

Laporan Hasil Pratikum  
Algoritma Dan Struktur Data  
Case Method 2



Disusun oleh :

Kelompok 4

- |    |                                  |                |
|----|----------------------------------|----------------|
| 1. | Muhammad Daffi Firos Zaidan (17) | (244107020182) |
| 2. | Raditya Riefki (23)              | (244107020204) |
| 3. | Zacky Rio Orlando (27)           | (244107020086) |

Program Studi D-IV Teknik Informatika

Jurusan Teknologi Informasi

Praktikum

2025

## Studi Kasus - Sistem Antrian Pasien di Klinik Dokter

Untuk menangani sistem antrian pasien di sebuah klinik dokter, diperlukan sebuah system sederhana yang mampu memproses beberapa aktivitas:

a. Menambahkan antrian pasien yang datang ke klinik menggunakan struktur data **Linked List**

b. Menampilkan daftar pasien dalam antrian saat ini.

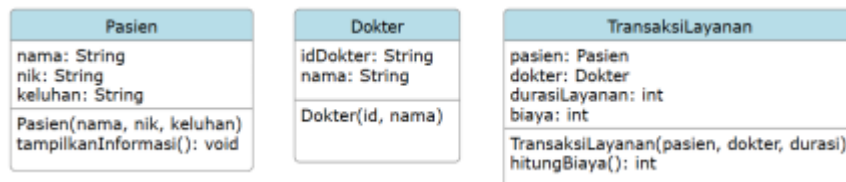
c. Melayani pasien berdasarkan urutan kedatangan dengan mencatat transaksi layanan untuk setiap pasien yang telah dilayani, meliputi ID dokter, nama dokter, dan durasi layanan dalam jam. Biaya layanan dihitung berdasarkan durasi layanan dengan tariff Rp 50.000 per jam. Transaksi ini disimpan menggunakan struktur **Queue (array)**.

d. Mengecek sisa pasien yang masih menunggu di antrian.

e. Menampilkan riwayat transaksi layanan yang sudah dilakukan.

Perhatikan bahwa pasien yang datang masuk ke antrian menggunakan **Linked List**. Setelah pasien dilayani, maka admin memasukkan data dokter yang melayani kemudian transaksi layanan (termasuk perhitungan biaya layanan) tersebut disimpan ke dalam **Queue (array)**.

Buatlah program Java dengan menggunakan class diagram berikut:



- Kode Program

- Class Pasien

```
package CaseMethod2;

public class Pasien {
    public String nama;
    public String nik;
    public String keluhan ;
    ;

    public Pasien(String nama, String nik, String keluhan) {
        this.nama = nama;
        this.nik = nik;
        this.keluhan = keluhan;
    }

    public void tampil(){
        System.out.println("Nama pasien: " + nama);
        System.out.println("NIK: " + nik );
        System.out.println("Keluhan: " + keluhan );
    }

    public String getNama() {
        return nama;
    }

}
```

- Class NodePasien

```
package CaseMethod2;

public class NodePasien {
    Pasien data;
    NodePasien next;

    public NodePasien(Pasien data, NodePasien next) {
        this.data = data;
        this.next = next;
    }

}
```

## ➤ Class Dokter

```
package CaseMethod2;

public class Dokter {
    public String idDokter;
    public String nama;

    public Dokter(String idDokter, String nama) {
        this.idDokter = idDokter;
        this.nama = nama;
    }
}
```

