LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR

QUEUE

ABIM MUSTAWA

244107020078

KELAS TI-1B

PRODI D-IV TEKNIK

INFORMATIKA JURUSAN

TEKNOLOGI INFORMASI

POLITEKNIK NEGERI

MALANG 2025

1. Percobaan Praktikum

1.1 Praktikum 1 – Operasi Dasar Queue

Pada percobaan ini, kita akan menerapkan operasi dasar pada algoritma Queue. Perhatikan Diagram Class Queue berikut ini:

Queue

- data: int[]
- front: int
- rear: int
- size: int
- max: int
- Queue(n: int)
- isFull(): boolean
- isEmpty(): boolean
- enqueue(dt: int): void
- dequeue(): int
- peek: void
- print(): void
- clear(): void

Berdasarkan diagram class tersebut, akan dibuat program class Queue dalam Java.

Langkah-langkah Percobaan

1. Buat folder baru bernama P1Jobsheet10 di dalam repository Praktikum ASD, kemudian buat class baru dengan nama Queue.
2. Tambahkan atribut-atribut Queue sesuai diagram class, kemudian tambahkan pula konstruktornya seperti berikut ini :

```
int[] data;
int front, rear, size, max;

public Queue01(int n) {
    max = n;
    data = new int[max];
    front = rear = -1;
}
```

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("Elemen terdepan: " + data[front]);
    }else {
        System.out.println(x:"Queue 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) {
            System.out.println(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. Selanjutnya, buat class baru dengan nama QueueMain tetap pada package

Praktikum1. Buat method menu bertipe void untuk memilih menu program pada saat dijalankan.

```
public static void menu(){
    System.out.println(x:"Masukkan operasi yang diinginkan:");
    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. Celar");
    System.out.println(x:"--------------------------------");
    System.out.print(s:"Pilih menu: ");
}
```

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

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

13. Lakukan instansiasi objek Queue dengan nama Q dengan mengirimkan parameter n sebagai kapasitas elemen queue

```
Queue01 Q = new Queue01(n);
```

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

```
int pilih;
```

15. 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:
            break;
    }
} while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5 );
```

16. Compile dan jalankan class QueueMain, kemudian amati hasilnya.

Verifikasi Hasil Percobaan

```
Masukkan kapasitas queue: 4
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Celar
--------------------------------
Pilih menu: 1
Masukkan data baru: 15
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Celar
--------------------------------
Pilih menu: 1
Masukkan data baru: 31
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Celar
--------------------------------
Pilih menu: 4
Elemen terdepan: 15
Masukkan operasi yang diinginkan:
1. Enqueue
2. Dequeue
3. Print
4. Peek
5. Celar
--------------------------------
Pilih menu:
```

Pertanyaan

1. Pada konstruktor, mengapa nilai awal atribut front dan rear bernilai -1, sementara atribut size bernilai 0?
   Jawab:
   Fornt = rear = -1 menandakan queue kosong atau belum ada elemen, size = 0 karena belum ada yang dimasukkan, variabel size akan bertambah setiap kali elemen baru ditambahkan dan akan berkurang setiap kali elemen dikeluarkan.

2. Pada method Enqueue, jelaskan maksud dan kegunaan dari potongan kode berikut!
   Jawab:
   ```
   if (rear == max - 1) {
      rear = 0;
   }
   ```
   Maksud dari kode diatas yaitu ketika pointer rear mencapai akhir array (max-1), rear akan kembali ke index 0.

3. Pada method Dequeue, jelaskan maksud dan kegunaan dari potongan kode berikut!
   Jawab:
   ```
   if (front == max - 1) {
      front = 0;
   }
   ```
   Maksud dari kode diatas yaitu ketika pointer front mencapai akhir array, front akan kembali ke index 0 untuk memproses eleman berikutnya.

4. Pada method print, mengapa pada proses perulangan variabel i tidak dimulai dari 0 (int i=0), melainkan int i=front?
   Jawab:
   Karena elemen selalu diproses dari front (depan) ke belakang (rear), elemen terdepan tidak selalu dimulai dari index 0, sehingga elemen pertama mungkin berada di tengah atau akhir array.

5. Perhatikan kembali method print, jelaskan maksud dari potongan kode berikut!
   Jawab:
   ```
   i = (i + 1) % max;
   ```
   kode diatas digunakan untuk menggerakan index i, % max berfungsi untuk memastikan ketika i + 1 memcapai max, nilai 1 akan kembali ke 0.

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

```
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!

Jawab:

```
if (IsFull()) {
    System.out.println(x:"Queue sudah penuh");
    System.exit(status:0);
```

```
if (IsEmpty()) {
    System.out.println(x:"Queue masih kosong");
    System.exit(status:0);
```

1.2 Praktikum 2 – Antrian Layanan Akademik

Pada percobaan ini, kita akan membuat program yang mengilustrasikan Layanan pada Admin Akademik. Perhatikan Diagram Class berikut ini:

Mahasiswa

- nim:String
- nama: String
- prodi: String
- kelas: String
- Mahasiswa(nim: String, nama: String, prodi: String, kelas: String)
- void tampilkanData()

Langkah-langkah Percobaan

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. Dan tambahkan method tampilkanData berikut :

```java
public class Mahasiswa01 {

    String nim, nama, prodi, kelas;
    public Mahasiswa01(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. Karena pada Praktikum 1, data yang disimpan pada queue hanya berupa array bertipe integer, sedangkan pada Praktikum 2 data yang digunakan adalah object, maka perlu dilakukan modifikasi pada class AntrianLayanan tersebut.

