

Nama : Agrisyandi Maraja Hutabarat
NIM : 254107020071
Mata Kuliah : Praktikum Dasar Pemrograman
Pertemuan ke-9

PERCOBAAN 1

```
package jobsheet7;

import java.util.Scanner;

public class SiakadFor01 {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        double nilai, tertinggi = 0, terendah = 100;
        for (int i = 1; i <= 10; i++) {
            System.out.print("Masukkan nilai mahasiswa ke-" + i + ": ");
            nilai = input.nextDouble();
            if (nilai > tertinggi) {
                tertinggi = nilai;
            }
            if (nilai < terendah) {
                terendah = nilai;
            }
        }
        System.out.println("Nilai tertinggi: " + tertinggi);
        System.out.println("Nilai terendah: " + terendah);
        input.close();
    }
}
```

Hasilnya

```
Masukkan nilai mahasiswa ke-1: 76.5
Masukkan nilai mahasiswa ke-2: 82.3
Masukkan nilai mahasiswa ke-3: 62.1
Masukkan nilai mahasiswa ke-4: 88.4
Masukkan nilai mahasiswa ke-5: 65.9
Masukkan nilai mahasiswa ke-6: 67.9
Masukkan nilai mahasiswa ke-7: 90.1
Masukkan nilai mahasiswa ke-8: 55.3
Masukkan nilai mahasiswa ke-9: 73.7
Masukkan nilai mahasiswa ke-10: 78.6
Nilai tertinggi: 90.1
Nilai terendah: 55.3
```

Pertanyaan

1. Sebutkan dan tunjukkan masing-masing komponen perulangan FOR pada kode program Percobaan 1!

Jawaban:

- `int = 1`; inisialisasi
- `i = <= 10`; syarat perulangan atau limitnya untuk terus diulang
- `i++`; sebagai peningkatan pada syarat agar kondisinya bisa berakhir increasement
- `{ ... }`; sebagai pengunci/penutup suatu program tertentu.

2. Mengapa variabel tertinggi diinisialisasi 0 dan terendah diinisialisasi 100? Apa yang terjadi jika variabel tertinggi diinisialisasi 100 dan terendah diinisialisasi 0?

Jawaban:

Program akan otomatis menganggap 100 sebagai tertinggi dan 0 sebagai terendah tanpa melihat data yang lain

3. Jelaskan fungsi dan alur kerja dari potongan kode berikut!

```
if (nilai > tertinggi) {  
    tertinggi = nilai;  
}  
if (nilai < terendah) {  
    terendah = nilai;  
}
```

Jawaban:

Jika nilai lebih besar dari “tertinggi”, maka program akan terus mencari data dengan nilai tertinggi dari semua data yang ada. Jika nilai lebih kecil dari “terendah”, maka program akan terus mencari data dengan nilai terendah dari semua data yang ada.

4. Modifikasi kode program sehingga terdapat perhitungan untuk menentukan berapa mahasiswa yang lulus dan yang tidak lulus berdasarkan batas kelulusan (nilai minimal 60). Tampilkan jumlah mahasiswa lulus dan tidak lulus setelah menampilkan nilai tertinggi dan terendah!

Jawaban:

```
package jobsheet7;

import java.util.Scanner;

public class SiakadFor01 {

    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        double nilai, tertinggi = 0, terendah = 100, nilaiMinimum = 60;
        int lulus = 0, tidakLulus = 0;

        for (int i = 1; i <= 10; i++) {
            System.out.print("Masukkan nilai mahasiswa ke-" + i + ": ");
            nilai = input.nextDouble();
            if (nilai > tertinggi) {
                tertinggi = nilai;
            }
            if (nilai < terendah) {
                terendah = nilai;
            }
            if (nilai >= nilaiMinimum) {
                lulus++;
            } else {
                tidakLulus++;
            }
        }

        System.out.println("Nilai tertinggi: " + tertinggi);
        System.out.println("Nilai terendah: " + terendah);
        System.out.println("Mahasiswa yang lulus ada : " + lulus + " orang.");
        System.out.println("Mahasiswa yang tidak lulus ada : " + tidakLulus + " orang.");
        input.close();
    }
}
```

