Laporan Hasil Praktikum

Algoritma Dan Struktur Data

Jobsheet 10

Angel Chelssa Leoniy Eka Permatasari

244107020202

1E

Program Studi Teknologi Informasi

Jurusan Teknik Informatika

POLINEMA

2025

PERCOBAAN 1

1. Membuat folder jobsheet 10 dengan nemabahkan file dengan nama Queue03.java
2. Tambahakn atribut sesuai dengan class, tambahkan konstruktor seperti yang telah ditentukan

```
public class Queue03 {
    int[] data;
    int front;
    int rear;
    int size;
    int max;

    public Queue03(int n) {
        max = n;
        data = new int[max];
        size = 0;
        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 kodsong: ");
    }
}
```

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

QueueMain03.java 1

```
public class QueueMain03 {
    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. Clear");
        System.out.println(x:"---------------------------------");
    }
```

11. Buat fungsi main, kemudian deklarasikan Scanner dengan nama sc.

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

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

```
Queue03 Q = new Queue03(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;
    }
} while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5);
```

Kode Program

Kode program class Queue03.java

```
public class Queue03 {
    int[] data;
    int front;
    int rear;
    int size;
    int max;

    public Queue03(int n) {
        max = n;
        data = new int[max];
        size = 0;
        front = rear = -1;
    }

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

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

```
            }
        }

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

        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);
            }
        }

        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 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++;
            }
        }

        public int Dequeue(){
            int dt = 0;
            if (IsEmpty()) {
                System.out.println("Queue masih kosong");
            } else {
                dt = data[front];
                size--;
```

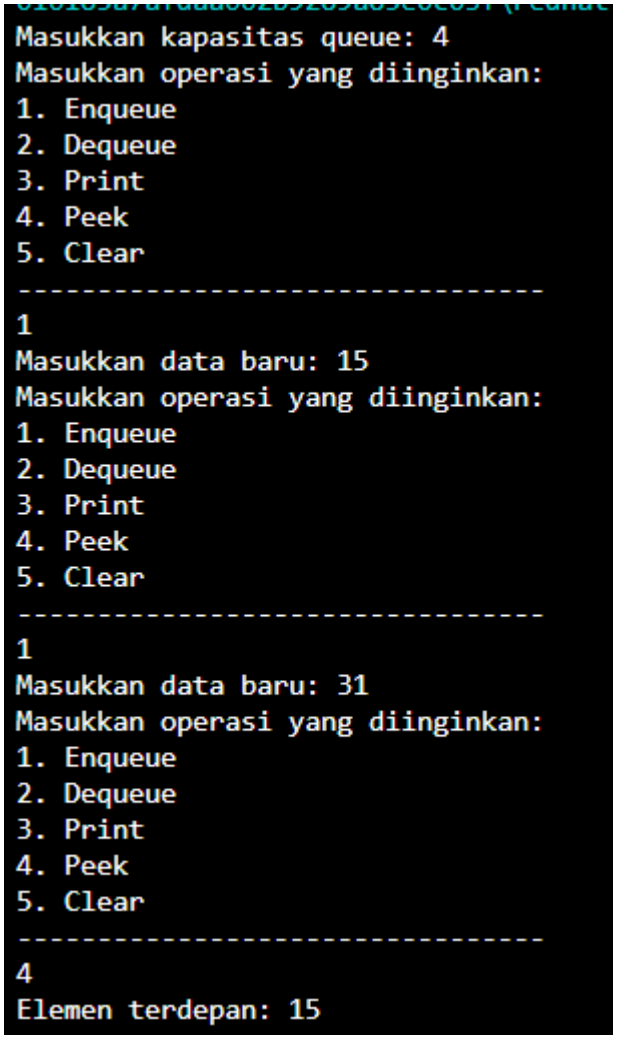
```
                if (IsEmpty()) {
                    front = rear = -1;
                } else {
                    if (front == max - 1) {
                        front = 0;
                    } else {
                        front++;
                    }
                }
            }
            return dt;
        }
    }
```

Kode program QueueMain03.java

```
    import java.util.Scanner;
    public class QueueMain03 {
        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("---------------------------------
    ");
        }

