

Laporan Hasil Praktikum Algoritma Struktur Data

Jobsheet 10



Febryan Akhmad Taajuddin

244107020180

Kelas 1E

Program Studi Teknik Informatika

Jurusan Teknologi Informasi

Politeknik Negeri Malang

2025

Percobaan 1

1. Tambahkan method sequentialSearching pada class MahasiswaBerprestasi10

```
public class Queue10 {  
    int[] data;  
    int front;  
    int rear;  
    int size;  
    int max;  
  
    public Queue10 (int n) {  
        max = n;  
        data = new int [max];  
        size = 0;  
        front = rear = -1;  
    }  
}
```

2. Buat method isEmpty

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

3. Buat method IsFull

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

4. Buat method peek

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

5. Buat method print

```
public 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 elemen = " + size);
    }
}
```

6. Method clear

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

7. Method Enqueue

```
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++;
    }
}
```

8. Method dequeue

```
public 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;
}
```

9. Buat class QueueMain10

```
import java.util.Scanner;
public class QueueMain10 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Masukkan kapasitas queue: ");
        int n = sc.nextInt();
        Queue10 Q = new Queue10(n);
        int pilih;

        do {
            menu();
            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 == 2 || pilih == 3 || pilih == 4 || pilih == 5);
        sc.close();
    }

    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("-----");
    }
}
```

10. Run Kode Program

```
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
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Clear
-----
```

Pertanyaan percobaan 1

1. Pada konstruktor, mengapa nilai awal atribut front dan rear bernilai -1, sementara atribut size bernilai 0?
2. Pada method Enqueue, jelaskan maksud dan kegunaan dari potongan kode berikut!

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

3. Pada method Dequeue, jelaskan maksud dan kegunaan dari potongan kode berikut!

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

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

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

6. Tunjukkan potongan kode program yang merupakan queue overflow!
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!

Jawaban percobaan 1

1. Atribut front dan rear -1 karena untuk mengetahui array tersebut kosong atau tidak, atribut size = 0 karena dihitung sesuai dengan jumlah array yang ditambahkan
2. Jika rear sudah berada di indeks terakhir maka pindah array ke indeks 0 atau indeks pertama
3. Jika front sudah berada di indeks terakhir maka pindah ke array indeks 0 untuk mengisi elemen yang kosong selama masih ada ruang kosong di array
4. Karena data pertama tidak selalu di indeks ke 0 karena yang ditempati data pertama bisa di sembarang indeks, maka dari itu print harus dilakukan dari i = front
5. Untuk memindahkan front ke indeks berikutnya, dan kembali ke indeks 0 jika sudah sampai akhir array
6. Kode program

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

7. Modifikasi kode program

```
public void Enqueue(int dt){  
    if (IsFull()) {  
        System.out.println("Queue sudah penuh! Program dihentikan");  
        System.exit(1);  
        return;  
    }  
    if (IsEmpty()) {  
        System.out.println("Queue masih kosong! Program dihentikan");  
        System.exit(1);  
        return -1;  
    }  
}
```

- Run kode program

```
Masukkan kapasitas queue: 1  
Masukkan operasi yang diinginkan:  
1. Enqueue  
2. Dequeue  
3. Print  
4. Peek  
5. Clear  
-----  
1  
Masukkan data baru: 100  
Masukkan operasi yang diinginkan:  
1. Enqueue  
2. Dequeue  
3. Print  
4. Peek  
5. Clear  
-----  
1  
Masukkan data baru: 23  
Queue sudah penuh! Program dihentikan
```

```
Masukkan kapasitas queue: 0  
Masukkan operasi yang diinginkan:  
1. Enqueue  
2. Dequeue  
3. Print  
4. Peek  
5. Clear  
-----  
2  
Queue masih kosong! Program dihentikan
```

Percobaan 2

1. Class Mahasiswa10

```
public class Mahasiswa10 {
    String nim;
    String nama;
    String prodi;
    String kelas;

    public Mahasiswa10(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);
    }
}
```

2. Class AntrianLayanan

```
public class AntrianLayanan {
    Mahasiswa10[] data;
    int front;
    int rear;
    int size;
    int max;

    public AntrianLayanan (int max) {
        this.max = max;
        this.data = new Mahasiswa10[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("Antrian kosong.");
        }else{
            System.out.print("Mahasiswa terdepan: ");
            System.out.println("NIM - NAMA - PRODI - KELAS");
            data[front].tampilkanData();
        }
    }
}
```

```

public 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();
    }
}

public int getJumlahAntrian(){
    return size;
}

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 tambahAntrian(Mahasiswa0 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.");
}

public Mahasiswa0 layaniMahasiswa(){
    if (IsEmpty()) {
        System.out.println("Antrian kosong.");
        return null;
    }
    Mahasiswa0 mhs = data[front];
    front = (front + 1) % max;
    size--;
    return mhs;
}
}

