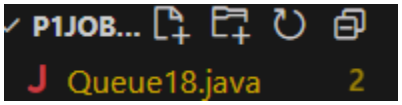
Nama : Rachmad Aprisandhy
Kelas : TI-1B
NO Abs : 18

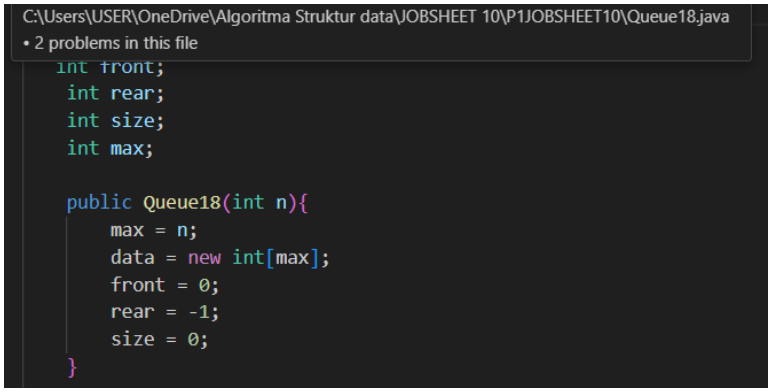
# JOBSHEET 10

## A. Praktikum 1

1) Buat folder dan class baru Bernama p1JOBSHEET10 dan class Queue




2) Tambahkan atribut-atribut Queue sesuai diagram class, kemudian tambahkan pula konstruktornya




```
int front;
int rear;
int size;
int max;

public Queue18(int n){
    max = n;
    data = new int[max];
    front = 0;
    rear = -1;
    size = 0;
}
```

3) Buat method IsEmpty bertipe boolean yang digunakan untuk mengecek apakah queue kosong.

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

4) Buat method IsFull bertipe boolean yang digunakan untuk mengecek apakah queue sudah penuh.

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

5) Buat method peek bertipe void untuk menampilkan elemen queue pada posisi paling depan.

```
public void peek() {
    if (!isEmpty()) {
        System.out.println("Element terdepan :" + data[front]);
    } else {
        System.out.println(x:"data masih kosong");
    }
}
```

6) Buat method print bertipe void untuk menampilkan seluruh elemen pada queue mulai dari posisi front sampai dengan posisi rear.

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

7) Buat method clear bertipe void untuk menghapus semua elemen pada queue

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

8) Buat method Enqueue bertipe void untuk menambahkan isi queue dengan parameter dt yang bertipe integer

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

9) Buat method Dequeue bertipe int untuk mengeluarkan data pada queue di posisi belakang

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

10) buat class baru dengan nama QueueMain tetap pada package Praktikum1. Buat method menu bertipe void untuk memilih menu program pada saat dijalankan.

```
import java.util.Scanner;
public  class QueueMain18 {
    public static void menu() {
        System.out.println (x:"Masukkan operasi yang anda inginkan:");
        System.out.println (x:"1. Enqueue");
        System.out.println (x:"2. Dequeue");
        System.out.println (x:"3. Print");
        System.out.println (x:"4. Peek");
        System.out.println (x:"5. Clear");
        System.out.println (x:"---------------------------------------");

    }
```

11) Buat fungsi main, kemudian deklarasikan Scanner dengan nama sc. Buat variabel n untuk menampung masukan berupa jumlah maksimal elemen yang dapat disimpan pada queue.

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

    System.out.print(s:"Masukkan kapasitas queue: ");
    int n = sc.nextInt();
    Queue18 Q = new Queue18(n);
```

12) Deklarasikan variabel dengan nama pilih bertipe integer untuk menampung pilih menu dari pengguna.

```
int pilih;
```

13) Lakukan perulangan menggunakan do-while untuk menjalankan program secara terus menerus sesuai masukan yang diberikan. Di dalam perulangan tersebut, terdapat pemilihan kondisi menggunakan switch-case untuk menjalankan operasi queue sesuai dengan masukan pengguna.