Masukkan nilai mahasiswa ke-1: 76.5
Masukkan nilai mahasiswa ke-2: 82.3
Masukkan nilai mahasiswa ke-3: 62.1
Masukkan nilai mahasiswa ke-4: 88.4
Masukkan nilai mahasiswa ke-5: 65.9
Masukkan nilai mahasiswa ke-6: 67.9
Masukkan nilai mahasiswa ke-7: 90.1
Masukkan nilai mahasiswa ke-8: 55.3
Masukkan nilai mahasiswa ke-9: 73.7
Masukkan nilai mahasiswa ke-10: 78.6
Nilai tertinggi: 90.1
Nilai terendah: 55.3
Mahasiswa yang lulus ada : 9 orang.
Mahasiswa yang tidak lulus ada : 1 orang.

PERCOBAAN 2

```
package jobsheet7;

import java.util.Scanner;

public class SiakadWhile01 {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int nilai, jml, i = 0;
        System.out.print(s:"Masukkan jumlah mahasiswa: ");
        jml = sc.nextInt();
        while (i < jml) {
            System.out.print("Masukkan nilai mahasiswa ke-" + (i + 1) + ": ");
            nilai = sc.nextInt();

            if (nilai < 0 || nilai > 100) {
                System.out.println(x:"Nilai tidak valid, masukkan lagi nilai yang valid!");
                continue;
            }

            if (nilai > 80 && nilai <= 100) {
                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah A");
            } else if (nilai > 73 && nilai <= 80) {
                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah B+");
            } else if (nilai > 65 && nilai <= 73) {
                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah B");
            } else if (nilai > 60 && nilai <= 65) {
                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah C+");
            } else if (nilai > 50 && nilai <= 60) {
                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah C");
            } else if (nilai > 39 && nilai <= 50) {
                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah D");
            } else {
                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah E");
            }
            i++;
        }
        sc.close();
    }
}
```

Hasilnya

```
Masukkan jumlah mahasiswa: 5
Masukkan nilai mahasiswa ke-1: 85
Nilai mahasiswa ke-1 adalah A
Masukkan nilai mahasiswa ke-2: 63
Nilai mahasiswa ke-2 adalah C+
Masukkan nilai mahasiswa ke-3: 101
Nilai tidak valid, masukkan lagi nilai yang valid!
Masukkan nilai mahasiswa ke-3: 23
Nilai mahasiswa ke-3 adalah E
Masukkan nilai mahasiswa ke-4: -15
Nilai tidak valid, masukkan lagi nilai yang valid!
Masukkan nilai mahasiswa ke-4: 70
Nilai mahasiswa ke-4 adalah B
Masukkan nilai mahasiswa ke-5: 55
Nilai mahasiswa ke-5 adalah C
```

Pertanyaan

1. Pada potongan kode berikut, tentukan maksud dan kegunaan dari sintaks berikut:

```
if (nilai < 0 || nilai > 100) {  
    System.out.println(x: "Nilai tidak valid. Masukkan lagi nilai yang valid!");  
    continue;  
}
```

- a. $\text{nilai} < 0 \parallel \text{nilai} > 100$
- b. continue

Jawaban;

- a. Kondisi yang akan memeriksa apakah nilai yang dimasukkan berada dibawah 0 dan diatas 100 (diluar rentang 0-100); \parallel berarti “atau” atau “OR”
 - b. Melanjutkan perintah diluar perintah yang ada sebelumnya jika nilainya sesuai dengan kondisi yang ditetapkan, mengabaikan program yang satu bagian dengannya jika ada program yang dibuat dibawah status ini.
2. Mengapa sintaks $i++$ dituliskan di akhir perulangan WHILE? Apa yang terjadi jika posisinya dituliskan di awal perulangan WHILE?