```


3. Class LayananaAkademikSIKAD10

```
import java.util.Scanner;
public class LayananaAkademikSIKAD10 {
    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("Pilihan 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();
                    Mahasiswa10 mhs = new Mahasiswa10(nim, nama,
prodi, kelas);
                    antrian.tambahAntrian(mhs);
                    break;
```

```

        case 2:
            Mahasiswa10 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 0:
            System.out.println("Terima kasih.");
            break;
        default:
            System.out.println("Pilihan tidak valid.");
    }
} while (pilihan != 0);
sc.close();
}
}

```

4. Run 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
Pilihan 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
Pilihan 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
Pilihan 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
Pilihan 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
Pilihan 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
Pilihan 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
Pilihan menu: 0
Terima kasih.

```

Pertanyaan percobaan 2

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!

Jawaban percobaan 2

1. Kode Program
 - Class LayananAkademikSIKAD10

```

System.out.println("6. Cek Antrian paling belakang");
case 6:
    antrian.LihatAkhir();

```

- Class Antrian Layanan

```

public void LihatAkhir(){
    if (IsEmpty()) {
        System.out.println("Antrian kosong.");
    }else{
        System.out.print("Mahasiswa terdepan: ");
        System.out.println("NIM - NAMA - PRODI - KELAS");
        data[rear].tampilkanData();
    }
}

```

- Run 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. Cek Antrian paling belakang
0. Keluar
Pilihan menu: 1
NIM :1
NAMA :roni
Prodi :ti
Kelas :1e
roni 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. Cek Antrian paling belakang
0. Keluar
Pilihan menu: 1
NIM :2
NAMA :riko
Prodi :ti
Kelas :1e
riko 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. Cek Antrian paling belakang
0. Keluar
Pilihan menu: 1
NIM :3
NAMA :rusdi
Prodi :ti
Kelas :1e
rusdi 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. Cek Antrian paling belakang
0. Keluar
Pilihan menu:
3
Mahasiswa terdepan: NIM - NAMA - PRODI - KELAS
1 - roni - 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
6. Cek Antrian paling belakang
0. Keluar
Pilihan menu: 6
Mahasiswa terdepan: NIM - NAMA - PRODI - KELAS
3 - rusdi - ti - 1e
Terima kasih.

```

```

=== Menu Antrian Layanan Akademik ===
1. Tambah Mahasiswa ke Antrian
3. Lihat Mahasiswa Terdepan
4. Lihat Semua Antrian
5. Jumlah Mahasiswa dalam Antrian
6. Cek Antrian paling belakang
0. Keluar
Pilihan menu: 0
Terima kasih.

```

Latihan Praktikum

1. Berikut kode programnya

- Class MahasiswaKRS10

```
public class MahasiswaKRS10 {
    String nim;
    String nama;
    String jurusan;

    public MahasiswaKRS10(String nim, String nama,
String jurusan){
        this.nim = nim;
        this.nama = nama;
        this.jurusan = jurusan;
    }

    public void tampil(){
        System.out.println("NIM      :"+nim);
        System.out.println("Nama      :"+nama);
        System.out.println("Jurusan :"+jurusan);
    }
}
```

- Class KRS10

```
public class KRS10 {
    int max = 10;
    MahasiswaKRS10[] antrian = new MahasiswaKRS10[max];
    int front = -1, rear = -1;
    int totalDilayani = 0;

    boolean isEmpty() {
        return front == -1;
    }

    boolean isFull() {
        return (rear + 1) % max == front;
    }

    void Enqueue(MahasiswaKRS10 mhs) {
        if (isFull()) {
            System.out.println("Antrian penuh");
            return;
        }
        if (isEmpty()) {
            front = rear = 0;
        }else{
            rear = (rear + 1) % max;
        }
        antrian[rear] = mhs;
    }
}
```

```

void dequeue(){
    if (isEmpty()) {
        System.out.println("Antrian kosong");
        return;
    }
    for (int i = 0; i < 2; i++) {
        if (isEmpty()) break;

        System.out.println("Memproses Mahasiswa:");
        antrian[front].tampil();
        if (front == rear) {
            front = rear = -1;
        }else{
            front = (front + 1) % max;
        }
        totalDilayani++;
    }
}

void tampilAntrian() {
    if (isEmpty()) {
        System.out.println("Antrian kosong");
        return;
    }
    System.out.println("Daftar Antrian");
    int i = front;
    while (true) {
        System.out.println("- "+antrian[i].nama + "
("+ antrian[i].nim +")");
        if (i == rear) break;
        i = (i+1) % max;
    }
}

void tampilDuaTerdepan(){
    if (isEmpty()) {
        System.out.println("Antrian kosong");
        return;
    }
    System.out.println("Dua Mahasiswa tedepan");
    int i = front;
    for (int j = 0; j < 2 && i != (rear+1) % max;
j++) {
        antrian[i].tampil();
        i = (i+1) % max;
    }
}