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

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

            Queue03 Q = new Queue03(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);
        }
    }
```

Hasil kode



PERTANYAAN

1. Karena -1 itu tandanya **belum ada data sama sekali** di antrian. Jadi posisi depan dan belakangnya belum valid. Sementara size = 0 ya karena isi antriannya emang kosong.
2. Untuk menambahkan data ke antrian. Jadi mengecek :
    a. Jika antrian penuh, muncul pesan “Queue penuh”.
    b. Jika kosong, posisi depan dan belakang diset ke 0.
    c. Jika udah ada isinya, rear digeser ke kanan buat nyiapin tempat data baru.
    d. Lalu, jika data baru masuk dan size ditambah.
3. Untuk mengambil data paling depan, lalu geser semua data lainnya agar rapat ke kiri.
    a. Jika data kosong, muncul pesan “Queue kosong”.
    b. Jika ada data, ambil data di depan (front), lalu data di belakangnya digeser semua ke kiri satu per satu.

c. rear dikurangi karena data berkurang.
d. size juga dikurangi, lalu data yang diambil dikembalikan.

4. Karena **data valid antrian itu mulai dari posisi front**, bukan dari 0. Bisa jadi posisi 0 udah dihapus sebelumnya lewat dequeue, jadi kita mulai dari yang masih aktif aja.

5. Itu buat **nampilin semua isi antrian**, mulai dari yang paling depan (front) sampai belakang (rear). Jadi biar kita bisa lihat siapa aja yang lagi antri sekarang.
6. Bagian yang menunjukan overflow adalah

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

Yang artinya tidak bisa menembah data karena antrian 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!

Jawaban :

```
System.exit(status:0);
```

Kode tersbut di tambahan dalam methode Enqueue dan method Dequeue

```
public void Enqueue(int dt) {
    if (IsFull()) {
        System.out.println(x:"Queue sudah penuh");
        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");
        System.exit(status:0);
    } else {
        dt = data[front];
        size--;
        if (IsEmpty()) {
            front = rear = -1;
        } else {
            if (front == max - 1) {
                front = 0;
            } else {
                front++;
            }
        }
    }
    return dt;
}
```

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.
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.
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.
5. Berikutnya method peek dan print yaitu untuk menampilkan data antrian layanan paling depan dan menampilkan semua data antrian layanan.

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

Kode program Mahasiswa03.java

```
public class Mahasiswa03 {

    String nim;
    String nama;
    String prodi;
    String kelas;

    public Mahasiswa03(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);
    }

}
```

Kode program AntrianLayanan03.java

```
public class AntrianLayanan03 {
    Mahasiswa03[] data;
    int front;
    int rear;
    int size;
    int max;

    public AntrianLayanan03(int max) {
        this.max = max;
        this.data = new Mahasiswa03[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;
```

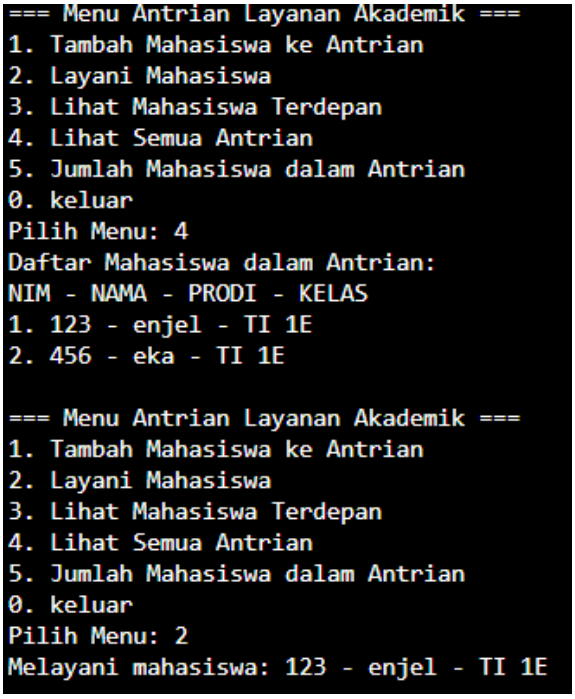
```
        }else {
            return false;
        }
    }

public void tambahAntrian(Mahasiswa03 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 Mahasiswa03 layaniMahasiswa04() {
        if(IsEmpty()) {
            System.out.println("Antrian kosong.");
            return null;
        }
        Mahasiswa03  mhs = data[front];
        front = (front + 1) % max;
        size--;
        return mhs;
    }

