena front belum tentu nilai nya adalah 0
5. digunakan untuk memutar indeks ke awal array jika sudah sampai akhir, agar queue bisa berjalan secara melingkar

```

6. void Enqueue(int dt) {
    if(IsFull()) {
        System.out.println(x:"Queue sudah penuh");
    } else {

```

```

7. int Dequeue() {
    int dt = 0;
    if(IsEmpty()) {
        System.out.println(x:"Queue Masih Kosong");
        System.exit(1);
    }

```

```

void Enqueue(int dt) {
    if(IsFull()) {
        System.out.println(x:"Queue sudah penuh");
        System.exit(status:1);
    }

```

Praktikum 2

1. Buat class Mahasiswa.java

```
public class Mahasiswa {
    String nim, nama, prodi, kelas;

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

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

2. File AntriLayanan.java

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

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

    boolean isEmpty() { ... }

    boolean IsFull() { ... }

    void lihatTerdepan() {
        if(!isEmpty()) {
            System.out.print(s:"Mahasiswa Terdepan : ");
            System.out.println(x:"NIM - Nama - Prodi - Kelas");
            data[front].tampilkanData();
        } else {
            System.out.println(x:"Antrian Kosong");
        }
    }

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

    void tambahAntrian(Mahasiswa mhs) {
        if(IsFull()) {
            System.out.println(x:"Antrian sudah penuh tidak dapat menambahkan mahasiswa");
            return;
        }

        rear = (rear + 1) % max;
        data[rear] = mhs;
        size++;
        System.out.println(mhs.nama + " berhasil masuk ke antrian");
    }

    void print() {
        if(IsEmpty()) {
            System.out.println(x:"Antrian Kosong");
            return;
        } else {
            System.out.println(x:"Daftar Mahasiswa dalam antrian");
            System.out.println(x:"NIM - Nama - prodi - Kelas");
            for(int i = 0; i < size; i++) {
                int index = (front + i) % max;
                System.out.print((i+1) + ". ");
                data[index].tampilkanData();
            }
        }
    }

    int getJumlahAntrian() {
        return size;
    }

    Mahasiswa layaniMahasiswa() {
        if(IsEmpty()) {
            System.out.println(x:"Antrian kosong");
            return null;
        }

        Mahasiswa mhs = data[front];
        front = (front + 1) % max;
        size--;
        return mhs;
    }
}
```

3. Buat file layananAkademikMahasiswa.java

```
public class LayananAkademikSiakad {
    public static void menu() {
        System.out.println(x:"=== Menu Antrian Layanan Akademik ===");
        System.out.println(x:"1. Tambah Mahasiswa ke Antrian");
        System.out.println(x:"2. Layani Mahasiswa");
        System.out.println(x:"3. Lihat Mahasiswa Terdepan");
        System.out.println(x:"4. Lihat Semua Antrian");
        System.out.println(x:"5. Jumlah Mahasiswa dalam antrian");
        System.out.println(x:"-----");
    }

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

        AntrianLayanan antrian = new AntrianLayanan(max:5);
        int pilihan;

        do {
            menu();
            pilihan = input.nextInt();
            switch (pilihan) {
                case 1:
                    System.out.print(s:"NIM: ");
                    String nim = input.nextLine();
                    System.out.print(s:"Nama: ");
                    String nama = input.nextLine();
                    System.out.print(s:"Prodi: ");
                    String prodi = input.nextLine();
                    System.out.print(s:"Kelas: ");
                    String kelas = input.nextLine();
                    Mahasiswa mhs = new Mahasiswa(nim, nama, prodi, kelas);
                    antrian.tambahAntrian(mhs);
                    break;
                case 2:
                    Mahasiswa dilayani = antrian.layaniMahasiswa();
                    if(dilayani != null) {
                        System.out.println(x:"Melayani Mahasiswa");
                        dilayani.tampilkanData();
                    }
                    break;
                case 3:
                    antrian.lihatTerdepan();
                    break;
                case 4:
                    antrian.print();
                    break;
                case 5:
                    System.out.println("Jumlah dalam antrian: " + antrian.getJumlahAntrian());
                    break;
                default:
                    System.out.println(x:"Pilihan tidak valid");
                    break;
            }
        } while (pilihan != 0);

        input.close();
    }
}
```