```

```

void tampilPalingAkhir() {
    if (isEmpty()) {
        System.out.println("Antian kosong");
        return;
    }
    System.out.println("Mahasiswa terakhir dalam
antrian:");
    antrian[rear].tampil();
}

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

void cetakJumlahAntrian() {
    if (isEmpty()) {
        System.out.println("Jumlah antrian: 0");
        return;
    }
    int count = 0;
    int i = front;
    while (true) {
        count++;
        if (i == rear) break;
        i = (i + 1) % max;
    }
    System.out.println("Jumlah antrian: " + count);
}

void cetakJumlahDilayani() {
    System.out.println("Jumlah mahasiswa yang sudah
diproses KRS: " + totalDilayani);
}

void cetakBelumDilayani() {
    int sisa = 30 - totalDilayani;
    System.out.println("Mahasiswa yang belum
melakukan proses KRS: " + sisa);
}
}

```

- Class KRSMain10

```
import java.util.Scanner;
public class KRSMain10 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        KRS10 antrian = new KRS10();
        int pilihan;

        do {
            System.out.println("\n===== MENU
ANTRIAN KRS =====");
            System.out.println("1. Tambah Mahasiswa ke
Antrian");
            System.out.println("2. Proses 2 Mahasiswa
(KRS)");
            System.out.println("3. Tampilkan Semua
Antrian");
            System.out.println("4. Tampilkan 2
Mahasiswa Terdepan");
            System.out.println("5. Tampilkan Mahasiswa
Terakhir");
            System.out.println("6. Kosongkan Antrian");
            System.out.println("7. Cetak Jumlah
Antrian");
            System.out.println("8. Cetak Jumlah yang
Sudah Dilayani");
            System.out.println("9. Cetak Jumlah yang
Belum Dilayani");
            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("Jurusan : ");
                    String jurusan = sc.nextLine();
                    MahasiswaKRS10 m = new
MahasiswaKRS10(nim, nama, jurusan);
                    antrian.Enqueue(m);
                    break;
```



```

        case 2:
            antrian.dequeue();
            break;
        case 3:
            antrian.tampilAntrian();
            break;
        case 4:
            antrian.tampilDuaTerdepan();
            break;
        case 5:
            antrian.tampilPalingAkhir();
            break;
        case 6:
            antrian.kosongkanAntrian();
            break;
        case 7:
            antrian.cetakJumlahAntrian();
            break;
        case 8:
            antrian.cetakJumlahDilayani();
            break;
        case 9:
            antrian.cetakBelumDilayani();
            break;
        case 0:
            System.out.println("Terima kasih!");
            break;
        default:
            System.out.println("Menu tidak
tersedia.");
    }
    } while (pilihan != 0);
    sc.close();
}
}