    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;
    }
}
```

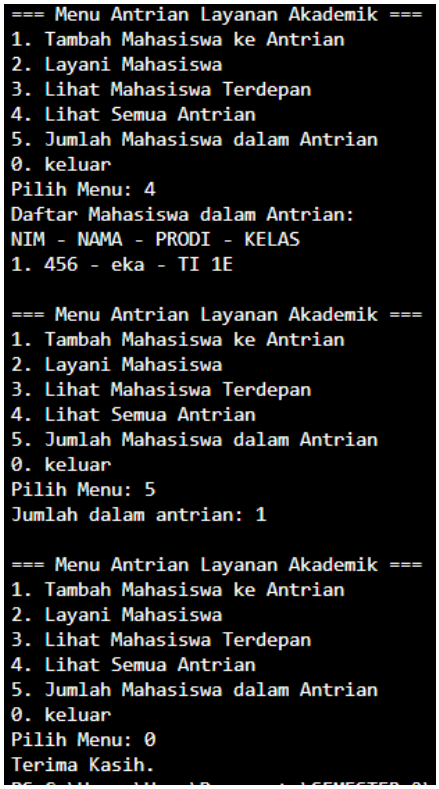
Kode program LayananAkademinSIAKAD03.java

```
import java.util.Scanner;
public class LayananAkademikSIAKAD03 {
```

```
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        AntrianLayanan03  antrian = new AntrianLayanan03 (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();
                    Mahasiswa03 mhs = new Mahasiswa03(nim, nama,
prodi, kelas);
                    antrian.tambahAntrian(mhs);
                    break;
                case 2:
                    Mahasiswa03 dilayani =
antrian.layaniMahasiswa04();
                    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();
    }
}
```

Hasil kode





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!

Kode program

Kode program class AntrianLayanan03.java

```
public class AntrianLayanan03 {
    Mahasiswa03[] data;
    int front;
    int rear;
    int size;
    int max;

    public AntrianLayanan03(int max) {
        this.max = max;
        this.data = new Mahasiswa03[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 tambahAntrian(Mahasiswa03 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 Mahasiswa03 layaniMahasiswa04() {
        if(IsEmpty()) {
            System.out.println("Antrian kosong.");
            return null;
        }
        Mahasiswa03  mhs = data[front];
        front = (front + 1) % max;
        size--;
        return mhs;
    }

    public void lihatTerdepan(){
        if(IsEmpty()) {
            System.out.println("Antrian kosong.");
        }else {
```

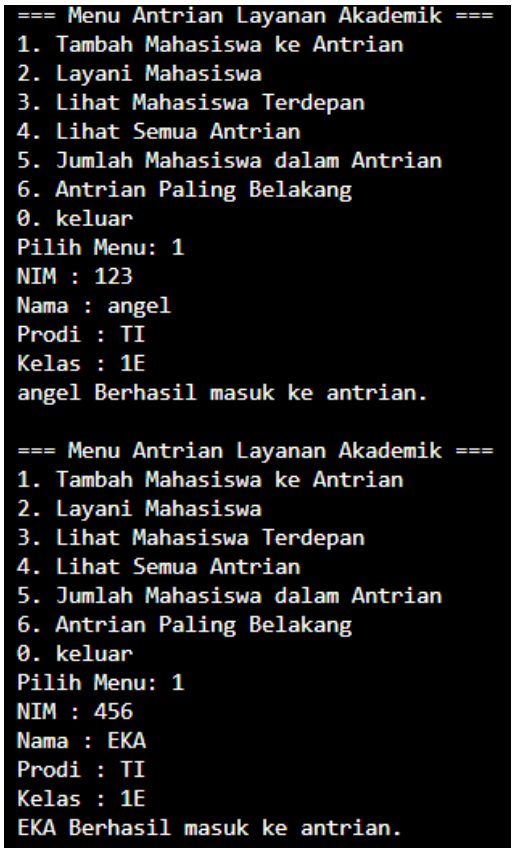
```
            System.out.print("Mahasiswa terdepan: ");
            System.out.println("NIM - NAMA - PRODI - KELAS");
            data[front].tampilkanData();
        }
    }