4. Hasil run

```
=== Menu Antrian Layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam antrian
-----
1
NIM: 1
Nama: wahda
Prodi: TI
Kelas: 1B
wahda berhasil masuk ke antrian
=== Menu Antrian Layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam antrian
-----
1
NIM: 2
Nama: Galih
Prodi: TI
Kelas: 1B
Galih berhasil masuk ke antrian

=== Menu Antrian Layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam antrian
-----
4
Daftar Mahasiswa dalam antrian
NIM - Nama - prodi - Kelas
1. 1 - wahda - TI - 1B
2. 2 - Galih - TI - 1B
=== Menu Antrian Layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam antrian
-----
2
Melayani Mahasiswa
1 - wahda - TI - 1B

=== Menu Antrian Layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam antrian
-----
5
Jumlah dalam antrian: 1
```

Pertanyaan

```
void lihatTerbelakang() {
    if(!isEmpty()) {
        System.out.print(s:"Mahasiswa Terbelakang : ");
        System.out.println(x:"NIM - Nama - Prodi - Kelas");
        data[rear].tampilkanData();
    } else {
        System.out.println(x:"Antrian Kosong");
    }
}

case 6:
    antrian.lihatTerbelakang();
    break;
```

Tugas

1. Mahasiswa.java

```
public class Mahasiswa {
    String nim, nama, prodi, kelas;

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

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

2. AntrianLayanan.java

```
public class AntrianLayanan {
    Mahasiswa[] data;
    int front, rear, size, max, sudahDilayani = 0;

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

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

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

    void tampilkanTerdepan() {
        if(!isEmpty()) {
            System.out.println("Mahasiswa Terdepan:");
            System.out.println("NIM - Nama - Prodi - Kelas");
            data[front].tampilkanData();
            if(size > 1) {
                int second = (front + 1) % max;
                data[second].tampilkanData();
            } else {
                System.out.println("Hanya satu antrian tersedia");
            }
        } else {
            System.out.println("Antrian Kosong");
        }
    }

    void tampilkanTerbelakang() {
        if(!isEmpty()) {
            System.out.println("Mahasiswa Terbelakang :");
            System.out.println("NIM - Nama - Prodi - Kelas index ke " + rear);
            data[rear].tampilkanData();
        } else {
            System.out.println("Antrian Kosong");
        }
    }

    void print() {
        if(!isEmpty()) {
            System.out.println("Antrian Kosong");
            return;
        } else {
            System.out.println("Daftar Mahasiswa dalam antrian");
            System.out.println("NIM - Nama - Prodi - Kelas");
            for(int i = 0; i < size; i++) {
                int index = (front + i) % max;
                System.out.println(i + " : ");
                data[index].tampilkanData();
            }
        }
    }

    void CheckAntrianKosong() {
        if(!isEmpty()) {
            System.out.println("Antrian Masih Kosong");
        } else {
            System.out.println("Masih Tersisa " + (max - size) + " posisi kosong");
        }
    }

    void CheckAntrianPenuh() {
        if(isFull()) {
            System.out.println("Antrian Penuh");
        } else {
            System.out.println("Masih Tersisa " + (max - size) + " posisi kosong");
        }
    }

    void MengosongkanAntrian() {
        front = 0;
        rear = -1;
        size = 0;
        System.out.println("Antrian berhasil dikosongkan");
    }

    void tambahAntrian(Mahasiswa mhs) {
        if(isFull() || size >= 10) {
            System.out.println("Antrian sudah penuh tidak dapat menambahkan mahasiswa");
            return;
        }
        if(front == -1) {
            front = 0;
        }
        rear = (rear + 1) % max;
        data[rear] = mhs;
        size++;
        System.out.println(mhs.nama + " berhasil masuk ke antrian index ke " + rear);
    }

    Mahasiswa layaniMahasiswa() {
        if(!isEmpty()) {
            System.out.println("Antrian kosong");
            return null;
        } else if (sudahDilayani >= 30) {
            System.out.println("Dosen Tidak Dapat Menyetujui KRS lagi");
            return null;
        }
        System.out.println("Index ke " + front);
        Mahasiswa mhs = data[front];
        front = (front + 1) % max;
        size--;
        sudahDilayani++;
        return mhs;
    }

    int getJumlahAntrian() {
        return size;
    }

    int getJumlahSudahDilayani() {
        return sudahDilayani;
    }
}
```

3. LayananPersetujuanKrs.java

