POLITEKNIK NEGERI MALANG

TEKNOLOGI INFORMASI

TEKNIK INFORMATIKA



Anggota:

Achmad Nabil Afgareza (TI-1A / 3)

Muhammad Nuril Huda(TI-1A / 19)

## **Kode Program**

* Kode Program BBM

public class BBM {

    String namaBBM;

    double hargaPerLiter;

    public BBM(String namaBBM, double hargaPerLiter) {

        this.namaBBM = namaBBM;

        this.hargaPerLiter = hargaPerLiter;

    }

}

* Kode Program Kendaraan

public class Kendaraan {

    String platNomor;

    String tipe;

    String merk;

    Kendaraan(String platNomor, String tipe, String merk) {

        this.platNomor = platNomor;

        this.tipe = tipe;

        this.merk = merk;

    }

}

public class TransaksiPengisian {

    Kendaraan kendaraan;

    BBM bbm;

    double jumlahLiter;

    double totalBayar;

    public TransaksiPengisian(Kendaraan kendaraan, BBM bbm, double jumlahLiter) {

        this.kendaraan = kendaraan;

        this.bbm = bbm;

        this.jumlahLiter = jumlahLiter;

        this.totalBayar = jumlahLiter \* bbm.hargaPerLiter;

    }

}

* Kode Program TransaksiPengisian
* Kode Program Node

public class Node {

    Kendaraan data;

    Node next;

    public Node(Kendaraan data) {

        this.data = data;

        this.next = null;

    }

}

* Kode Program AntrianKendaraan

public class AntrianKendaraan {

    Node head;

    Node tail;

    int size;

    public AntrianKendaraan() {

        head = null;

        tail = null;

        size = 0;

    }

    // Menambahkan kendaraan ke antrian (Linked List Insert Di tail)

    public void tambahAntrian(Kendaraan kendaraan) {

        Node newNode = new Node(kendaraan);

        if (head == null) {

            head = tail = newNode;

        } else {

            tail.next = newNode;

            tail = newNode;

        }

        size++;

        System.out.println("Kendaraan " + kendaraan.platNomor + " ditambahkan ke antrian.");

    }

// Melayani kendaraan (hapus dari antrian di head)

    public Kendaraan layaniKendaraan() {

        if (head == null) {

            System.out.println("Antrian kosong!");

            return null;

        }

        Kendaraan dilayani = head.data;

        head = head.next;

        if (head == null) {

            tail = null;

        }

        size--;

        return dilayani;

    }

    // Menampilkan antrian (Traverse)

    public void tampilkanAntrian() {

        if (head == null) {

            System.out.println("Antrian kosong.");

            return;

        }

        System.out.println("\n=== DAFTAR ANTRIAN KENDARAAN ===");

        Node current = head;

        int urutan = 1;

        while (current != null) {

            System.out.println(urutan + ". Plat: " + current.data.platNomor + ", Tipe: " + current.data.tipe + ", Merk: " + current.data.merk);

            current = current.next;

            urutan++;

        }

    }

// Mendapatkan jumlah kendaraan dalam antrian

    public int getJumlahAntrian() {

        return size;

    }

    public boolean isEmpty() {

        return head == null;

    }

}

* Kode Program QueueTransaksi

public class QueueTransaksi {

    TransaksiPengisian[] queue;

    int front;

    int rear;

    int size;

    int kapasitas = 100;

    public QueueTransaksi() {

        this.queue = new TransaksiPengisian[kapasitas];

        this.front = 0;

        this.rear = -1;

        this.size = 0;

    }

    // Menambahkan transaksi ke queue (Enqueue)

    public void tambahTransaksi(TransaksiPengisian transaksi) {

        if (size >= kapasitas) {

            System.out.println("Queue transaksi penuh!");

            return;

        }

        rear = (rear + 1);

        queue[rear] = transaksi;

        size++;

        System.out.println("Transaksi berhasil dicatat.");

    }

* Kode Program SPBUMain

// Menampilkan riwayat transaksi

    public void tampilkanRiwayatTransaksi() {

        if (size == 0) {

            System.out.println("Belum ada transaksi.");

            return;

        }

        System.out.println("\n=== RIWAYAT TRANSAKSI BBM ===");

        for (int i = 0; i < size; i++) {

            int index = (front + i);

            TransaksiPengisian t = queue[index];

            System.out.println((i + 1) + ". Kendaraan: " + t.kendaraan.platNomor +

                             ", BBM: " + t.bbm.namaBBM +

                             ", Liter: " + t.jumlahLiter +

                             ", Total: Rp " + String.format("%.2f", t.totalBayar));

        }

    }

    public boolean isEmpty() {

        return size == 0;

    }

    public int getSize() {

        return size;

    }

}

import java.util.Scanner;

public class SPBUMain {

    AntrianKendaraan antrian;

    QueueTransaksi queueTransaksi;

public SPBUMain() {

        antrian = new AntrianKendaraan();

        queueTransaksi = new QueueTransaksi();

    }

    // Method untuk menambah kendaraan ke antrian

    public void tambahKendaraanKeAntrian(String platNomor, String tipe, String merk) {

        Kendaraan kendaraan = new Kendaraan(platNomor, tipe, merk);

        antrian.tambahAntrian(kendaraan);

