

POLITEKNIK NEGERI MALANG
TEKNOLOGI INFORMASI
TEKNIK INFORMATIKA



Nama: Abdul Rahman Hanif Darmawan

NIM: 244107020232

Kelas: TI-1A

Prodi: D4-TEKNIK INFORMATIKA

JOBSHEET 10

2.1 Percobaan 1 : Operasi Dasar Queue

2.1.2 Verifikasi Hasil Percobaan

```
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
```

2.1.3 Pertanyaan

1. Pada konstruktor, mengapa nilai awal atribut front dan rear bernilai -1, sementara atribut size bernilai 0?
 - Nilai -1 menandakan bahwa queue dalam keadaan kosong sedangkan size=0 menunjukkan bahwa elemen dalam queue adalah nol.
2. Pada method Enqueue, jelaskan maksud dan kegunaan dari potongan kode berikut!

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

- Kode tersebut berfungsi untuk mengimplementasikan queue melingkar jika rear sudah di indeks terakhir array, maka setelahnya akan dilanjutkan kembali ke indeks 0.
3. Pada method Dequeue, jelaskan maksud dan kegunaan dari potongan kode berikut!

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

- Mirip dengan nomor 2, jika front mencapai indeks terakhir (max - 1), maka indeks pengambilan data selanjutnya akan kembali ke 0.
4. Pada method print, mengapa pada proses perulangan variabel i tidak dimulai dari 0 (int i=0), melainkan int i=front?
 - Karena data dalam queue tidak selalu dimulai dari indeks 0 terutama sudah terjadi proses enqueue-dequeue, dan mulai mencetak dari indeks front yang menandai elemen pertama dalam queue
 5. Perhatikan kembali method print, jelaskan maksud dari potongan kode berikut!

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

- Setelah indeks mencapai batas akhir (max - 1), ia akan kembali ke 0, fungsinya agar pencetakan elemen queue bisa menyusuri semua elemen secara sirkuler.
6. Tunjukkan potongan kode program yang merupakan queue overflow!

```
if (IsFull()) {  
    System.out.println(x:"Queue sudah penuh");  
} else {
```

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

2.2 Percobaan 2 : Antrian Layanan Akademik

2.2.2 Verifikasi Hasil Percobaan

```
=== 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
0. Keluar
Pilih menu: 1
NIM: 123
Nama: Aldi
Prodi: TI
Kelas: 1A
Aldi 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
0. Keluar
Pilih menu: 1
NIM: 124
Nama: Bobi
Prodi: TI
Kelas: 1G
Bobi 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
0. Keluar
Pilih menu: 4
Daftar Mahasiswa dalam Antrian:
NIM - NAMA - PRODI - KELAS
1. 123 - Aldi - TI - 1A
2. 124 - Bobi - TI - 1G

=== 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
0. Keluar
Pilih menu: 2
Melayani mahasiswa: 123 - Aldi - TI - 1A

=== 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
0. Keluar
Pilih menu: 4
Daftar Mahasiswa dalam Antrian:
NIM - NAMA - PRODI - KELAS
1. 124 - Bobi - TI - 1G
```

```
=== 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
0. Keluar
Pilih menu: 2
Melayani mahasiswa: 123 - Aldi - TI - 1A

=== 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
0. Keluar
Pilih menu: 4
Daftar Mahasiswa dalam Antrian:
NIM - NAMA - PRODI - KELAS
1. 124 - Bobi - TI - 1G

=== 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
0. Keluar
Pilih menu: 5
Jumlah dalam antrian: 1

=== 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
0. Keluar
Pilih menu: 0
Terima kasih.
```

