LAPORAN HASIL PRAKTIKUM

Algoritma dan Struktur Data

Jobsheet 10

NAMA : Handino Asa Galih r

NIM : 244107020237

KELAS : 1-E

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLINEMA

2025

2.Praktikum 1

Percobaan 1 : Operasi Dasar Queue

```
public class Queue {

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

    public Queue(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 kosong.");
        }
    }

    public 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);
        }
    }

    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");
            System.exit(1);
```

```
        } else {
            if (IsEmpty()) {
                front = rear = 0;
            } else {
                if (rear == max - 1) {
                    rear = 0;
                } else {
                    rear++;
                }
            }
            data[rear] = dt;
            size++;
        }
    }

    public int Dequeue() {
        int dt = 0;
        if (IsEmpty()) {
            System.out.println("Queue masih kosong");
            System.exit(1);
        } else {
            dt = data[front];
            size--;
            if (IsEmpty()) {
                front = rear = -1;
            } else {
                if (front == max - 1) {
                    front = 0;
                } else {
                    front++;
                }
            }
        }
        return dt;
    }
```

Kode program class QueueMain

```
import java.util.Scanner;

public class QueueMain {
    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();

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

Output

```
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
```

Pertanyaan :

1. Pada konstruktor, mengapa nilai awal atribut front dan rear bernilai -1, sementara atribut size bernilai 0?

   Alasan, Nilai awal atribut front dan rear bernilai -1 karena bentuk dari queue ini seperti array jadi wajib diinisialisasi dari -1 yang berarti “kosong” karena indeks array mulai dari 0, jika atribut size bernilai 0 karena sesuai dengan ukuran queue tersebut. Dan jika belum terisi, maka diawal pasti bernilai 0 lalu akan bertambah sesuai pengisiannya nanti.

2. Pada method Enqueue, jelaskan maksud dan kegunaan dari potongan kode berikut!

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

   Ketika rear sudah mencapai indeks paling terakhir dari array, meskipun rear sudah mencapi indeks paling belakang maka ada ruang kosong di awal array (karena beberapa elemen di depan sudah di-dequeue), maka rear akan direset ke posisi 0.

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

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

   Saat operasi dequeue dilakukan hingga pointer front sudah mencapai indeks paling terakhir array, maka front direset ke posisi 0.

4. Pada method print, mengapa pada proses perulangan variabel i tidak dimulai dari 0 (int i=0), melainkan int i=front?

   Karena, elemen-elemen queue tidak selalu berada di posisi awal array. Posisi awal elemen queue ditunjukkan oleh front, front dapat berada di posisi mana saja dalam array karena konsep queue FIFO (First In First Out).

5. Perhatikan kembali method print, jelaskan maksud dari potongan kode berikut!

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

   Operasi modulo memastikan bahwa ketika i melebihi batas array, maka nilai i akan kembali ke 0, sehingga iterasi dapat berlanjut dari awal array tersebut.

6. Tunjukkan potongan kode program yang merupakan queue overflow!

```
public void Enqueue(int dt) {
    if (IsFull()) {
        System.out.println(x:"Queue sudah penuh");
        System.exit(status:1);
    } else {
```

7. pada saat terjadi queue overflow dan queue underflow, program tersebut tetap dapat berjalan dan hanya menampilkan teks informasi. Lakukan modifikasi program sehingga pada saat terjadi queue overflow dan queue underflow, program dihentikan!

Kode program yang dirubah menggunakan metode helper di class :

```
public void Enqueue(int dt) {
    if (IsFull()) {
        System.out.println(x:"Queue overflow!");
        System.out.println(x:"Program berhenti karena queue overflow!");
    } else {
        if (IsEmpty()) {
            front = rear = 0;
        } else {
            rear = (rear + 1) % max;
        }
        data[rear] = dt;
        size++;
    }
}

ic int Dequeue() {
if (IsEmpty()) {
    System.out.println(x:"Queue underflow!");
    System.out.println(x:"Program berhenti karena queue underflow!");
    return -1;
} else {
    int dt = data[front];
    size--;
    if (IsEmpty()) {
        front = rear = -1;
    } else {
        front = (front + 1) % max;
    }
    return dt;
}
```

Lalu, lakukan perubahan juga pada QueueMain.