    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 lihatAkhir() {
        if (IsEmpty()) {
            System.out.println("Antrian kosong.");
        } else {
            System.out.println("Mahasiswa dalam antrian terkahir");
            System.out.println("NIM - NAMA - PRODI - KELAS");
            data[rear].tampilkanData();
        }
    }
}
```

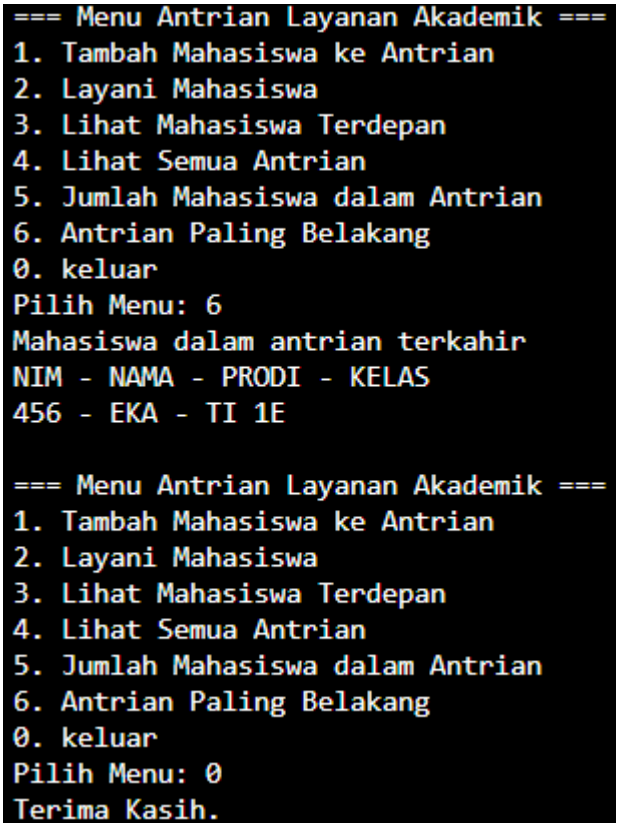
Kode program class LayanaAkademinSIAKAD03.java

```
import java.util.Scanner;
public class LayananAkademikSIAKAD03 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        AntrianLayanan03 antrian = new AntrianLayanan03(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("6. 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();
                    Mahasiswa03 mhs = new Mahasiswa03(nim, nama, prodi,
kelas);
                    antrian.tambahAntrian(mhs);
                    break;
                case 2:
                    Mahasiswa03 dilayani = antrian.layaniMahasiswa04();
                    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();
    }
}
```

Hasil kode





LATIHAN

Kode program

Kode program class Mahasiswa03.java

```
public class Mahasiswa03 {

    String nim;
    String nama;
    String prodi;
    String kelas;

    public Mahasiswa03(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);
    }
}
```

Kode program class AntrianKRS03.java

```
public class AntrianKRS03 {
    Mahasiswa03[]  data;
    int front;
    int rear;
    int size;
    int max;
```

```
    public AntrianKRS03(int max) {
        this.max = max;
        this.data = new Mahasiswa03[max];
        this.size = 0;
        this.front = 0;
        this.rear= -1;
    }