```
        do {
            menu ();
            pilih = sc.nextInt();
            switch (pilih) {
                case 1:
                    System.out.print(s:"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(x:"Pilihan tidak valid. Silakan coba lagi.");
            }
        } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5);
    }
}
```

14) Hasilnya

```
Masukkan operasi yang anda inginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-------------------------------------
1
Masukkan data baru: 31
Masukkan operasi yang anda inginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-------------------------------------
4
Element terdepan :15
Masukkan operasi yang anda inginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-------------------------------------
```

B. Pertanyaan

1. Pada konstruktor, mengapa nilai awal atribut front dan rear bernilai -1, sementara atribut size bernilai 0?
   Nilai front dan rear ditunjuk ke -1 agar dapat digunakan sebagai pertanda queue masih kosong sementara size 0 menunjukkan jumlah elemen dalam queue dimulai dari angka 0
2. Pada method Enqueue, jelaskan maksud dan kegunaan dari potongan kode berikut!

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

   **Jika rear berada di akhir array (max - 1), maka kita perlu mengembalikan kembali ke awal array, yaitu posisi 0.**
3. Pada method Dequeue, jelaskan maksud dan kegunaan dari potongan kode berikut!

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

   Jika front sudah mencapai indeks akhir (max - 1), kita kembali ke awal (0) agar queue tetap berjalan.

4. Pada method print, mengapa pada proses perulangan variabel i tidak dimulai dari 0 (int i=0), melainkan int i=front?
   karena data tidak selalu dimulai dengan indeks 0
5. Perhatikan kembali method print, jelaskan maksud dari potongan kode berikut!

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

   Potongan ini digunakan untuk **melakukan iterasi circular**.Jika i + 1 masih di dalam batas array, lanjut ke indeks berikutnya.Jika i + 1 == max, maka akan kembali ke 0.
6. Tunjukkan potongan kode program yang merupakan queue overflow!

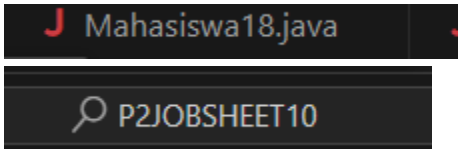
```
public void enqueue ( int dt) {
    if (isFull()) {
        System.out.println(x:"Queue sudah penuh");
```

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!

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

C. Percobaan 2

1. Buat folder baru bernama P2Jobsheet10 di dalam repository Praktikum ASD, kemudian buat class baru dengan nama Mahasiswa




2. Tambahkan atribut-atribut Nasabah seperti pada Class Diagram, kemudian tambahkan pula konstruktornya seperti gambar berikut ini.

```java
public class Mahasiswa18 {

    String NIM;
    String nama;
    String prodi;
    String kelas;


    public Mahasiswa18(String NIM, String nama, String prodi, String kelas) {
        this.NIM = NIM;
        this.nama = nama;
        this.prodi = prodi;
        this.kelas = kelas;
    }

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

3. Salin kode program class Queue pada Praktikum 1 untuk digunakan kembali pada Praktikum 2 ini, ganti nama class-nya dengan AntrianLayanan

```java
public class AntrianLayanan18 {
    Mahasiswa18[] data;
    int front;
     int rear;
     int size;
     int max;

    public AntrianLayanan18 (int max){
        this.max = max;
        this.data = new Mahasiswa18[max];
        this.front = 0;
        this.rear = -1;
        this.size = 0;
    }

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

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

    public void lihatTerdepan() {
        if (!isEmpty()) {
            System.out.println(x:"Antrian kosong ");
        } else {
            System.out.println(x:"mahasiswa terdepan: ");
```

```java
    } else {
        System.out.println(x:"mahasiswa terdepan: ");
        System.out.println(x:"NIM - NAMA - PRODI - KELAS");
        data[front].tampilkanData();
    }
}

public void tampilkanSemua  () {
    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.err.println((i + 1) + ". ");
            data[index].tampilkanData();
        }
    }
}

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

public void TambahAntrian (Mahasiswa18 mhs ) {
    if (isFull()) {
```