```
case 1:
    System.out.print(s:"Masukkan data baru: ");
    int dataMasuk = sc.nextInt();
    Q.Enqueue(dataMasuk);
    break;
case 2:
    int dataKeluar = Q.Dequeue();
    System.out.println("Data yang dikeluarkan: " + dataKeluar);
    break;
```

Percobaan 2 : Antrian Layanan Akademik

Kode Program class Mahasiswa

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

    public Mahasiswa(String nim, String nama, String prodi, String
kelas) {
        this.nim = nim;
        this.nama = nama;
        this.prodi = prodi;
        this.kelas = kelas;
    }

    public void tampilkanData() {
        System.out.println(nim + " - " + nama + " - " + prodi + " - " +
kelas);
    }
}
```

Kode program AntrianLayanan

```
public class AntrianLayanan {

    Mahasiswa[] data;
    int front;
    int rear;
    int size;
    int max;

    public AntrianLayanan(int max) {
        this.max = max;
        this.data = new Mahasiswa[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 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(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.");
    }

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

Kode program class LayananAkademikSIAKAD

```
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("6. Cek Antrian paling belakang");
            System.out.println("0. Keluar");
            System.out.print("Pilih menu: ");
            pilihan = sc.nextInt();
            sc.nextLine();
            switch (pilihan) {
                case 1:
                    System.out.print("NIM   : ");
                    String nim = sc.nextLine();
                    System.out.print("Nama  : ");
                    String nama = sc.nextLine();
                    System.out.print("Prodi : ");
                    String prodi = sc.nextLine();
                    System.out.print("Kelas : ");
                    String kelas = sc.nextLine();
                    Mahasiswa mhs = new Mahasiswa(nim, nama, prodi,
kelas);
                    antrian.tambahAntrian(mhs);
                    break;
                case 2:
                    Mahasiswa dilayani = antrian.layaniMahasiswa();
                    if (dilayani != null) {
                        System.out.print("Melayani mahasiswa: ");
                        dilayani.tampilkanData();
                    }
                    break;
                case 3:
                    antrian.lihatTerdepan();
                    break;
                case 4:
                    antrian.tampilkanSemua();
                    break;
```

```
                    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();
    }
}
```

Output!

```
 === 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
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
6. Cek Antrian paling belakang
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
6. Cek Antrian paling belakang
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
6. Cek Antrian paling belakang
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
6. Cek Antrian paling belakang
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
6. Cek Antrian paling belakang
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
6. Cek Antrian paling belakang
0. Keluar
Pilih menu: 0
Terima kasih.
```

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!

Melakukan penambahan method baru di kode program class Antrianlayanan dengan nama method LihatAkhir

```
    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();
        }
    }
}
```

Lalu, tambahkan juga dafar menu 6. Kemudian cek antrian paling belakang di class LayananAkademikSIAKAD

```
System.out.println(x:"6. Cek Antrian paling belakang");
```

```
case 6:
    antrian.lihatAkhir();
    break;
case 0:
```

Tugas

Kode Program class Mahasiswa

```
public class Mahasiswa {

    String nim;

    String nama;

    String prodi;

    String kelas;


    public Mahasiswa(String nim, String nama, String prodi, String
kelas) {

        this.nim = nim;

        this.nama = nama;

        this.prodi = prodi;

        this.kelas = kelas;

    }


    public void tampilkanData() {

        System.out.printf("%-12s %-20s %-15s %-10s\n", nim, nama, prodi,
kelas);

    }

}
```

Kode program class AntrianKRS

```
public class AntrianKRS {
    Mahasiswa[] data;
    int front;
    int rear;
    int size;
    int max;
    int jmlSudahKRS = 0;
    int maksimalKRS = 30;