## ➤ Class AntrianPasien

```
package CaseMethod2;

import java.util.Scanner;

public class AntrianPasien {
    public Scanner sc = new Scanner(System.in);
    NodePasien head;
    NodePasien tail;
    QueueTransaksi transaksi = new QueueTransaksi(10);

    int maxSize = 10;
    int size = 0;

    public AntrianPasien(QueueTransaksi transaksi) {
        this.transaksi = transaksi;
    }

    public boolean isEmpty(){
        return (head == null);
    }

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

    public void TambahPasien(Pasien data){
        if (isFull()) {
            System.out.println("Antrian Penuh.");
        }
        NodePasien ndInput = new NodePasien(data, null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        }else{
            tail.next = ndInput;
            tail = ndInput;
        }
    }

    public void LihatAntrian(){
        if (!isEmpty()) {
            NodePasien temp = head;
            System.out.println("-- Antrian Pasien --");
            System.out.println("Antrian: \t");
            while (temp != null) {
                temp.data.tampil();
                temp = temp.next;
                System.out.println();
                System.out.println("=====");
            }
        }else{
            System.out.println("Antrian masih kosong!");
        }
    }
}
```

```

public void LayaniPasien() {
    if (isEmpty()) {
        System.out.println("Antrian masih kosong!");
        return;
    }

    Pasien pasienDilayani = head.data;

    if (head == tail) {
        head = tail = null;
    } else {
        head = head.next;
    }
    System.out.println("--- Layani Pasien ---");
    System.out.println("Pasien " + pasienDilayani.getNama() + "
dipanggil");

    Scanner sc = new Scanner(System.in);
    System.out.print("Masukkan ID Dokter: ");
    String idDokter = sc.nextLine();

    System.out.print("Masukkan Nama Dokter: ");
    String namaDokter = sc.nextLine();

    System.out.print("Masukkan Durasi Layanan (jam): ");
    int durasi = sc.nextInt();
    sc.nextLine();

    Dokter dokter = new Dokter(idDokter, namaDokter);
    TransaksiLayanan riwayatPasien = new TransaksiLayanan(pasienDilayani,
dokter, durasi);

    transaksi.enqueue(riwayatPasien);

    System.out.println(">> Pasien telah dilayani, transaksi berhasil
dicatat.");
    System.out.println();
}

public void cekSisaAntrian () {
    int count = -0;
    NodePasien temp = head;

    while (temp != null) {
        temp = temp.next;
        count++;
    }
    System.out.println(">> Sisa Pasien dalam antrian : " + count);
}
}

```

## ➤ Class QueueTransaksi

```
package CaseMethod2;

public class QueueTransaksi {
    public TransaksiLayanan [] queue;
    public int front;
    public int rear;
    public int size;
    public int maxSize;

    public QueueTransaksi(int maxSize) {
        this.maxSize = maxSize;
        this.queue = new TransaksiLayanan[maxSize];
        this.front = 0;
        this.rear = -1;
        this.size = 0;
    }

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

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

    public void enqueue(TransaksiLayanan transaksi) {
        if (isFull()) {
            System.out.println("Transaksi Penuh.");
            return;
        }

        rear = (rear + 1) % maxSize;
        queue[rear] = transaksi;
        size++;
    }

    public void tampilkanRiwayatTransaksi() {
        if (size == 0) {
            System.out.println("Belum ada riwayat transaksi layanan.");
            return;
        }

        System.out.println("\n--- Riwayat Transaksi ---");
        int index = front;

        for (int i = 0; i < size; i++) {
            System.out.println(queue[index].pasien.nama + " (" +
                queue[index].durasiLayanan + " jam): Rp " + queue[index].biaya);
            index = (index + 1) % maxSize;
        }
        System.out.println("Total transaksi: " + size);
    }
}
```

## ➤ Class TransaksiLayanan

```
package CaseMethod2;

public class TransaksiLayanan {
    Dokter dokter;
    Pasien pasien;
    public int durasiLayanan;
    public int biaya;

    public TransaksiLayanan (Pasien pasien, Dokter dokter, int
durasiLayanan) {
        this.pasien = pasien;
        this.dokter = dokter;
        this.durasiLayanan = durasiLayanan;
        this.biaya = hitungBiaya();
    }

    public int hitungBiaya(){
        return durasiLayanan * 50000;
    }

    public void tampilTransaksi(){
        System.out.println(pasien.nama + "(" +durasiLayanan +")"+ ":
Rp " + biaya);
    }
}
```