4. Selanjutnya, buat class baru dengan nama LayananAkademikSIAKAD tetap pada package yang sama. Buat fungsi main, deklarasikan Scanner dengan nama sc.

```
C:\Users\USER\OneDrive\Algoritma Struktur data\JOBSHEET
10\P2JOBSHEET10\LayananAkademikSiakad18.java • 2 problems in this file
```

```java
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner sc  = new Scanner(System.in);
        AntrianLayanan18 antrian = new AntrianLayanan18(max:5);
        int pilihan;
```

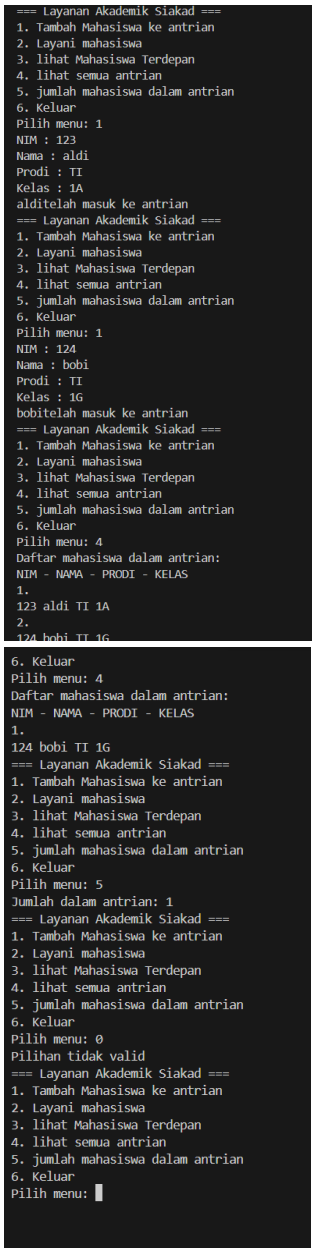
5. Tambahkan kode berikut untuk melakukan perulangan menu sesuai dengan masukan yang diberikan oleh pengguna.

```
import java.util.Scanner;
public class LayananAkademikSiakad18 {

    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner sc  = new Scanner(System.in);
        AntrianLayanan18 antrian = new AntrianLayanan18(max:5);
        int pilihan;
        do {
            System.out.println(x:"=== Layanan Akademik Siakad ===");
            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:"6. Keluar");
            System.out.print(s:"Pilih menu: ");
            pilihan = sc.nextInt();
            sc.nextLine();

        switch  (pilihan) {
            case 1:
                System.out.print (s:"NIM : ");
                String NIM = sc.nextLine();
                System.out.print (s:"Nama : ");
                String nama = sc.nextLine();
                System.out.print (s:"Prodi : ");
                String prodi = sc.nextLine();
                System.out.print (s:"Kelas : ");
                String kelas = sc.nextLine();
                Mahasiswa18 mhs = new Mahasiswa18(NIM, nama, prodi, kelas);
                antrian.TambahAntrian(mhs);
                break;
            case 2:
                Mahasiswa18 mahasiswadilayani = antrian.LayaniMahasiswa();
                if (mahasiswadilayani != null) {
                    System.out.println(x:"Melayani mahasiswa : ");
```



```
                String kelas = sc.nextLine();
                Mahasiswa18 mhs = new Mahasiswa18(NIM, nama, prodi, kelas);
                antrian.TambahAntrian(mhs);
                break;
            case 2:
                Mahasiswa18 mahasiswadilayani = antrian.LayaniMahasiswa();
                if (mahasiswadilayani != null) {
                    System.out.println(x:"Melayani mahasiswa : ");
                    mahasiswadilayani.tampilkanData();
                }
                break;
            case 3:
                antrian.lihatTerdepan();
                break;
            case 4:
                antrian.tampilkanSemua();
                break;
            case 5:
                System.out.println("Jumlah dalam antrian: " + antrian.getJumlahAntrian());
                break;
            case 6:
                System.out.println(x:"Terima kasih telah");
                break;
            default:
                System.out.println(x:"Pilihan tidak valid");
        }

    }while (pilihan != 6);
        sc.close();
    }
}
```