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

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

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

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

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

    public void layaniKRS() {
        if (isEmpty()) {
            System.out.println("Tidak Ada Antrian.");
            return;
        }

        int sisaKRS = maksimalKRS - jmlSudahKRS;

        if (sisaKRS <= 0) {
            System.out.println("Proses KRS dihentikan. DPA sudah
melayani maksimal 30 mahasiswa.");
            return;
        }

        if (size == 1 || sisaKRS == 1) {
            System.out.println("Proses KRS untuk 1 mahasiswa:");
            Mahasiswa mhs1 = data[front];
            System.out.printf("%-12s %-20s %-15s %-10s\n", "NIM",
"Nama", "Prodi", "Kelas");
            System.out.printf("%-4d ", 1);
            mhs1.tampilkanData();

            front = (front + 1) % max;
            size--;
```

```
            jmlSudahKRS++;
            System.out.println("Proses KRS selesai untuk 1 mahasiswa.");
        } else {
            System.out.println("Proses KRS untuk 2 mahasiswa:");
            System.out.printf("%-4s %-12s %-20s %-15s %-10s\n", "No",
"NIM", "Nama", "Prodi", "Kelas");
            System.out.println("----------------------------------------
----------------------");

            Mahasiswa mhs1 = data[front];
            Mahasiswa mhs2 = data[(front + 1) % max];

            System.out.printf("%-4d ", 1);
            mhs1.tampilkanData();

            System.out.printf("%-4d ", 2);
            mhs2.tampilkanData();

            front = (front + 2) % max;
            size -= 2;
            jmlSudahKRS += 2;
            System.out.println("Proses KRS selesai untuk 2 mahasiswa.");
        }
    }

    public void tampilkanSemua() {
        if (isEmpty()) {
            System.out.println("Antrian kosong.");
        } else {
            System.out.println("Daftar mahasiswa dalam antrian:");
            System.out.println("----------------------------------------
----------------------");
            System.out.printf("%-4s %-12s %-20s %-15s %-10s\n", "No",
"NIM", "Nama", "Prodi", "Kelas");
            System.out.println("----------------------------------------
----------------------");
```

```
            for (int i = 0; i < size; i++) {
                int index = (front + i) % max;
                System.out.printf("%-4d ", (i + 1));
                data[index].tampilkanData();
            }
            System.out.println("----------------------------------------
----------------------");
        }
    }

    public void tampilkanTerdepan() {
        if (isEmpty()) {
            System.out.println("Tidak Ada Antrian.");
        } else if (size == 1) {
            System.out.println("Mahasiswa terdepan:");
            System.out.printf("%-12s %-20s %-15s %-10s\n", "NIM",
"Nama", "Prodi", "Kelas");
            data[front].tampilkanData();
        } else {
            System.out.println("Dua mahasiswa terdepan:");
            System.out.printf("%-4s %-12s %-20s %-15s %-10s\n", "No",
"NIM", "Nama", "Prodi", "Kelas");
            System.out.println("----------------------------------------
----------------------");

            Mahasiswa mhs1 = data[front];
            Mahasiswa mhs2 = data[(front + 1) % max];

            System.out.printf("%-4d ", 1);
            mhs1.tampilkanData();

            System.out.printf("%-4d ", 2);
            mhs2.tampilkanData();

            System.out.println("----------------------------------------
----------------------");
        }
```

```
    }


    public void tampilkanAkhir() {
        if (isEmpty()) {
            System.out.println("Tidak Ada Antrian.");
        } else {
            System.out.println("Mahasiswa terakhir:");
            System.out.printf("%-12s %-20s %-15s %-10s\n", "NIM",
"Nama", "Prodi", "Kelas");
            data[rear].tampilkanData();
        }
    }


    public int getJumlahAntrian() {
        return size;
    }


    public int getJumlahKRS() {
        return jmlSudahKRS;
    }


    public int getJumlahBelumKRS() {
        return maksimalKRS - jmlSudahKRS;
    }
}
```

ClassMain LayananKRS