```
import java.util.Scanner;

public class LayananPersetujuanKRS {
    public static void menu() {
        System.out.println(x:"== Menu Antrian Layanan Akademik ==");
        System.out.println(x:"1. Check Antrian Kosong");
        System.out.println(x:"2. Check Antrian Penuh");
        System.out.println(x:"3. Mengosongkan Antrian");
        System.out.println(x:"4. Menambahkan Antrian");
        System.out.println(x:"5. Meproses Antrian");
        System.out.println(x:"6. Menampilkan Semua Antrian");
        System.out.println(x:"7. Menampilkan 2 Antrian Terdepan");
        System.out.println(x:"8. Menampilkan Antrian Paling Terakhir");
        System.out.println(x:"9. Cetak Jumlah Antrian");
        System.out.println(x:"10. Cetak Jumlah Yang Sudah Diproses");
        System.out.println(x:"-----");
    }

    Run | Debug
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        AntrianLayanan antrian = new AntrianLayanan(max:10);
        int pilihan;

        do {
            menu();
            pilihan = input.nextInt();
            switch (pilihan) {
                case 1:
                    antrian.CheckAntrianKosong();
                    break;
                case 2:
                    antrian.CheckAntrianPenuh();
                    break;
                case 3:
                    antrian.MengosongkanAntrian();
                    break;
                case 4:
                    antrian.MenambahkanAntrian();
                    break;
                case 5:
                    antrian.MeprosesAntrian();
                    break;
                case 6:
                    antrian.MenampilkanSemuaAntrian();
                    break;
                case 7:
                    antrian.Menampilkan2AntrianTerdepan();
                    break;
                case 8:
                    antrian.MenampilkanAntrianPalingTerakhir();
                    break;
                case 9:
                    antrian.CetakJumlahAntrian();
                    break;
                case 10:
                    antrian.CetakJumlahYangSudahDiproses();
                    break;
            }
        } while (pilihan != 0);
    }
}

case 4:
    input.nextLine();
    System.out.println(x:"-- Menambahkan Antrian --");
    System.out.print(s:"MIM: ");
    String nim = input.nextLine();
    System.out.print(s:"Nama: ");
    String nama = input.nextLine();
    System.out.print(s:"Prodi: ");
    String prodi = input.nextLine();
    System.out.print(s:"Kelas: ");
    String kelas = input.nextLine();
    Mahasiswa mhs = new Mahasiswa(nim, nama, prodi, kelas);
    antrian.tambahAntrian(mhs);
    break;

case 5:
    for(int i = 0; i < 2; i++) {
        Mahasiswa dilayani = antrian.layaniMahasiswa();
        if(dilayani != null) {
            System.out.println(x:"Melayani Mahasiswa");
            dilayani.tampilkanData();
        }
    }
    break;

case 6:
    antrian.print();
    break;

case 7:
    antrian.lihatTerdepan();
    break;

case 8:
    antrian.lihatTerbelakang();
    break;

case 9:
    System.out.println("Jumlah dalam antrian: " + antrian.getJumlahAntrian());
    break;

case 10:
    System.out.println("Jumlah data sudah dilayani: " + antrian.getJumlahSudahDilayani());
    break;
```

Hasil Run

```
KMS
=== Menu Antrian Layanan Akademik ===
1. Check Antrian Kosong
2. Check Antrian Penuh
3. Mengosongkan Antrian
4. Menambahkan Antrian
5. Memproses Antrian
6. Menampilkan Semua Antrian
7. Menampilkan 2 Antrian Terdepan
8. Menampilkan Antrian Paling Terakhir
9. Cetak Jumlah Antrian
10. Cetak Jumlah Yang Sudah Diproses
-----
1
Antrian Masih Kosong
=== Menu Antrian Layanan Akademik ===
1. Check Antrian Kosong
2. Check Antrian Penuh
3. Mengosongkan Antrian
4. Menambahkan Antrian
5. Memproses Antrian
6. Menampilkan Semua Antrian
7. Menampilkan 2 Antrian Terdepan
8. Menampilkan Antrian Paling Terakhir
9. Cetak Jumlah Antrian
10. Cetak Jumlah Yang Sudah Diproses
-----
2
Masih Tersisa 10 posisi kosong