2.2.3 Pertanyaan

1. 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 LayananAkademikSIKAD sehingga method LihatAkhir dapat dipanggil!

```
do {
    System.out.println("\n=== Menu Antrian Layanan Akademik ===");
    System.out.println("1. Tambah Mahasiswa ke Antrian");
    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("6. Cek Antrian paling belakang");
    System.out.println("0. Keluar");
    System.out.print("Pilih menu: ");
    pilihan = sc.nextInt();
    sc.nextLine();
    switch (pilihan) {
        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();
            Mahasiswa mhs = new Mahasiswa(nim, nama, prodi, kelas);
            antrian.tambahAntrian(mhs);
            break;
        case 2:
            Mahasiswa dilayani = antrian.layaniMahasiswa();
            if (dilayani != null) {
                System.out.print("Melayani mahasiswa: ");
                dilayani.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:
            antrian.lihatAkhir();
            break;
        case 0:
            System.out.println("Terima kasih.");
            break;
        default:
            System.out.println("Pilihan tidak valid.");
    }
} while (pilihan != 0);
sc.close();
}
```

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

2.3 Tugas

AntrianKRS.java

```
package Tugas;
public class AntrianKRS {
    Mahasiswa[] data;
    int front, rear, size, max;
    public AntrianKRS(int max) {
        this.max = max;
        this.data = new Mahasiswa[max];
        this.front = 0;
        this.rear = -1;
        this.size = 0;
    }

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

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

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

    public void tambahAntrian(Mahasiswa mhs) {
        if (isFull()) {
            System.out.println("Antrian penuh!");
            return;
        }
        rear = (rear + 1) % max;
        data[rear] = mhs;
        size++;
        System.out.println("Mahasiswa berhasil ditambahkan ke antrian.");
    }

    public void prosesKRS() {
        if (size < 2) {
            System.out.println("Minimal 2 mahasiswa dibutuhkan untuk proses KRS.");
            return;
        }
        for (int i = 0; i < 2; i++) {
            Mahasiswa mhs = data[front];
            mhs.sudahKRS = true;
            System.out.print("Mahasiswa diproses: ");
            mhs.tampilkanData();
            front = (front + 1) % max;
            size--;
        }
    }

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

```
public void lihatTerdepan() {
    if (!isEmpty()) {
        System.out.print("Mahasiswa terdepan: ");
        data[front].tampilkanData();
    } else {
        System.out.println("Antrian kosong.");
    }
}

public void lihatDuaTerdepan() {
    if (size == 0) {
        System.out.println("Antrian kosong.");
    } else if (size == 1) {
        System.out.println("Hanya 1 mahasiswa dalam antrian.");
        data[front].tampilkanData();
    } else {
        System.out.println("2 mahasiswa terdepan:");
        data[front].tampilkanData();
        data[(front + 1) % max].tampilkanData();
    }
}

public void lihatAkhir() {
    if (!isEmpty()) {
        System.out.print("Mahasiswa paling belakang: ");
        data[rear].tampilkanData();
    } else {
        System.out.println("Antrian kosong.");
    }
}

public int getJumlahAntrian() {
    return size;
}

public int getJumlahSudahKRS() {
    int count = 0;
    for (int i = 0; i < max; i++) {
        if (data[i] != null && data[i].sudahKRS) count++;
    }
    return count;
}

public int getJumlahBelumKRS() {
    return getJumlahAntrian();
}
}
```

Mahasiswa.java

```
package Tugas;

public class Mahasiswa {

    String nim, nama, prodi, kelas;

    boolean sudahKRS = false;

    public Mahasiswa(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 +
(sudahKRS ? " [Sudah KRS]" : ""));