Jawaban:

Karena nilainya harus urut, jika tidak maka hasilnya akan terlihat berantakan.

3. Apabila jumlah mahasiswa yang dimasukkan adalah 19, berapa kali perulangan WHILE akan berjalan?

Jawaban:

Jika jumlah mahasiswa yang dimasukkan adalah 19, maka perulangan akan terjadi sebanyak 19 kali.

4. Modifikasi kode program sehingga apabila terdapat mahasiswa yang mendapat nilai A, program menampilkan pesan tambahan "Bagus, pertahankan nilainya"!

Jawaban:

```

package jobsheet7;

import java.util.Scanner;

public class SiakadWhile01 {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int nilai, jml, i = 0;
        System.out.print(s:"Masukkan jumlah mahasiswa: ");
        jml = sc.nextInt();
        while (i < jml) {
            System.out.print("Masukkan nilai mahasiswa ke-" + (i + 1) + ": ");
            nilai = sc.nextInt();

            if (nilai < 0 || nilai > 100) {
                System.out.println(x:"Nilai tidak valid, masukkan lagi nilai yang valid!");
                continue;
            }

            if (nilai > 80 && nilai <= 100) {
                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah A");
                System.out.println(x:"Bagus, pertahankan nilainya!");
            } else if (nilai > 73 && nilai <= 80) {
                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah B+");
            } else if (nilai > 65 && nilai <= 73) {
                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah B");
            } else if (nilai > 60 && nilai <= 65) {
                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah C+");
            } else if (nilai > 50 && nilai <= 60) {
                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah C");
            } else if (nilai > 39 && nilai <= 50) {
                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah D");
            } else {
                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah E");
            }
            i++;
        }
        sc.close();
    }
}

```

Hasilnya

```

Masukkan jumlah mahasiswa: 5
Masukkan nilai mahasiswa ke-1: 85
Nilai mahasiswa ke-1 adalah A
Bagus, pertahankan nilainya!
Masukkan nilai mahasiswa ke-2: 63
Nilai mahasiswa ke-2 adalah C+
Masukkan nilai mahasiswa ke-3: 101
Nilai tidak valid, masukkan lagi nilai yang valid!
Masukkan nilai mahasiswa ke-3: 23
Nilai mahasiswa ke-3 adalah E
Masukkan nilai mahasiswa ke-4: -15
Nilai tidak valid, masukkan lagi nilai yang valid!
Masukkan nilai mahasiswa ke-4: 70
Nilai mahasiswa ke-4 adalah B
Masukkan nilai mahasiswa ke-5: 55
Nilai mahasiswa ke-5 adalah C

```


PERCOBAAN 3

```
package jobsheet7;

import java.util.Scanner;

public class KafeDowWhile01 {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int kopi, teh, roti;
        String namaPelanggan;
        int hargaKopi = 12000;
        int hargaTeh = 7000;
        int hargaRoti = 20000;
        do {
            System.out.print(s:"Masukkan nama pelanggan (ketik 'batal' untuk keluar): ");
            namaPelanggan = sc.nextLine();
            if (namaPelanggan.equalsIgnoreCase(anotherString:"batal")) {
                System.out.println(x:"Transaksi dibatalkan.");
                break;
            }
            System.out.print(s:"Jumlah kopi: ");
            kopi = sc.nextInt();
            System.out.print(s:"Jumlah teh: ");
            teh = sc.nextInt();
            System.out.print(s:"Jumlah roti: ");
            roti = sc.nextInt();

            int totalHarga = (kopi * hargaKopi) + (teh * hargaTeh) + (roti * hargaRoti);
            System.out.println("Total yang harus dibayar: Rp " + totalHarga);
            sc.nextLine();
        } while (true);
        System.out.println(x:"Semua transaksi selesai.");
        sc.close();
    }
}
```

