

JOBSHEET 10
ALGORITMA STRUKTUR
DATA



Burhnauddin ihsan
244107020189
TI 1E/06

PROGRAM STUDI D-IV TEKNIK INFORMATIKA JURUSAN
TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2025

PERCOBAAN 1

1. Kode program

Queue

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

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

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

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

    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.println(data[i] + " ");
                i = (i+1) % max;
            }
            System.out.println(data[i] + " ");
            System.out.println("Jumlah elemen = " + size);
        }
    }

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

    void enqueue (int num) {
        if (isFull()) {
            return;
        } else {
            if (isEmpty()) {
                front = rear = 0;
            }
        }
    }
}
```

```

    } else {
        if (rear == max - 1) {
            rear = 0;
        } else {
            rear++;
        }
    }
    data[rear] = num;
    size++;
}

int dequeue () {
    int num = 0;
    if (isEmpty()) {
        System.out.println("queue masih kosong");
    } else {
        num = data[front];
        size--;
        if (isEmpty()) {
            front = rear = -1;
        } else {
            if (front == max - 1) {
                front = 0;
            } else {
                front++;
            }
        }
    }
    return num;
}
}

```

QueueMain

```

import java.util.Scanner;

public class QueueMain {
    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("masukkan pilihan : ");
            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;
    }
    } while (pilih >= 1 && pilih <= 5);

}

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("-----");
}

}

```

2. hasil dari kode program

```

Masukkan kapasitas queue : 4
Masukkan operasi yang diinginkan
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
masukkan pilihan : 1
Masukkan data baru : 15
Masukkan operasi yang diinginkan
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
masukkan pilihan : 1
Masukkan data baru : 31
Masukkan operasi yang diinginkan
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
masukkan pilihan : 4
Elemen terdepan : 15
Masukkan operasi yang diinginkan
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----

```

PERTANYAAN

1. karena pada saat inisiasi awal, data queue masih kosong, sehingga ditandai dengan front dan rearnya yang bernilai -1 (supaya tidak menandakan suatu index)
2. kondisi tersebut untuk handle ketika data paling belakang sudah penuh, data yang baru akan ditaruh pada index 0 (mengubah nilai front menjadi 0 terlebih dahulu)
3. kondisi tersebut untuk handle saat elemen di ujung dikeluarkan (dequeue), posisi front harus berputar kembali ke awal array (indeks 0).
4. karena data pada index ke-0 belum tentu ada isinya, data dimulai pada index front dan data yang paling belakang ada pada index rear
5. untuk handle data pada index sebelum rear yang melebihi max, ketika melebihi max maka index otomatis mulai dari 0 karena ada operasi modulus tsb.
6. potongan kode program yang merupakan queue overflow

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

7. Modifikasi program

```
if (isFull()) {  
    return;
```

PERCOBAAN 2

1. kode program

mahasiswa

```
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);  
    }  
}
```

AntrianLayanan

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

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

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

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

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

    void tambahAntrian (Mahasiswa mhs) {
        if (isFull()) {
            System.out.println("Antrian penuh, tidak dapat menambah
mahasiswa");
            return;
        }
        rear = (rear + 1) % max;
        data[rear] = mhs;
        size++;
        System.out.println(mhs.nama + " berhasil masuk ke
antrian");
    }

    Mahasiswa layaniMahasiswa() {
        if (isEmpty()) {
            System.out.println("antrian kosong");
            return null;
        }

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

```

void lihatTerdepan() {
    if (isEmpty()) {
        System.out.println("antrian kosong");
    } else {
        System.out.println("Mahasiswa terdepan : ");
        System.out.println("NIM - NAMA - PRODI - KELAS");
        data[front].tampilkanData();
    }
}

void lihatAkhir() {
    if (isEmpty()) {
        System.out.println("antrian kosong");
    } else {
        System.out.println("Mahasiswa terakhir : ");
        System.out.println("NIM - NAMA - PRODI - KELAS");
        data[rear].tampilkanData();
    }
}

void tampilkanSemua() {
    if (isEmpty()) {
        System.out.println("Antrian Kosong");
        return;
    }
    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.print((i+1) + ". ");
        data[index].tampilkanData();
    }
}

int getJumlahAntrian() {
    return size;
}
}

```

TampilanSiakadAkademik

```

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 pilihan;