6. Hasilnya


=== Layanan Akademik Siakad ===
1. Tambah Mahasiswa ke antrian
2. Layani mahasiswa
3. lihat Mahasiswa Terdepan
4. lihat semua antrian
5. jumlah mahasiswa dalam antrian
6. Keluar
Pilih menu: 1
NIM : 123
Nama : aldi
Prodi : TI
Kelas : 1A
alditelah masuk ke antrian
=== Layanan Akademik Siakad ===
1. Tambah Mahasiswa ke antrian
2. Layani mahasiswa
3. lihat Mahasiswa Terdepan
4. lihat semua antrian
5. jumlah mahasiswa dalam antrian
6. Keluar
Pilih menu: 1
NIM : 124
Nama : bobi
Prodi : TI
Kelas : 1G
bobitelah masuk ke antrian
=== Layanan Akademik Siakad ===
1. Tambah Mahasiswa ke antrian
2. Layani mahasiswa
3. lihat Mahasiswa Terdepan
4. lihat semua antrian
5. jumlah mahasiswa dalam antrian
6. Keluar
Pilih menu: 4
Daftar mahasiswa dalam antrian:
NIM - NAMA - PRODI - KELAS
1.
123 aldi TI 1A
2.
124 bobi TI 1G
6. Keluar
Pilih menu: 4
Daftar mahasiswa dalam antrian:
NIM - NAMA - PRODI - KELAS
1.
124 bobi TI 1G
=== Layanan Akademik Siakad ===
1. Tambah Mahasiswa ke antrian
2. Layani mahasiswa
3. lihat Mahasiswa Terdepan
4. lihat semua antrian
5. jumlah mahasiswa dalam antrian
6. Keluar
Pilih menu: 5
Jumlah dalam antrian: 1
=== Layanan Akademik Siakad ===
1. Tambah Mahasiswa ke antrian
2. Layani mahasiswa
3. lihat Mahasiswa Terdepan
4. lihat semua antrian
5. jumlah mahasiswa dalam antrian
6. Keluar
Pilih menu: 0
Pilihan tidak valid
=== Layanan Akademik Siakad ===
1. Tambah Mahasiswa ke antrian
2. Layani mahasiswa
3. lihat Mahasiswa Terdepan
4. lihat semua antrian
5. jumlah mahasiswa dalam antrian
6. Keluar
Pilih menu:

D. Pertanyaan

Lakukan modifikasi program dengan menambahkan method baru bernama LihatAkhir pada class AntrianLayanan yang digunakan untuk mengecek antrian yang berada di posisi belakang. Tambahkan pula daftar menu 6. Cek Antrian paling belakang pada class LayananAkademikSIAKAD sehingga method LihatAkhir dapat dipanggil!

```
public void LihatAkhir() {
    if (isEmpty()) {
        System.out.println(x:"Antrian kosong");
    } else {
        System.out.println(x:"Mahasiswa di posisi paling belakang:");
        System.out.println(x:"NIM - NAMA - PRODI - KELAS");
        data[rear].tampilkanData();
    }
}

}
```

```
case 6:
    antrian.LihatAkhir();
    break;
case 7:
    System.out.println(x:"Terima kasih telah menggunakan layan
    break;
default:
    System.out.println(x:"Pilihan tidak valid");
```

```
=== Layanan Akademik Siakad ===
1. Tambah Mahasiswa ke antrian
2. Layani mahasiswa
3. lihat Mahasiswa Terdepan
4. lihat semua antrian
5. jumlah mahasiswa dalam antrian
6. cek antrian paling belakang
7. Keluar
Pilih menu: 1
NIM : 123
Nama : paidi
Prodi : TI
Kelas : 1B
paiditelah masuk ke antrian
=== Layanan Akademik Siakad ===
1. Tambah Mahasiswa ke antrian
2. Layani mahasiswa
3. lihat Mahasiswa Terdepan
4. lihat semua antrian
5. jumlah mahasiswa dalam antrian
6. cek antrian paling belakang
7. Keluar
Pilih menu: 1
NIM : 124
Nama : wahyu
Prodi : TI
Kelas : 1G
wahyutelah masuk ke antrian
=== Layanan Akademik Siakad ===
1. Tambah Mahasiswa ke antrian
2. Layani mahasiswa
3. lihat Mahasiswa Terdepan
4. lihat semua antrian
5. jumlah mahasiswa dalam antrian
6. cek antrian paling belakang
7. Keluar
Pilih menu: 6
Mahasiswa di posisi paling belakang:
NIM - NAMA - PRODI - KELAS
124 wahyu TI 1G
PS C:\Users\USER\OneDrive\Algoritma Struktur data\JOBSHEET 10\P2JOBSHEET10>
```