    }

}
```

MainKRS.java

```
package Tugas;
import java.util.Scanner;
public class MainKRS {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        AntrianKRS antrian = new AntrianKRS(10);
        int pilihan;
        do {
            System.out.println("\n=== MENU ANTRIAN KRS ===");
            System.out.println("1. Tambah Mahasiswa");
            System.out.println("2. Proses 2 Mahasiswa (KRS)");
            System.out.println("3. Lihat 2 Terdepan");
            System.out.println("4. Lihat Semua Antrian");
            System.out.println("5. Lihat Antrian Belakang");
            System.out.println("6. Cek Antrian Kosong / Penuh");
            System.out.println("7. Kosongkan Antrian");
            System.out.println("8. Cetak Jumlah Antrian");
            System.out.println("9. Cetak Jumlah Sudah KRS");
            System.out.println("10. Cetak Jumlah Belum KRS");
            System.out.println("0. Keluar");
            System.out.print("Pilih: ");
            pilihan = sc.nextInt();
            sc.nextLine();
            switch (pilihan) {
                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();
                    antrian.tambahAntrian(new Mahasiswa(nim, nama, prodi,
kelas));
                    break;
                case 2:
                    antrian.prosesKRS();
                    break;
                case 3:
                    antrian.lihatDuaTerdepan();
                    break;
                case 4:
                    antrian.tampilkanSemua();
                    break;
                case 5:
                    antrian.lihatAkhir();
                    break;
                case 6:
                    System.out.println(antrian.isEmpty() ? "Antrian
kosong." : "Antrian tidak kosong.");
                    System.out.println(antrian.isFull() ? "Antrian penuh."
: "Antrian belum penuh.");
                    break;
                case 7:
                    antrian.kosongkanAntrian();
                    break;
                case 8:
                    System.out.println("Jumlah antrian: " +
antrian.getJumlahAntrian());
                    break;
                case 9:
                    System.out.println("Jumlah sudah KRS: " +
antrian.getJumlahSudahKRS());
                    break;
                case 10:
                    System.out.println("Jumlah belum KRS: " +
antrian.getJumlahBelumKRS());
                    break;
                case 0:
                    System.out.println("Terima kasih!");
                    break;
                default:
                    System.out.println("Pilihan tidak valid.");
            }
        } while (pilihan != 0);
        sc.close();
    }
}
```


Output

```
=== MENU ANTRIAN KRS ===
1. Tambah Mahasiswa
2. Proses 2 Mahasiswa (KRS)
3. Lihat 2 Terdepan
4. Lihat Semua Antrian
5. Lihat Antrian Belakang
6. Cek Antrian Kosong / Penuh
7. Kosongkan Antrian
8. Cetak Jumlah Antrian
9. Cetak Jumlah Sudah KRS
10. Cetak Jumlah Belum KRS
0. Keluar
Pilih: 1
NIM: 123
Nama: Aldi
Prodi: TI
Kelas: 1A
Mahasiswa berhasil ditambahkan ke antrian.

=== MENU ANTRIAN KRS ===
1. Tambah Mahasiswa
2. Proses 2 Mahasiswa (KRS)
3. Lihat 2 Terdepan
4. Lihat Semua Antrian
5. Lihat Antrian Belakang
6. Cek Antrian Kosong / Penuh
7. Kosongkan Antrian
8. Cetak Jumlah Antrian
9. Cetak Jumlah Sudah KRS
10. Cetak Jumlah Belum KRS
0. Keluar
Pilih: 1
NIM: 124
Nama: Bayu
Prodi: TI
Kelas: 1A
Mahasiswa berhasil ditambahkan ke antrian.

=== MENU ANTRIAN KRS ===
1. Tambah Mahasiswa
2. Proses 2 Mahasiswa (KRS)
3. Lihat 2 Terdepan
4. Lihat Semua Antrian
5. Lihat Antrian Belakang
6. Cek Antrian Kosong / Penuh
7. Kosongkan Antrian
8. Cetak Jumlah Antrian
9. Cetak Jumlah Sudah KRS
10. Cetak Jumlah Belum KRS
0. Keluar
Pilih: 1
NIM: 125
Nama: Intan
Prodi: TI
Kelas: 1G
Mahasiswa berhasil ditambahkan ke antrian.
```

```
=== MENU ANTRIAN KRS ===
1. Tambah Mahasiswa
2. Proses 2 Mahasiswa (KRS)
3. Lihat 2 Terdepan
4. Lihat Semua Antrian
5. Lihat Antrian Belakang
6. Cek Antrian Kosong / Penuh
7. Kosongkan Antrian
8. Cetak Jumlah Antrian
9. Cetak Jumlah Sudah KRS
10. Cetak Jumlah Belum KRS
0. Keluar
Pilih: 1
NIM: 126
Nama: Ariq
Prodi: TI
Kelas: 1B
Mahasiswa berhasil ditambahkan ke antrian.