## ➤ Class SistemAntrianKlinik

```
package CaseMethod2;

import java.util.Scanner;

public class SistemAntrianKlinik {
    static Scanner sc = new Scanner(System.in);
    public static Pasien tambahPasien(){
        System.out.print("Nama Pasien: ");
        String nama = sc.nextLine();
        System.out.print("NIK: ");
        String nik = sc.nextLine();
        System.out.print("Keluhan: ");
        String keluhan = sc.nextLine();
        Pasien p = new Pasien(nama, nik, keluhan);

        System.out.println(">> Pasien masuk ke dalam antrian.");
        System.out.println();
        return p;
    }

    public static void main(String[] args) {
        QueueTransaksi t = new QueueTransaksi(10);
        AntrianPasien list = new AntrianPasien(t);
        int pilihan;

        do{
            System.out.println("=== SISTEM ANTRIAN KLINIK ===");
            System.out.println("1. Tambah Pasien ke Antrian");
            System.out.println("2. Lihat Antrian");
            System.out.println("3. Layani Pasien");
            System.out.println("4. Cek Sisa Antrian Pasien");
            System.out.println("5. Lihat Riwayat Transaksi");
            System.out.println("0. Keluar");
            System.out.print("Pilih Menu: ");
            pilihan = sc.nextInt();
            sc.nextLine();

            switch (pilihan) {
                case 1:
                    Pasien p = tambahPasien();
                    list.TambahPasien(p);
                    break;
                case 2:
                    list.LihatAntrian();
                    break;
                case 3:
                    list.LayaniPasien();
                    break;
                case 4:
                    list.cekSisaAntrian();
                    break;
                case 5:
                    t.tampilkanRiwayatTransaksi();
                    break;
                case 0:
                    System.out.println("Terimakasih Sudah Menggunakan Layanan Kami.");
                    break;
                default:
                    System.out.println("Pilihan tidak valid!");
            }

        }while (pilihan != 0);
    }
}
```

- Hasil Output dari Kode Program

- Menu 1

```

=== SISTEM ANTRIAN KLINIK ===
1. Tambah Pasien ke Antrian
2. Lihat Antrian
3. Layani Pasien
4. Cek Sisa Antrian Pasien
5. Lihat Riwayat Transaksi
0. Keluar
Pilih Menu: 1
Nama Pasien: Dinda
NIK: 12345
Keluhan: Sakit Kepala
>> Pasien masuk ke dalam antrian.

```

```

=== SISTEM ANTRIAN KLINIK ===
1. Tambah Pasien ke Antrian
2. Lihat Antrian
3. Layani Pasien
4. Cek Sisa Antrian Pasien
5. Lihat Riwayat Transaksi
0. Keluar
Pilih Menu: 1
Nama Pasien: Rofik
NIK: 17890
Keluhan: Nyeri dada
>> Pasien masuk ke dalam antrian.

```

```

=== SISTEM ANTRIAN KLINIK ===
1. Tambah Pasien ke Antrian
2. Lihat Antrian
3. Layani Pasien
4. Cek Sisa Antrian Pasien
5. Lihat Riwayat Transaksi
0. Keluar
Pilih Menu: 1
Nama Pasien: Nola
NIK: 23896
Keluhan: Sakit perut
>> Pasien masuk ke dalam antrian.

```

- Menu 2

```

-- Antrian Pasien --
Antrian:
Nama pasien: Dinda
NIK: 12345
Keluhan: Sakit Kepala

=====
Nama pasien: Rofik
NIK: 17890
Keluhan: Nyeri dada

=====
Nama pasien: Nola
NIK: 23896
Keluhan: Sakit perut

```

➤ Menu 3

```
Pilih Menu: 3
--- Layani Pasien ---
Pasien Dinda dipanggil
Masukkan ID Dokter: 001
Masukkan Nama Dokter: Rika
Masukkan Durasi Layanan (jam): 1
>> Pasien telah dilayani, transaksi berhasil dicatat.

=== SISTEM ANTRIAN KLINIK ===
1. Tambah Pasien ke Antrian
2. Lihat Antrian
3. Layani Pasien
4. Cek Sisa Antrian Pasien
5. Lihat Riwayat Transaksi
0. Keluar
Pilih Menu: 3
--- Layani Pasien ---
Pasien Rofik dipanggil
Masukkan ID Dokter: 002
Masukkan Nama Dokter: Farhan
Masukkan Durasi Layanan (jam): 3
>> Pasien telah dilayani, transaksi berhasil dicatat.
```

➤ Menu 4

```
=== SISTEM ANTRIAN KLINIK ===
1. Tambah Pasien ke Antrian
2. Lihat Antrian
3. Layani Pasien
4. Cek Sisa Antrian Pasien
5. Lihat Riwayat Transaksi
0. Keluar
Pilih Menu: 4
>> Sisa Pasien dalam antrian : 1
```

➤ Menu 5

```
=== SISTEM ANTRIAN KLINIK ===
1. Tambah Pasien ke Antrian
2. Lihat Antrian
3. Layani Pasien
4. Cek Sisa Antrian Pasien
5. Lihat Riwayat Transaksi
0. Keluar
Pilih Menu: 5

--- Riwayat Transaksi ---
Dinda (1 jam): Rp 50000
Rofik (3 jam): Rp 150000
Total transaksi: 2
```