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

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

     public void tambahAntrian(Mahasiswa03 mhs) {
        if (IsFull()) {
            System.out.println("Antrian penuh, tidak dapat menambahkan
mahasiswa.");
            return;
        }
        rear = (rear + 1) % max;
        data [rear] = mhs;
        size++;
        System.out.println(mhs.nama + " berhasil masuk ke antrian.");
    }

    public Mahasiswa03[] layaniMahasiswa04() {
        System.out.println("Daftar Mahasiswa dalam Antrian: ");
        System.out.println("NIM - NAMA - PRODI - KELAS");

        if (IsEmpty()) {
            System.out.println("Antrian Kosong.");
            return null;
        }
        Mahasiswa03[] mhsDipanggil = new Mahasiswa03[2];
        for (int i = 0; i < 2; i++){
            if (size > 0) {
                mhsDipanggil[i] = data[front];
                front = (front + 1) % max;
                size--;
            }else {
                mhsDipanggil[i] = null;
            }
        }
        return mhsDipanggil;
    }

    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.println((i+1) + ". ");
            data[index].tampilkanData();
        }
    }

    public 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();
        }
    }

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

    public int getJumlahAntrian() {
        return size;
    }

    public int getJumlahProsesKRD() {
        return 30 - size;
    }

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

Kode program class LayananAkademikKRS03.java

```
import java.util.Scanner;
public class LayananAkademikSIAKAD03 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        AntrianLayanan03 antrian = new AntrianLayanan03(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("6. 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();
                    Mahasiswa03 mhs = new Mahasiswa03(nim, nama, prodi,
kelas);
                    antrian.tambahAntrian(mhs);
                    break;
                case 2:
                    Mahasiswa03 dilayani = antrian.layaniMahasiswa04();
                    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();
    }
}
```

Hasil kode

```
1. Tambah Mahasiswa ke Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Jumlah Antrian
7. Jumlah Mahasiswa yang Belum Proses KRS
0. Keluar
Pilih Menu: 1
NIM : 123
Nama : enjel
Prodi : TI
Kelas : 1E
enjel berhasil masuk ke antrian.

=== Menu Antrian KRS ===
1. Tambah Mahasiswa ke Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Jumlah Antrian
7. Jumlah Mahasiswa yang Belum Proses KRS
0. Keluar
Pilih Menu: 1
NIM : 456
Nama : leoniy
Prodi : TI
Kelas : 1E
leoniy berhasil masuk ke antrian.
```

```
=== Menu Antrian KRS ===
1. Tambah Mahasiswa ke Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Jumlah Antrian
7. Jumlah Mahasiswa yang Belum Proses KRS
0. Keluar
Pilih Menu: 2
Mahasiswa yang dipanggil untuk proses KRS:
Daftar Mahasiswa dalam Antrian:
NIM - NAMA - PRODI - KELAS
123 - enjel - TI 1E
456 - leoniy - TI 1E

=== Menu Antrian KRS ===
1. Tambah Mahasiswa ke Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Jumlah Antrian
7. Jumlah Mahasiswa yang Belum Proses KRS
0. Keluar
Pilih Menu: 3
Antrian kosong.

=== Menu Antrian KRS ===
1. Tambah Mahasiswa ke Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Jumlah Antrian
7. Jumlah Mahasiswa yang Belum Proses KRS
0. Keluar
Pilih Menu: 4
Antrian kosong.
```

```
=== Menu Antrian KRS ===
1. Tambah Mahasiswa ke Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Jumlah Antrian
7. Jumlah Mahasiswa yang Belum Proses KRS
0. Keluar
Pilih Menu: 5
Antrian kosong.

=== Menu Antrian KRS ===
1. Tambah Mahasiswa ke Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Jumlah Antrian
7. Jumlah Mahasiswa yang Belum Proses KRS
0. Keluar
Pilih Menu: 6
Jumlah antrian: 0

=== Menu Antrian KRS ===
1. Tambah Mahasiswa ke Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Jumlah Antrian
7. Jumlah Mahasiswa yang Belum Proses KRS
0. Keluar
Pilih Menu: 7
Jumlah mahasiswa yang belum melakukan proses KRS: 0
```

```
=== Menu Antrian KRS ===
1. Tambah Mahasiswa ke Antrian
2. Proses KRS (2 Mahasiswa)
3. Tampilkan Semua Antrian
4. Tampilkan 2 Antrian Terdepan
5. Tampilkan Antrian Paling Akhir
6. Jumlah Antrian
7. Jumlah Mahasiswa yang Belum Proses KRS
0. Keluar
Pilih Menu: 0
Terima kasih
```