=== MENU ANTRIAN KRS ===
1. Tambah Mahasiswa
2. Proses 2 Mahasiswa (KRS)
3. Lihat 2 Terdepan
4. Lihat Semua Antrian
5. Lihat Antrian Belakang
6. Cek Antrian Kosong / Penuh
7. Kosongkan Antrian
8. Cetak Jumlah Antrian
9. Cetak Jumlah Sudah KRS
10. Cetak Jumlah Belum KRS
0. Keluar
Pilih: 3
2 mahasiswa terdepan:
123 - Aldi - TI - 1A
124 - Bayu - TI - 1A

=== MENU ANTRIAN KRS ===
1. Tambah Mahasiswa
2. Proses 2 Mahasiswa (KRS)
3. Lihat 2 Terdepan
4. Lihat Semua Antrian
5. Lihat Antrian Belakang
6. Cek Antrian Kosong / Penuh
7. Kosongkan Antrian
8. Cetak Jumlah Antrian
9. Cetak Jumlah Sudah KRS
10. Cetak Jumlah Belum KRS
0. Keluar
Pilih: 4
Daftar Mahasiswa dalam Antrian:
123 - Aldi - TI - 1A
124 - Bayu - TI - 1A
125 - Intan - TI - 1G
126 - Ariq - TI - 1B
```

```

=== MENU ANTRIAN KRS ===
1. Tambah Mahasiswa
2. Proses 2 Mahasiswa (KRS)
3. Lihat 2 Terdepan
4. Lihat Semua Antrian
5. Lihat Antrian Belakang
6. Cek Antrian Kosong / Penuh
7. Kosongkan Antrian
8. Cetak Jumlah Antrian
9. Cetak Jumlah Sudah KRS
10. Cetak Jumlah Belum KRS
0. Keluar
Pilih: 5
Mahasiswa paling belakang: 126 - Ariq - TI - 1B

=== MENU ANTRIAN KRS ===
1. Tambah Mahasiswa
2. Proses 2 Mahasiswa (KRS)
3. Lihat 2 Terdepan
4. Lihat Semua Antrian
5. Lihat Antrian Belakang
6. Cek Antrian Kosong / Penuh
7. Kosongkan Antrian
8. Cetak Jumlah Antrian
9. Cetak Jumlah Sudah KRS
10. Cetak Jumlah Belum KRS
0. Keluar
Pilih: 6
Antrian tidak kosong.
Antrian belum penuh.

=== MENU ANTRIAN KRS ===
1. Tambah Mahasiswa
2. Proses 2 Mahasiswa (KRS)
3. Lihat 2 Terdepan
4. Lihat Semua Antrian
5. Lihat Antrian Belakang
6. Cek Antrian Kosong / Penuh
7. Kosongkan Antrian
8. Cetak Jumlah Antrian
9. Cetak Jumlah Sudah KRS
10. Cetak Jumlah Belum KRS
0. Keluar
Pilih: 8
Jumlah antrian: 4

=== MENU ANTRIAN KRS ===
1. Tambah Mahasiswa
2. Proses 2 Mahasiswa (KRS)
3. Lihat 2 Terdepan
4. Lihat Semua Antrian
5. Lihat Antrian Belakang
6. Cek Antrian Kosong / Penuh
7. Kosongkan Antrian
8. Cetak Jumlah Antrian
9. Cetak Jumlah Sudah KRS
10. Cetak Jumlah Belum KRS
0. Keluar
Pilih: 2
Mahasiswa diproses: 123 - Aldi - TI - 1A [Sudah KRS]
Mahasiswa diproses: 124 - Bayu - TI - 1A [Sudah KRS]

```

```

=== MENU ANTRIAN KRS ===
1. Tambah Mahasiswa
2. Proses 2 Mahasiswa (KRS)
3. Lihat 2 Terdepan
4. Lihat Semua Antrian
5. Lihat Antrian Belakang
6. Cek Antrian Kosong / Penuh
7. Kosongkan Antrian
8. Cetak Jumlah Antrian
9. Cetak Jumlah Sudah KRS
10. Cetak Jumlah Belum KRS
0. Keluar
Pilih: 9
Jumlah sudah KRS: 2