```

- Output

```
===== MENU ANTRIAN KRS =====
1. Tambah Mahasiswa ke Antrian
2. Proses 2 Mahasiswa (KRS)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Kosongkan Antrian
7. Cetak Jumlah Antrian
8. Cetak Jumlah yang Sudah Dilayani
9. Cetak Jumlah yang Belum Dilayani
0. Keluar
Pilih menu: 1
NIM      : 1
Nama     : agus
Jurusan  : ti

===== MENU ANTRIAN KRS =====
1. Tambah Mahasiswa ke Antrian
2. Proses 2 Mahasiswa (KRS)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Kosongkan Antrian
7. Cetak Jumlah Antrian
8. Cetak Jumlah yang Sudah Dilayani
9. Cetak Jumlah yang Belum Dilayani
0. Keluar
Pilih menu: 1
NIM      : 2
Nama     : hardi
Jurusan  : ti
```

```
===== MENU ANTRIAN KRS =====
1. Tambah Mahasiswa ke Antrian
2. Proses 2 Mahasiswa (KRS)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Kosongkan Antrian
7. Cetak Jumlah Antrian
8. Cetak Jumlah yang Sudah Dilayani
9. Cetak Jumlah yang Belum Dilayani
0. Keluar
Pilih menu: 1
NIM      : 3
Nama     : wijaya
Jurusan  : ti

===== MENU ANTRIAN KRS =====
1. Tambah Mahasiswa ke Antrian
2. Proses 2 Mahasiswa (KRS)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Kosongkan Antrian
7. Cetak Jumlah Antrian
8. Cetak Jumlah yang Sudah Dilayani
9. Cetak Jumlah yang Belum Dilayani
0. Keluar
Pilih menu: 2
Memproses Mahasiswa:
NIM      :1
Nama     :agus
Jurusan  :ti
Memproses Mahasiswa:
NIM      :2
Nama     :hardi
Jurusan  :ti
```

```
===== MENU ANTRIAN KRS =====
1. Tambah Mahasiswa ke Antrian
2. Proses 2 Mahasiswa (KRS)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Kosongkan Antrian
7. Cetak Jumlah Antrian
8. Cetak Jumlah yang Sudah Dilayani
9. Cetak Jumlah yang Belum Dilayani
0. Keluar
Pilih menu: 3
Daftar Antrian
- wijaya (3)
```

```
===== MENU ANTRIAN KRS =====
1. Tambah Mahasiswa ke Antrian
2. Proses 2 Mahasiswa (KRS)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Kosongkan Antrian
7. Cetak Jumlah Antrian
8. Cetak Jumlah yang Sudah Dilayani
9. Cetak Jumlah yang Belum Dilayani
0. Keluar
Pilih menu: 4
Dua Mahasiswa terdepan
NIM      :3
Nama     :wijaya
Jurusan  :ti
```

```
===== MENU ANTRIAN KRS =====
1. Tambah Mahasiswa ke Antrian
2. Proses 2 Mahasiswa (KRS)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Kosongkan Antrian
7. Cetak Jumlah Antrian
8. Cetak Jumlah yang Sudah Dilayani
9. Cetak Jumlah yang Belum Dilayani
0. Keluar
Pilih menu: 5
Mahasiswa terakhir dalam antrian:
NIM      :3
Nama     :wijaya
Jurusan  :ti
```

```
===== MENU ANTRIAN KRS =====
1. Tambah Mahasiswa ke Antrian
2. Proses 2 Mahasiswa (KRS)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Kosongkan Antrian
7. Cetak Jumlah Antrian
8. Cetak Jumlah yang Sudah Dilayani
9. Cetak Jumlah yang Belum Dilayani
0. Keluar
Pilih menu: 6
Antrian dikosongkan.
```

```
===== MENU ANTRIAN KRS =====
1. Tambah Mahasiswa ke Antrian
2. Proses 2 Mahasiswa (KRS)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Kosongkan Antrian
7. Cetak Jumlah Antrian
8. Cetak Jumlah yang Sudah Dilayani
9. Cetak Jumlah yang Belum Dilayani
0. Keluar
Pilih menu: 7
Jumlah antrian: 0
```

```
===== MENU ANTRIAN KRS =====
1. Tambah Mahasiswa ke Antrian
2. Proses 2 Mahasiswa (KRS)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Kosongkan Antrian
7. Cetak Jumlah Antrian
8. Cetak Jumlah yang Sudah Dilayani
9. Cetak Jumlah yang Belum Dilayani
0. Keluar
Pilih menu: 8
Jumlah mahasiswa yang sudah diproses KRS: 2
```

```

===== MENU ANTRIAN KRS =====
1. Tambah Mahasiswa ke Antrian
2. Proses 2 Mahasiswa (KRS)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Kosongkan Antrian
7. Cetak Jumlah Antrian
8. Cetak Jumlah yang Sudah Dilayani
9. Cetak Jumlah yang Belum Dilayani
0. Keluar
Pilih menu: 9
Mahasiswa yang belum melakukan proses KRS: 28

===== MENU ANTRIAN KRS =====
1. Tambah Mahasiswa ke Antrian
2. Proses 2 Mahasiswa (KRS)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Kosongkan Antrian
7. Cetak Jumlah Antrian
8. Cetak Jumlah yang Sudah Dilayani
9. Cetak Jumlah yang Belum Dilayani
0. Keluar
Pilih menu: 0
Terima kasih!

```

- Class Diagram