```
public class LayananKRS {

    public static void menu() {

        System.out.println("\n+-----------------------------------------
--+");

        System.out.println("| No |          Menu Layanan
KRS            |");

        System.out.println("+----+-------------------------------------
+");

        System.out.println("| 1  | Tambah Mahasiswa ke
Antrian          |");

        System.out.println("| 2  | Proses KRS
Mahasiswa                 |");

        System.out.println("| 3  | Lihat Antrian Mahasiswa
Terdepan     |");

        System.out.println("| 4  | Lihat Semua
Antrian                  |");

        System.out.println("| 5  | Lihat Antrian Mahasiswa
Terakhir     |");

        System.out.println("| 6  | Jumlah
Antrian                       |");

        System.out.println("| 7  | Jumlah yang Sudah Proses
KRS         |");

        System.out.println("| 8  | Jumlah yang Belum Proses
KRS         |");

        System.out.println("| 9  | Kosongkan
Antrian                    |");

        System.out.println("| 0  |
Keluar                                |");

        System.out.println("+-----------------------------------------
+");

        System.out.print("Pilih menu: ");

    }


    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        AntrianKRS antrian = new AntrianKRS(10);

        int pilihan;
```

```
        do {
            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:
                    antrian.layaniKRS();
                    break;
                case 3:
                    antrian.tampilkanTerdepan();
                    break;
                case 4:
                    antrian.tampilkanSemua();
                    break;
                case 5:
                    antrian.tampilkanAkhir();
                    break;
                case 6:
                    System.out.println("Jumlah antrian: " +
antrian.getJumlahAntrian());
                    break;
```

```
                case 7:
                    System.out.println("Jumlah yang sudah proses KRS: "
+ antrian.getJumlahKRS());
                    break;
                case 8:
                    System.out.println("Jumlah yang belum proses KRS: "
+ antrian.getJumlahBelumKRS());
                    break;
                case 9:
                    antrian.clear();
                    break;
                case 0:
                    System.out.println("Terima kasih Telah Menggunakan
Layanan KRS. Have a nice day!");
                    break;
                default:
                    System.out.println("Pilihan tidak valid. Silahkan
pilih menu yang tersedia.");
            }
        } while (pilihan != 0);


        sc.close();
    }
}
```

Output

```
+-----------------------------------------+
| No |          Menu Layanan KRS          |
+----+-----------------------------------+
| 1  | Tambah Mahasiswa ke Antrian        |
| 2  | Proses KRS Mahasiswa               |
| 3  | Lihat Antrian Mahasiswa Terdepan   |
| 4  | Lihat Semua Antrian                |
| 5  | Lihat Antrian Mahasiswa Terakhir   |
| 6  | Jumlah Antrian                     |
| 7  | Jumlah yang Sudah Proses KRS       |
| 8  | Jumlah yang Belum Proses KRS       |
| 9  | Kosongkan Antrian                  |
| 0  | Keluar                             |
+-----------------------------------------+
Pilih menu: 1
NIM   : 2441
Nama  : Dino Asa Galih R
Prodi : TI
Kelas : 1E
Dino Asa Galih R berhasil masuk ke antrian.

+-----------------------------------------+
| No |          Menu Layanan KRS          |
+----+-----------------------------------+
| 1  | Tambah Mahasiswa ke Antrian        |
| 2  | Proses KRS Mahasiswa               |
| 3  | Lihat Antrian Mahasiswa Terdepan   |
| 4  | Lihat Semua Antrian                |
| 5  | Lihat Antrian Mahasiswa Terakhir   |
| 6  | Jumlah Antrian                     |
| 7  | Jumlah yang Sudah Proses KRS       |
| 8  | Jumlah yang Belum Proses KRS       |
| 9  | Kosongkan Antrian                  |
| 0  | Keluar                             |
+-----------------------------------------+
Pilih menu: 1
NIM   : 2442
Nama  : Doni Rahmatullah
Prodi : TI
Kelas : 1E
Doni Rahmatullah berhasil masuk ke antrian.
```