        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("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.println("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 0:
            System.out.println("Terima kasih.");
            break;
        default:
            System.out.println("Pilihan tidak valid.");
    }
} while (pilihan != 0);

    sc.close();
}
}

```


2. hasil dari kode program

```
=== 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 : 134
Nama : burhanuddin
Prodi : TI
Kelas : 1e
burhanuddin 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 : 213
Nama : ihsan
Prodi : TI
Kelas : 1e
ihsan 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: 2
Melayani mahasiswa:
134 - burhanuddin - TI - 1e
```

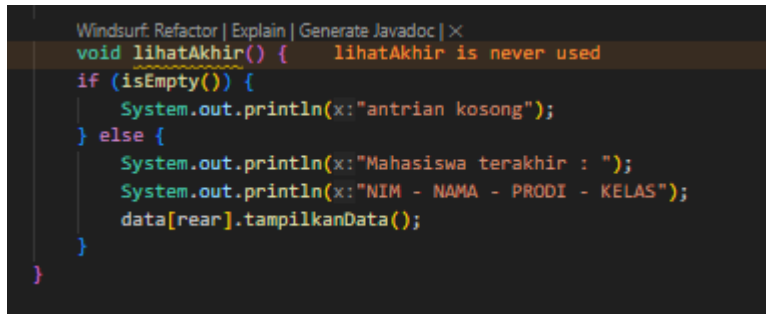
```
=== Menu Antrian Layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
2. Layani Mahasiswa
3. Lihat Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
0. Keluar
Pilih menu: 4
Daftar Mahasiswa dalam Antrian
NIM - NAMA - PRODI - KELAS
1. 213 - ihsan - TI - 1e

=== 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.
PS C:\Users\Lenovo\OneDrive\file\Praktikum-ASD>
```

PERTANYAAN

1. modifikasi program



```
Windsurf: Refactor | Explain | Generate Javadoc | X
void lihatAkhir() {      lihatAkhir is never used
    if (isEmpty()) {
        System.out.println(x:"antrian kosong");
    } else {
        System.out.println(x:"Mahasiswa terakhir : ");
        System.out.println(x:"NIM - NAMA - PRODI - KELAS");
        data[rear].tampilkanData();
    }
}
```

TUGAS

1. kode program

```
class AntrianKrs {

    Mahasiswa[] data;
    int front, rear, size, max, slot, mhsDilayani;

    AntrianKrs () {
        max = 10;
        this.data = new Mahasiswa[max];
        front = 0;
        rear = -1;
        size = 0;
        slot = 2;
        mhsDilayani = 0;
    }

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

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

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

    void tambahAntrian (Mahasiswa mhs) {
        if (isFull()) {
            System.out.println("Antrian penuh, tidak dapat menambah
mahasiswa");
            return;
        }

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

```

    }

    }

    void tambahAntrian (Mahasiswa mhs) {
        if (isFull()) {
            System.out.println("Antrian penuh, tidak dapat menambah
mahasiswa");
            return;
        }
        rear = (rear + 1) % max;
        data[rear] = mhs;
        size++;
        System.out.println(mhs.nama + " berhasil masuk ke
antrian");
    }

    Mahasiswa[] memanggilAntrian() {
        if (isEmpty()) {
            System.out.println("antrian kosong");
            return null;
        }

        Mahasiswa mhs[] = new Mahasiswa[2];
        slot = (this.size == 1) ? 1 : 2;

        for (int i = 0; i < slot; i++) {
            mhs[i] = data[front];
            front = (front+1) % max;
            size--;
            mhsDilayani++;
        }
        return mhs;
    }

    void tampilkanSemua() {
        if (isEmpty()) {
            System.out.println("Antrian Kosong");
            return;
        }
        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.print((i+1) + ". ");
            data[index].tampilkanData();
        }
    }

    void lihatDuaTerdepan() {
        if (isEmpty()) {
            System.out.println("antrian kosong");
        } else {
            for (int i = 0; i < slot; i++) {
                int index = (front + i) % max;
                System.out.println("Mahasiswa terdepan " + (i+1) + " : " );
                System.out.println("NIM - NAMA - PRODI - KELAS");
                data[index].tampilkanData();
            }
        }
    }

    void lihatAkhir() {
        if (isEmpty()) {
            System.out.println("antrian kosong");
        }
    }

```

```

else {
    System.out.println("Mahasiswa terakhir : ");
    System.out.println("NIM - NAMA - PRODI - KELAS");
    data[rear].tampilkanData();
}

}

int getJumlahAntrian() {
    return size;
}

int getTotalProsesKRS() {
    return mhsDilayani + size;
}

int getTotalMhsBelumDilayani() {
    return 30 - getTotalProsesKRS();
}

}