```java
public class AntrianLayanan01 {
    Mahasiswa01[] data;
    int front, rear, size, max;

    public AntrianLayanan01(int max){
        this.max = max;
        this.data = new Mahasiswa01[max];
        this.front = 0;
        this.rear = -1;
        this.size = 0;
    }
```

4. Lakukan modifikasi pada class AntrianLayanan dengan mengubah tipe int[] data menjadi Mahasiswa[] data karena pada kasus ini data yang akan disimpan berupa object Mahasiswa. Modifikasi perlu dilakukan pada atribut, method Enqueue, dan method Dequeue.

```
public void tambahAntrian(Mahasiswa01 mhs) {
    if (IsFull()) {
        System.out.println(x:"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 Mahasiswa01 layaniMahasiswa(){
    if (IsEmpty()) {
        System.out.println(x:"Antrian kosong.");
        return null;
    }
    Mahasiswa01 mhs = data[front];
    front = (front + 1) % max;
    size--;
    return mhs;
}
```

5. Berikutnya method peek dan print yaitu untuk menampilkan data antrian layanan paling depan dan menampilkan semua data antrian layanan. Ditambahkan dengan method getJumlahAntrian yaitu menampilkan nilai size

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

public void tampilkanSemua(){
    if (IsEmpty()) {
        System.out.println(x:"Antrian kosng.");
        return;
    }
    System.out.println(x:"Daftar Mahasiswa dalam Antrian.");
    System.out.println(x:"NIM - NAMA - PRODI - KELAS");
    for (int i = 0; i < size; i++) {
        int index = (front + i) % max;
        System.out.println((i + 1) + ". ");
        data[index].tampilkanData();
    }
}

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

6. Selanjutnya, buat class baru dengan nama LayananAkademikSIAKAD tetap pada package yang sama. Buat fungsi main, deklarasikan Scanner dengan nama sc.
7. Kemudian lakukan instansiasi objek AntrianLayanan dengan nama antrian dan nilai parameternya adalah nilai maksimal antrian yang ditentukan (misal sama dengan 5).
8. Deklarasikan variabel dengan nama pilihan bertipe integer untuk menampung pilih menu dari pengguna.

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    AntrianLayanan01 antrian = new AntrianLayanan01(max:5);
    int pilihan;
```

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

```
do {
    System.out.println(x:"\n=== Menu Antrian Layanan Akademik ===");
    System.out.println(x:"1. Tambah Mahasiswa ke Antrian");
    System.out.println(x:"2. Layani Mahasiswa");
    System.out.println(x:"3. Lihat Mahasiswa Terdepan");
    System.out.println(x:"4. Lihat Semua Antrian");
    System.out.println(x:"5. Jumlah Mahasiswa dalam Antrian");
    System.out.println(x:"0. 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();
            Mahasiswa01 mhs = new Mahasiswa01(nim, nama, prodi, kelas);
            antrian.tambahAntrian(mhs);
            break;

        case 2:
            Mahasiswa01 dilayani = antrian.layaniMahasiswa();
            if (dilayani != null) {
                System.out.print(s:"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(x:"Terima kasih.");
                break;

            default:
                System.out.println(x:"Pilihan tidak valid.");
            }

    } while (pilihan != 0);
    sc.close();
```

10. Compile dan jalankan class LayananAkademikSIAKAD, kemudian amati hasilnya

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: 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.
```

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 LayananAkademikSIAKAD sehingga method LihatAkhir dapat dipanggil!

   Jawab:

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

```
System.out.println(x:"6. Lihat Antrian Paling Belakang");
```

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

2. Latihan Praktikum

2.1 Latihan 1

Buatlah program antrian untuk mengilustasikan antrian persetujuan Kartu Rencana Studi (KRS) Mahasiswa oleh Dosen Pembina Akademik (DPA). Ketika seorang mahasiswa akan mengantri, maka dia harus mendaftarkan datanya (data mahasiswa seperti pada praktikum 2). Gunakan class untuk antrian seperti pada Praktikum 1 dan 2, dengan method-method yang berfungsi :

- Cek antrian kosong, Cek antrian penuh, Mengosongkan antrian.
- Menambahkan antrian, Memanggil antrian untuk proses KRS – setiap 1x panggilan terdiri dari 2 mahasiswa (pada antrian no 1 dan 2)
- Menampilkan semua antrian, Menampilkan 2 antrian terdepan, Menampilkan antrian paling akhir.
- Cetak jumlah antrian, Cetak jumlah yang sudah melakukan proses KRS
- Jumlah antrian maximal 10, jumlah yang ditangani masing-masing DPA 30 mahasiswa, cetak jumlah mahasiswa yang belum melakukan proses KRS.

Gambarkan Diagram Class untuk antriannya. Implementasikan semua method menggunakan menu pilihan pada fungsi main.

Jawab:

```
=== SISTEM ANTRIAN KRS DPA ===
1. Tambah Mahasiswa
2. Proses KRS (2 mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Jumlah Mahasiswa dalam Antrian
7. Jumlah Mahasiswa Sudah Diproses
8. Jumlah Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih menu: 1
NIM : 112
Nama : Abim
Prodi : TI
Kelas : 1B
Abim berhasil ditambahkan ke antrian KRS.