=== MENU ANTRIAN KRS ===
1. Tambah Mahasiswa
2. Proses 2 Mahasiswa (KRS)
3. Lihat 2 Terdepan
4. Lihat Semua Antrian
5. Lihat Antrian Belakang
6. Cek Antrian Kosong / Penuh
7. Kosongkan Antrian
8. Cetak Jumlah Antrian
9. Cetak Jumlah Sudah KRS
10. Cetak Jumlah Belum KRS
0. Keluar
Pilih: 10
Jumlah belum KRS: 2

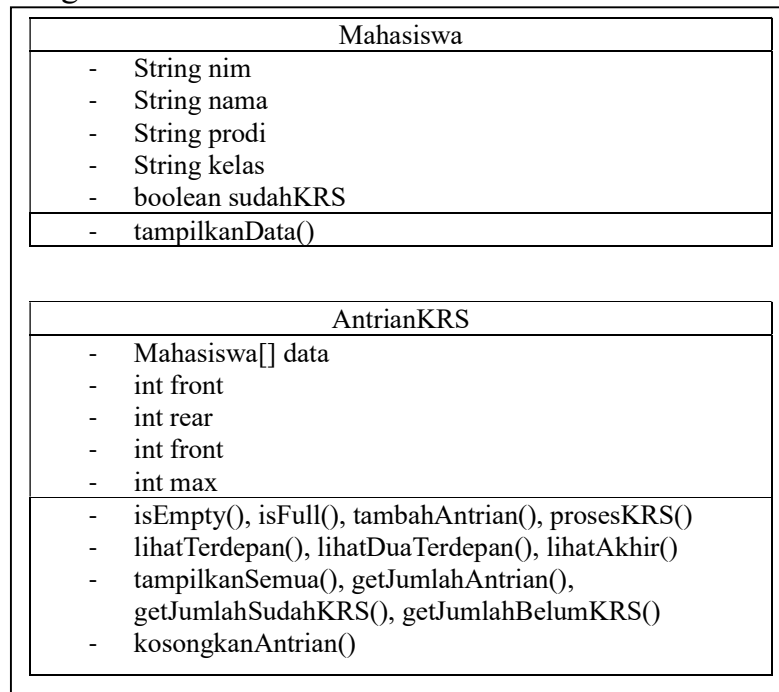
=== MENU ANTRIAN KRS ===
1. Tambah Mahasiswa
2. Proses 2 Mahasiswa (KRS)
3. Lihat 2 Terdepan
4. Lihat Semua Antrian
5. Lihat Antrian Belakang
6. Cek Antrian Kosong / Penuh
7. Kosongkan Antrian
8. Cetak Jumlah Antrian
9. Cetak Jumlah Sudah KRS
10. Cetak Jumlah Belum KRS
0. Keluar
Pilih: 7
Antrian dikosongkan.

=== MENU ANTRIAN KRS ===
1. Tambah Mahasiswa
2. Proses 2 Mahasiswa (KRS)
3. Lihat 2 Terdepan
4. Lihat Semua Antrian
5. Lihat Antrian Belakang
6. Cek Antrian Kosong / Penuh
7. Kosongkan Antrian
8. Cetak Jumlah Antrian
9. Cetak Jumlah Sudah KRS
10. Cetak Jumlah Belum KRS
0. Keluar
Pilih: 7
Antrian dikosongkan.

=== MENU ANTRIAN KRS ===
1. Tambah Mahasiswa
2. Proses 2 Mahasiswa (KRS)
3. Lihat 2 Terdepan
4. Lihat Semua Antrian
5. Lihat Antrian Belakang
6. Cek Antrian Kosong / Penuh
7. Kosongkan Antrian
8. Cetak Jumlah Antrian
9. Cetak Jumlah Sudah KRS
10. Cetak Jumlah Belum KRS
0. Keluar
Pilih: 6
Antrian kosong.
Antrian belum penuh.

```

Diagram Class



Link Github

<https://github.com/baynobu/ALSD/tree/c5cbfff0e271158a4b2c7a488cbf240a104f98f9/Pertemuan%2011>