```

KrsMain

```

import java.util.Scanner;

public class KrsMain {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        AntrianKrs antrian = new AntrianKrs();
        int pilihan;

        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. Jumlah Mahasiswa yang Sudah Diproses");
            System.out.println("7. Jumlah Mahasiswa yang Belum Diproses");
            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.memanggilAntrian();
                    if (dilayani != null) {
                        System.out.println("Melayani mahasiswa: ");
                        dilayani[0].tampilkanData();
                    }
            }
        } while (pilihan != 0);
    }
}

```

```

        if (dilayani[1] != null) {
            dilayani[1].tampilkanData();
        }
    }
    break;
case 3:
    antrian.lihatDuaTerdepan();
    break;
case 4:
    antrian.tampilkanSemua();
    break;
case 5:
    System.out.println("Jumlah dalam antrian: " +
antrian.getJumlahAntrian());
    break;
case 6:
    System.out.println("Mahasiswa yang sudah dalam proses
: " + antrian.getTotalProsesKRS());
    break;
case 7:
    System.out.println("Mahasiswa yang belum diproses : "
+ antrian.getTotalMhsBelumDilayani());
case 0:
    System.out.println("Terima kasih.");
    break;
default:
    System.out.println("Pilihan tidak valid.");
    }
} while (pilihan != 0);

sc.close();
}
}

```

2. hasil dari kode program

```
=== 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
6. Jumlah Mahasiswa yang Sudah Diproses
7. Jumlah Mahasiswa yang Belum Diproses
0. Keluar
Pilih menu: 1
NIM   : 231
Nama  : burhan
Prodi : TI
Kelas : 1e
burhan 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
6. Jumlah Mahasiswa yang Sudah Diproses
7. Jumlah Mahasiswa yang Belum Diproses
0. Keluar
Pilih menu: 1
NIM   : 231
Nama  : udin
Prodi : SIB
Kelas : 1e
udin 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
6. Jumlah Mahasiswa yang Sudah Diproses
7. Jumlah Mahasiswa yang Belum Diproses
0. Keluar
Pilih menu: 1
NIM   : 213
Nama  : ihsan
Prodi : TI
Kelas : 1c
ihsan 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
6. Jumlah Mahasiswa yang Sudah Diproses
7. Jumlah Mahasiswa yang Belum Diproses
0. Keluar
Pilih menu: 2
Melayani mahasiswa:
231 - burhan - TI - 1e
231 - udin - SIB - 1e
```

```
=== 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
6. Jumlah Mahasiswa yang Sudah Diproses
7. Jumlah Mahasiswa yang Belum Diproses
0. Keluar
Pilih menu: 3
Mahasiswa terdepan 1 :
NIM - NAMA - PRODI - KELAS
213 - ihsan - TI - 1c
Mahasiswa terdepan 2 :
NIM - NAMA - PRODI - KELAS
```

```
=== 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
6. Jumlah Mahasiswa yang Sudah Diproses
7. Jumlah Mahasiswa yang Belum Diproses
0. Keluar
Pilih menu:
1
NIM : 231
Nama : burhan
Prodi : TI
Kelas : 1e
burhan 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
6. Jumlah Mahasiswa yang Sudah Diproses
7. Jumlah Mahasiswa yang Belum Diproses
0. Keluar
Pilih menu: 1
NIM : 231
Nama : udin
Prodi : SIB
Kelas : 1e
udin 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
6. Jumlah Mahasiswa yang Sudah Diproses
7. Jumlah Mahasiswa yang Belum Diproses
0. Keluar
Pilih menu:
1
NIM : 213
Nama : ihsan
Prodi : TI
Kelas : 1c
ihsan 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
6. Jumlah Mahasiswa yang Sudah Diproses
7. Jumlah Mahasiswa yang Belum Diproses
0. Keluar
Pilih menu: 4
Daftar Mahasiswa dalam Antrian
NIM - NAMA - PRODI - KELAS
1. 231 - burhan - TI - 1e
2. 231 - udin - SIB - 1e
3. 213 - ihsan - TI - 1c

=== 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
6. Jumlah Mahasiswa yang Sudah Diproses
7. Jumlah Mahasiswa yang Belum Diproses
0. Keluar
Pilih menu: 5
Jumlah dalam antrian: 3

=== 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
6. Jumlah Mahasiswa yang Sudah Diproses
7. Jumlah Mahasiswa yang Belum Diproses
0. Keluar
Pilih menu: 6
Mahasiswa yang sudah dalam proses : 3

=== 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
6. Jumlah Mahasiswa yang Sudah Diproses
7. Jumlah Mahasiswa yang Belum Diproses
0. Keluar
Pilih menu: 7
Mahasiswa yang belum diproses : 27
Terima kasih.
```