=== SISTEM ANTRIAN KRS DPA ===
1. Tambah Mahasiswa
2. Proses KRS (2 mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Jumlah Mahasiswa dalam Antrian
7. Jumlah Mahasiswa Sudah Diproses
8. Jumlah Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih menu: 1
NIM : 113
Nama : Harisun
Prodi : TI
Kelas : 1A
Harisun berhasil ditambahkan ke antrian KRS.
```

```
=== SISTEM ANTRIAN KRS DPA ===
1. Tambah Mahasiswa
2. Proses KRS (2 mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Jumlah Mahasiswa dalam Antrian
7. Jumlah Mahasiswa Sudah Diproses
8. Jumlah Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih menu: 1
NIM : 114
Nama : Laily
Prodi : TI
Kelas : 1C
Laily berhasil ditambahkan ke antrian KRS.

=== SISTEM ANTRIAN KRS DPA ===
1. Tambah Mahasiswa
2. Proses KRS (2 mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Jumlah Mahasiswa dalam Antrian
7. Jumlah Mahasiswa Sudah Diproses
8. Jumlah Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih menu: 3
Daftar Antrian KRS:
No. NIM - NAMA - PRODI - KELAS
1. 112 - Abim - TI - 1B
2. 113 - Harisun - TI - 1A
3. 114 - Laily - TI - 1C
```

```
=== SISTEM ANTRIAN KRS DPA ===
1. Tambah Mahasiswa
2. Proses KRS (2 mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Jumlah Mahasiswa dalam Antrian
7. Jumlah Mahasiswa Sudah Diproses
8. Jumlah Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih menu: 4
2 Mahasiswa Terdepan:
1. 112 - Abim - TI - 1B
2. 113 - Harisun - TI - 1A

=== SISTEM ANTRIAN KRS DPA ===
1. Tambah Mahasiswa
2. Proses KRS (2 mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Jumlah Mahasiswa dalam Antrian
7. Jumlah Mahasiswa Sudah Diproses
8. Jumlah Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih menu: 5
Mahasiswa terakhir:
114 - Laily - TI - 1C
```

```
=== SISTEM ANTRIAN KRS DPA ===
1. Tambah Mahasiswa
2. Proses KRS (2 mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Jumlah Mahasiswa dalam Antrian
7. Jumlah Mahasiswa Sudah Diproses
8. Jumlah Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih menu: 6
Jumlah dalam antrian: 3

=== SISTEM ANTRIAN KRS DPA ===
1. Tambah Mahasiswa
2. Proses KRS (2 mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Jumlah Mahasiswa dalam Antrian
7. Jumlah Mahasiswa Sudah Diproses
8. Jumlah Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih menu: 7
Sudah diproses: 0
```

```
=== SISTEM ANTRIAN KRS DPA ===
1. Tambah Mahasiswa
2. Proses KRS (2 mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Jumlah Mahasiswa dalam Antrian
7. Jumlah Mahasiswa Sudah Diproses
8. Jumlah Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih menu: 2
Memproses KRS untuk 2 mahasiswa:
1. 112 - Abim - TI - 1B
2. 113 - Harisun - TI - 1A
Total mahasiswa diproses: 2

=== SISTEM ANTRIAN KRS DPA ===
1. Tambah Mahasiswa
2. Proses KRS (2 mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Jumlah Mahasiswa dalam Antrian
7. Jumlah Mahasiswa Sudah Diproses
8. Jumlah Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih menu: 3
Daftar Antrian KRS:
No. NIM - NAMA - PRODI - KELAS
1. 114 - Laily - TI - 1C
```

```
=== SISTEM ANTRIAN KRS DPA ===
1. Tambah Mahasiswa
2. Proses KRS (2 mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Jumlah Mahasiswa dalam Antrian
7. Jumlah Mahasiswa Sudah Diproses
8. Jumlah Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih menu: 9
Antrian berhasil dikosongkan

=== SISTEM ANTRIAN KRS DPA ===
1. Tambah Mahasiswa
2. Proses KRS (2 mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Mahasiswa Terdepan
5. Tampilkan Mahasiswa Terakhir
6. Jumlah Mahasiswa dalam Antrian
7. Jumlah Mahasiswa Sudah Diproses
8. Jumlah Mahasiswa Belum Diproses
9. Kosongkan Antrian
0. Keluar
Pilih menu: 3
Antrian kosong.
```