```
+-----------------------------------------+
| No |          Menu Layanan KRS          |
+----+-----------------------------------+
| 1  | Tambah Mahasiswa ke Antrian        |
| 2  | Proses KRS Mahasiswa               |
| 3  | Lihat Antrian Mahasiswa Terdepan   |
| 4  | Lihat Semua Antrian                |
| 5  | Lihat Antrian Mahasiswa Terakhir   |
| 6  | Jumlah Antrian                     |
| 7  | Jumlah yang Sudah Proses KRS       |
| 8  | Jumlah yang Belum Proses KRS       |
| 9  | Kosongkan Antrian                  |
| 0  | Keluar                             |
+-----------------------------------------+
Pilih menu: 1
NIM   : 2443
Nama  : Galih Handayani
Prodi : TI
Kelas : 1E
Galih Handayani berhasil masuk ke antrian.

+-----------------------------------------+
| No |          Menu Layanan KRS          |
+----+-----------------------------------+
| 1  | Tambah Mahasiswa ke Antrian        |
| 2  | Proses KRS Mahasiswa               |
| 3  | Lihat Antrian Mahasiswa Terdepan   |
| 4  | Lihat Semua Antrian                |
| 5  | Lihat Antrian Mahasiswa Terakhir   |
| 6  | Jumlah Antrian                     |
| 7  | Jumlah yang Sudah Proses KRS       |
| 8  | Jumlah yang Belum Proses KRS       |
| 9  | Kosongkan Antrian                  |
| 0  | Keluar                             |
+-----------------------------------------+
Pilih menu: 1
NIM   : 2444
Nama  : Asa Ramadhan
Prodi : TI
Kelas : 1E
Asa Ramadhan berhasil masuk ke antrian.
```

```
+----------------------------------------+
| No |          Menu Layanan KRS         |
+----+-----------------------------------+
| 1  | Tambah Mahasiswa ke Antrian       |
| 2  | Proses KRS Mahasiswa              |
| 3  | Lihat Antrian Mahasiswa Terdepan  |
| 4  | Lihat Semua Antrian               |
| 5  | Lihat Antrian Mahasiswa Terakhir  |
| 6  | Jumlah Antrian                    |
| 7  | Jumlah yang Sudah Proses KRS      |
| 8  | Jumlah yang Belum Proses KRS      |
| 9  | Kosongkan Antrian                 |
| 0  | Keluar                            |
+----------------------------------------+
Pilih menu: 2
Proses KRS untuk 2 mahasiswa:
No   NIM          Nama                Prodi           Kelas
-----------------------------------------------------------------
1    2441         Dino Asa Galih R    TI              1E
2    2442         Doni Rahmatullah    TI              1E
Proses KRS selesai untuk 2 mahasiswa.

+----------------------------------------+
| No |          Menu Layanan KRS         |
+----+-----------------------------------+
| 1  | Tambah Mahasiswa ke Antrian       |
| 2  | Proses KRS Mahasiswa              |
| 3  | Lihat Antrian Mahasiswa Terdepan  |
| 4  | Lihat Semua Antrian               |
| 5  | Lihat Antrian Mahasiswa Terakhir  |
| 6  | Jumlah Antrian                    |
| 7  | Jumlah yang Sudah Proses KRS      |
| 8  | Jumlah yang Belum Proses KRS      |
| 9  | Kosongkan Antrian                 |
| 0  | Keluar                            |
+----------------------------------------+
Pilih menu: 3
Dua mahasiswa terdepan:
No   NIM          Nama                Prodi           Kelas
-----------------------------------------------------------------
1    2443         Galih Handayani     TI              1E
2    2444         Asa Ramadhan        TI              1E
-----------------------------------------------------------------
```