E. Tugas

```java
public class Mahasiswa18 {
    String nim, nama, prodi, kelas;

    public Mahasiswa18(String nim, String nama, String prodi, String kelas) {
        this.nim = nim;
        this.nama = nama;
        this.prodi = prodi;
        this.kelas = kelas;
    }

    public String toString() {
        return nim + " - " + nama + " - " + prodi + " - " + kelas;
    }

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

```java
public class AntrianKRS18 {
    Mahasiswa18[] data;
    int front;
    int rear;
    int size;
    int max;
    int totalDilayani;

    public AntrianKRS18(int max) {
        this.max = max;
        data = new Mahasiswa18[max];
        front = 0;
        rear = -1;
        size = 0;
        totalDilayani = 0;
    }

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

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

    public void TambahAntrian(Mahasiswa18 mhs) {
        if (isFull()) {
            System.out.println(x:"Antrian penuh. Tidak bisa menambah.");
            return;
        }
        rear = (rear + 1) % max;
        data[rear] = mhs;
        size++;
        System.out.println(mhs.nama + " telah ditambahkan ke antrian.");
    }
```

```java
    public void ProsesKRS() {
        if (size < 2) {
            System.out.println(x:"Tidak cukup mahasiswa untuk proses
            return;
        }
        System.out.println(x:"Memproses 2 mahasiswa:");
        for (int i = 0; i < 2; i++) {
            Mahasiswa18 mhs = LayaniMahasiswa();
            System.out.print(s:"- ");
            mhs.tampilkanData();
        }
    }

    public Mahasiswa18 LayaniMahasiswa() {
        if (isEmpty()) {
            System.out.println(x:"Antrian kosong.");
            return null;
        }
        Mahasiswa18 mhs = data[front];
        front = (front + 1) % max;
        size--;
        totalDilayani++;
        return mhs;
    }

    public void tampilkanSemua() {
        if (isEmpty()) {
            System.out.println(x:"Antrian kosong.");
            return;
        }
        System.out.println(x:"Daftar mahasiswa dalam antrian:");
        for (int i = 0; i < size; i++) {
            int index = (front + i) % max;
            System.out.print((i + 1) + ". ");
            data[index].tampilkanData();
```

ava: Ready

```
            for (int i = 0; i < Math.min(a:2, size); i++) {
                int index = (front + i) % max;
                data[index].tampilkanData();
            }
        }
    }

    public void tampilkanTerakhir() {
        if (isEmpty()) {
            System.out.println(x:"Antrian kosong.");
        } else {
            data[rear].tampilkanData();
        }
    }

    public void clear() {
        front = 0;
        rear = -1;
        size = 0;
        System.out.println(x:"Antrian dikosongkan.");
    }

    public int getJumlahAntrian() {
        return size;
    }

    public int getJumlahSudahProsesKRS() {
        return totalDilayani;
    }

    public int getSisaBelumProses() {
        return 30 - totalDilayani;
    }
}
```

```
import java.util.Scanner;

public class KRS18Main {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        AntrianKRS18 antrian = new AntrianKRS18(max:10);