    }

    // Method untuk melayani kendaraan

    public void layaniKendaraan(String namaBBM, double hargaPerLiter, double jumlahLiter) {

        if (antrian.isEmpty()) {

            System.out.println("Tidak ada kendaraan dalam antrian!");

            return;

        }

        Kendaraan kendaraanDilayani = antrian.layaniKendaraan();

        BBM bbm = new BBM(namaBBM, hargaPerLiter);

        TransaksiPengisian transaksi = new TransaksiPengisian(kendaraanDilayani, bbm, jumlahLiter);

        queueTransaksi.tambahTransaksi(transaksi);

        System.out.println(">> Transaksi berhasil dicatat");

    }

    // Method untuk menampilkan menu

    public void tampilkanMenu() {

        System.out.println("\n=== SISTEM ANTRIAN SPBU ===");

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

        System.out.println("2. Tampilkan Antrian");

        System.out.println("3. Lihat Jumlah Antrian");

        System.out.println("4. Layani Kendaraan");

        System.out.println("5. Tampilkan Riwayat Transaksi");

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

        System.out.print("Pilih menu (1-5): ");

    }

// Main

    public static void main(String[] args) {

        SPBUMain sistem = new SPBUMain();

        Scanner sc = new Scanner(System.in);

        int pilihan;

        do {

            sistem.tampilkanMenu();

            pilihan = sc.nextInt();

            sc.nextLine();

            switch (pilihan) {

                case 1:

                    System.out.print("Masukkan plat nomor: ");

                    String plat = sc.nextLine();

                    System.out.print("Masukkan jenis kendaraan: ");

                    String tipe = sc.nextLine();

                    System.out.print("Masukkan merk kendaraan: ");

                    String merk = sc.nextLine();

                    sistem.tambahKendaraanKeAntrian(plat, tipe, merk);

                    break;

                case 2:

                    sistem.antrian.tampilkanAntrian();

                    break;

                case 3:

                    System.out.println("Jumlah kendaraan dalam antrian: " +

                            sistem.antrian.getJumlahAntrian());

                    break;

case 4:

                    if (sistem.antrian.isEmpty()) {

                        System.out.println("Antrian kosong!");

                        break;

                    }

                    System.out.println("Petugas Melayani Kendaraan: " + sistem.antrian.head.data.platNomor);

                    System.out.print("Masukkan jenis BBM: ");

                    String bbm = sc.nextLine();

                    System.out.print("Masukkan harga per liter: ");

                    double harga = sc.nextDouble();

                    System.out.print("Masukkan jumlah liter: ");

                    double liter = sc.nextDouble();

                    sistem.layaniKendaraan(bbm, harga, liter);

                    break;

                case 5:

                    sistem.queueTransaksi.tampilkanRiwayatTransaksi();

                    break;

                case 0:

                    System.out.println("Terima kasih! Program selesai.");

                    break;

                default:

                    System.out.println("Pilihan tidak valid!");

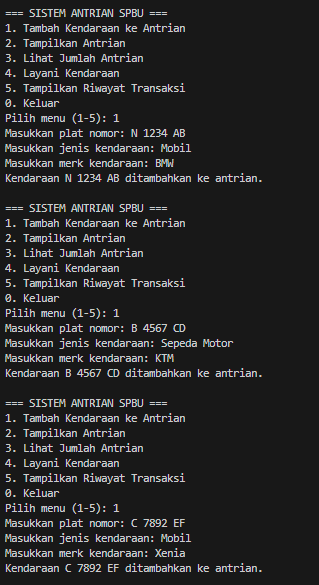
            }

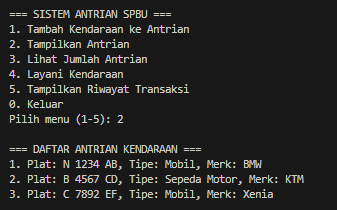
        } while (pilihan != 0);

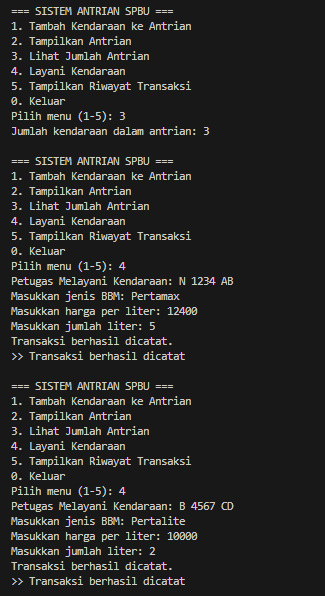
    }

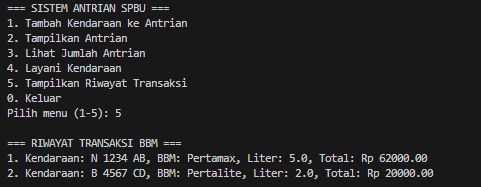
}

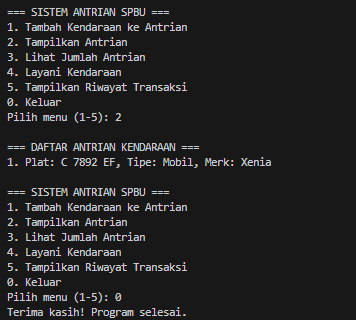
## **Hasil Kode Program**











## **Link Github:** [*https://github.com/nurilhuda05/Case-Method-ALSD*](https://github.com/nurilhuda05/Case-Method-ALSD)