```
+-----------------------------------------+
| No |          Menu Layanan KRS          |
+----+-----------------------------------+
| 1  | Tambah Mahasiswa ke Antrian        |
| 2  | Proses KRS Mahasiswa               |
| 3  | Lihat Antrian Mahasiswa Terdepan   |
| 4  | Lihat Semua Antrian                |
| 5  | Lihat Antrian Mahasiswa Terakhir   |
| 6  | Jumlah Antrian                     |
| 7  | Jumlah yang Sudah Proses KRS       |
| 8  | Jumlah yang Belum Proses KRS       |
| 9  | Kosongkan Antrian                  |
| 0  | Keluar                             |
+-----------------------------------------+
Pilih menu: 5
Mahasiswa terakhir:
NIM          Nama                Prodi          Kelas
2444         Asa Ramadhan        TI             1E
```

```
+-----------------------------------------+
| No |          Menu Layanan KRS          |
+----+-----------------------------------+
| 1  | Tambah Mahasiswa ke Antrian        |
| 2  | Proses KRS Mahasiswa               |
| 3  | Lihat Antrian Mahasiswa Terdepan   |
| 4  | Lihat Semua Antrian                |
| 5  | Lihat Antrian Mahasiswa Terakhir   |
| 6  | Jumlah Antrian                     |
| 7  | Jumlah yang Sudah Proses KRS       |
| 8  | Jumlah yang Belum Proses KRS       |
| 9  | Kosongkan Antrian                  |
| 0  | Keluar                             |
+-----------------------------------------+
Pilih menu: 6
Jumlah antrian: 2
```

```
+-----------------------------------------+
| No |          Menu Layanan KRS          |
+----+-----------------------------------+
| 1  | Tambah Mahasiswa ke Antrian        |
| 2  | Proses KRS Mahasiswa               |
| 3  | Lihat Antrian Mahasiswa Terdepan   |
| 4  | Lihat Semua Antrian                |
| 5  | Lihat Antrian Mahasiswa Terakhir   |
| 6  | Jumlah Antrian                     |
| 7  | Jumlah yang Sudah Proses KRS       |
| 8  | Jumlah yang Belum Proses KRS       |
| 9  | Kosongkan Antrian                  |
| 0  | Keluar                             |
+-----------------------------------------+
Pilih menu: 7
Jumlah yang sudah proses KRS: 2
```

```
+-----------------------------------------+
| No |          Menu Layanan KRS          |
+----+-----------------------------------+
| 1  | Tambah Mahasiswa ke Antrian        |
| 2  | Proses KRS Mahasiswa               |
| 3  | Lihat Antrian Mahasiswa Terdepan   |
| 4  | Lihat Semua Antrian                |
| 5  | Lihat Antrian Mahasiswa Terakhir   |
| 6  | Jumlah Antrian                     |
| 7  | Jumlah yang Sudah Proses KRS       |
| 8  | Jumlah yang Belum Proses KRS       |
| 9  | Kosongkan Antrian                  |
| 0  | Keluar                             |
+-----------------------------------------+
Pilih menu: 8
Jumlah yang belum proses KRS: 28
```


```
+-----------------------------------------+
| No |          Menu Layanan KRS          |
+----+-----------------------------------+
| 1  | Tambah Mahasiswa ke Antrian        |
| 2  | Proses KRS Mahasiswa               |
| 3  | Lihat Antrian Mahasiswa Terdepan   |
| 4  | Lihat Semua Antrian                |
| 5  | Lihat Antrian Mahasiswa Terakhir   |
| 6  | Jumlah Antrian                     |
| 7  | Jumlah yang Sudah Proses KRS       |
| 8  | Jumlah yang Belum Proses KRS       |
| 9  | Kosongkan Antrian                  |
| 0  | Keluar                             |
+-----------------------------------------+
Pilih menu: 8
Jumlah yang belum proses KRS: 28
```