        int pilihan;
        do {
            System.out.println(x:"\n=== Layanan Akademik Siakad ===");
            System.out.println(x:"1. Tambah Mahasiswa ke antrian");
            System.out.println(x:"2. Proses 2 mahasiswa (KRS)");
            System.out.println(x:"3. Lihat 2 mahasiswa terdepan");
            System.out.println(x:"4. Lihat semua antrian");
            System.out.println(x:"5. Lihat mahasiswa terakhir");
            System.out.println(x:"6. Cek antrian kosong/penuh");
            System.out.println(x:"7. Kosongkan antrian");
            System.out.println(x:"8. Cetak jumlah antrian");
            System.out.println(x:"9. Cetak jumlah sudah proses & belum");
            System.out.println(x:"10. Keluar");
            System.out.print(s:"Pilih menu: ");
            pilihan = sc.nextInt();
            sc.nextLine();

            switch (pilihan) {
                case 1:
                    System.out.print(s:"NIM   : ");
                    String nim = sc.nextLine();
                    System.out.print(s:"Nama  : ");
                    String nama = sc.nextLine();
                    System.out.print(s:"Prodi : ");
                    String prodi = sc.nextLine();
                    System.out.print(s:"Kelas : ");
                    String kelas = sc.nextLine();
                    Mahasiswa18 mhs = new Mahasiswa18(nim, nama, prodi, kelas);
                    antrian.TambahAntrian(mhs);
```

```
                    antrian.ProsesKRS();
                    break;
                case 3:
                    antrian.tampilkan2Terdepan();
                    break;
                case 4:
                    antrian.tampilkanSemua();
                    break;
                case 5:
                    System.out.println(x:"Mahasiswa terakhir:");
                    antrian.tampilkanTerakhir();
                    break;
                case 6:
                    System.out.println("Antrian kosong? " + antrian.isEmpty());
                    System.out.println("Antrian penuh? " + antrian.isFull());
                    break;
                case 7:
                    antrian.clear();
                    break;
                case 8:
                    System.out.println("Jumlah mahasiswa dalam antrian: " + antrian.getJumlahAntrian(
                    break;
                case 9:
                    System.out.println("Sudah proses KRS: " + antrian.getJumlahSudahProsesKRS());
                    System.out.println("Sisa belum proses: " + antrian.getSisaBelumProses());
                    break;
                case 10:
                    System.out.println(x:"Terima kasih!");
                    break;
                default:
                    System.out.println(x:"Pilihan tidak valid.");
            }
        } while (pilihan != 10);
    }
}
```

```
trl+Shift+M) - Total 11 Problems
KRS)
3. Lihat 2 mahasiswa terdepan
4. Lihat semua antrian
5. Lihat mahasiswa terakhir
6. Cek antrian kosong/penuh
7. Kosongkan antrian
8. Cetak jumlah antrian
9. Cetak jumlah sudah proses & belum
10. Keluar
Pilih menu: 8
Jumlah mahasiswa dalam antrian: 3

=== Layanan Akademik Siakad ===
1. Tambah Mahasiswa ke antrian
2. Proses 2 mahasiswa (KRS)
3. Lihat 2 mahasiswa terdepan
4. Lihat semua antrian
5. Lihat mahasiswa terakhir
6. Cek antrian kosong/penuh
7. Kosongkan antrian
8. Cetak jumlah antrian
9. Cetak jumlah sudah proses & belum
10. Keluar
Pilih menu: 9
Sudah proses KRS: 0
Sisa belum proses: 30

=== Layanan Akademik Siakad ===
1. Tambah Mahasiswa ke antrian
2. Proses 2 mahasiswa (KRS)
3. Lihat 2 mahasiswa terdepan
4. Lihat semua antrian
5. Lihat mahasiswa terakhir
6. Cek antrian kosong/penuh
7. Kosongkan antrian
8. Cetak jumlah antrian
9. Cetak jumlah sudah proses & belum
10. Keluar
Pilih menu:
PS C:\Users\USER\OneDrive\Algoritma Struktur data\JOBSHEET 10\Tugas> ^C
PS C:\Users\USER\OneDrive\Algoritma Struktur data\JOBSHEET 10\Tugas>
eted.  Java: Ready
```