4
-- Menambahkan Antrian --
NIM: 2
Nama: 2
Prodi: 2
Kelas: 2
2 berhasil masuk ke antrian index ke 1
=== Menu Antrian Layanan Akademik ===
1. Check Antrian Kosong
2. Check Antrian Penuh
3. Mengosongkan Antrian
4. Menambahkan Antrian
5. Memproses Antrian
6. Menampilkan Semua Antrian
7. Menampilkan 2 Antrian Terdepan
8. Menampilkan Antrian Paling Terakhir
9. Cetak Jumlah Antrian
10. Cetak Jumlah Yang Sudah Diproses
-----
6
Daftar Mahasiswa dalam antrian
NIM - Nama - Prodi - Kelas
1. 1 - 1 - 1 - 1
2. 2 - 2 - 2 - 2

4
-- Menambahkan Antrian --
NIM: 5
Nama: 5
Prodi: 5
Kelas: 5
5 berhasil masuk ke antrian index ke 2
=== Menu Antrian Layanan Akademik ===
1. Check Antrian Kosong
2. Check Antrian Penuh
3. Mengosongkan Antrian
4. Menambahkan Antrian
5. Memproses Antrian
6. Menampilkan Semua Antrian
7. Menampilkan 2 Antrian Terdepan
8. Menampilkan Antrian Paling Terakhir
9. Cetak Jumlah Antrian
10. Cetak Jumlah Yang Sudah Diproses
-----
7
Mahasiswa Terdepan:
NIM - Nama - Prodi - Kelas
1 - 1 - 1 - 1
2 - 2 - 2 - 2
=== Menu Antrian Layanan Akademik ===
1. Check Antrian Kosong
2. Check Antrian Penuh
3. Mengosongkan Antrian
4. Menambahkan Antrian
5. Memproses Antrian
6. Menampilkan Semua Antrian
7. Menampilkan 2 Antrian Terdepan
8. Menampilkan Antrian Paling Terakhir
9. Cetak Jumlah Antrian
10. Cetak Jumlah Yang Sudah Diproses
-----
8
Mahasiswa Terbelakang : NIM - Nama - Prodi - Kelas index ke 1
2 - 2 - 2 - 2

4
-- Menambahkan Antrian --
NIM: 5
Nama: 5
Prodi: 5
Kelas: 5
5 berhasil masuk ke antrian index ke 2
=== Menu Antrian Layanan Akademik ===
1. Check Antrian Kosong
2. Check Antrian Penuh
3. Mengosongkan Antrian
4. Menambahkan Antrian
5. Memproses Antrian
6. Menampilkan Semua Antrian
7. Menampilkan 2 Antrian Terdepan
8. Menampilkan Antrian Paling Terakhir
9. Cetak Jumlah Antrian
10. Cetak Jumlah Yang Sudah Diproses
-----
7
Mahasiswa Terdepan:
NIM - Nama - Prodi - Kelas
5 - 5 - 5 - 5
Antrian Kosong

3
Antrian berhasil dikosongkan
=== Menu Antrian Layanan Akademik ===
1. Check Antrian Kosong
2. Check Antrian Penuh
3. Mengosongkan Antrian
4. Menambahkan Antrian
5. Memproses Antrian
6. Menampilkan Semua Antrian
7. Menampilkan 2 Antrian Terdepan
8. Menampilkan Antrian Paling Terakhir
9. Cetak Jumlah Antrian
10. Cetak Jumlah Yang Sudah Diproses
-----
6
Antrian Kosong
```

+-----+

| Mahasiswa |

+-----+

| - nim: String |

| - nama: String |

| - prodi: String |

| - kelas: String |

+-----+

| + Mahasiswa(nim, |

| nama, prodi, kelas)|

| + tampilkanData(): |

| void |

+-----+

▲

|

|

+-----+

| AntrianLayanan |

+-----+

| - data: Mahasiswa[] |

| - front: int |

| - rear: int |

| - size: int |

| - max: int |

| - sudahDilayani: int |

+-----+

| + AntrianLayanan(max) |

| + IsEmpty(): boolean |

| + IsFull(): boolean |

| + tambahAntrian(Mahasiswa): void |

```
| + layaniMahasiswa(): Mahasiswa |
| + print(): void                |
| + lihatTerdepan(): void        |
| + lihatTerbelakang(): void     |
| + CheckAntrianKosong(): void    |
| + CheckAntrianPenuh(): void     |
| + MengosongkanAntrian(): void   |
| + getJumlahAntrian(): int       |
| + getJumlahSudahDilayani(): int |
```

```
+-----+
```



```
|
```

```
|
```

```
+-----+
```

```
| LayananPersetujuanKRS |
```

```
+-----+
```

```
| + menu(): void          |
```

```
| + main(String[]): void  |
```

```
+-----+
```