Hasilnya

```
Masukkan nama pelanggan (ketik 'batal' untuk keluar): Shan
Jumlah kopi: 3
Jumlah teh: 0
Jumlah roti: 1
Total yang harus dibayar: Rp 56000
Masukkan nama pelanggan (ketik 'batal' untuk keluar): Yudi
Jumlah kopi: 1
Jumlah teh: 4
Jumlah roti: 2
Total yang harus dibayar: Rp 80000
Masukkan nama pelanggan (ketik 'batal' untuk keluar): BATAL
Transaksi dibatalkan.
Semua transaksi selesai.
```

Pertanyaan

1. Pada penggunaan DO-WHILE ini, apabila nama pelanggan yang dimasukkan pertama kali adalah “batal”, maka berapa kali perulangan dilakukan?

Jawaban:

Tidak ada pengulangan yang dilakukan, karena “batal” adalah syarat untuk keluar dari kondisi loop tersebut.

2. Sebutkan kondisi berhenti yang digunakan pada perulangan DO-WHILE tersebut!

Jawaban:

Kondisi yang digunakan adalah “break”

3. Apa fungsi dari penggunaan nilai true pada kondisi DO-WHILE?

Jawaban:

Agar program dapat terus diulang setelah di-run, hingga syarat penghentian perulangannya dipenuhi.

4. Mengapa perulangan DO-WHILE tersebut tetap berjalan meskipun tidak ada komponen inisialisasi dan update?

Jawaban:

Karena kondisi loop pada DO-WHILE tidak bergantung pada komponen inisialisasi dan update.

TUGAS

1. Seorang pengelola bioskop ingin membuat program untuk menghitung total penjualan tiket dalam satu hari. Tiket dijual dengan harga Rp 50.000 per tiket. Program harus menghitung total tiket yang terjual dan total harga penjualan tiket selama satu hari dengan ketentuan sebagai berikut:
 - Jika pelanggan membeli lebih dari 4 tiket, pelanggan mendapatkan diskon 10%.
 - Jika pelanggan membeli lebih dari 10 tiket, pelanggan mendapatkan diskon 15%.
 - Jika input jumlah tiket tidak valid (negatif), program akan mengabaikan input tersebut dan meminta input ulang.

Commit dan push program Anda ke Github dengan pesan "Tugas 1"

```
import java.util.Scanner;

public class TiketBioskop {

    Run | Debug | Run main | Debug main
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int hargaTiket = 50000;
        double totalPenjualanTiketPerHari = 0.0;

        int totalTiketHarian = 0;

        System.out.println(x:"Selamat datang!!");
        System.out.println(x:"Masukkan jumlah tiket per transaksi.");

        do {

            System.out.print(s:"\nMasukkan jumlah tiket: ");
            int jumlahTiket = sc.nextInt();

            if (jumlahTiket == 0) {
                System.out.println(x:"Kasir ditutup.");
                break;
            }

            if (jumlahTiket < 0) {
                System.out.println(x:"Jumlah tiket tidak valid, silakan coba lagi.");
                continue;
            }

            double diskon;
            if (jumlahTiket > 4 && jumlahTiket <= 10) {
                diskon = 0.10;
            } else if (jumlahTiket > 10) {
                diskon = 0.15;
            } else {
                diskon = 0.0;
            }

            double hargaAwal = jumlahTiket * hargaTiket;
            double potonganDiskon = hargaAwal * diskon;
            double hargaAkhir = hargaAwal - potonganDiskon;

            System.out.println("Harga Tiket: Rp " + hargaAwal);
            System.out.println("Diskon: Rp " + potonganDiskon);
            System.out.println("Total yang harus dibayar: Rp " + hargaAkhir);

            totalTiketHarian = totalTiketHarian + jumlahTiket;
            totalPenjualanTiketPerHari = totalPenjualanTiketPerHari + hargaAkhir;

        } while (true);

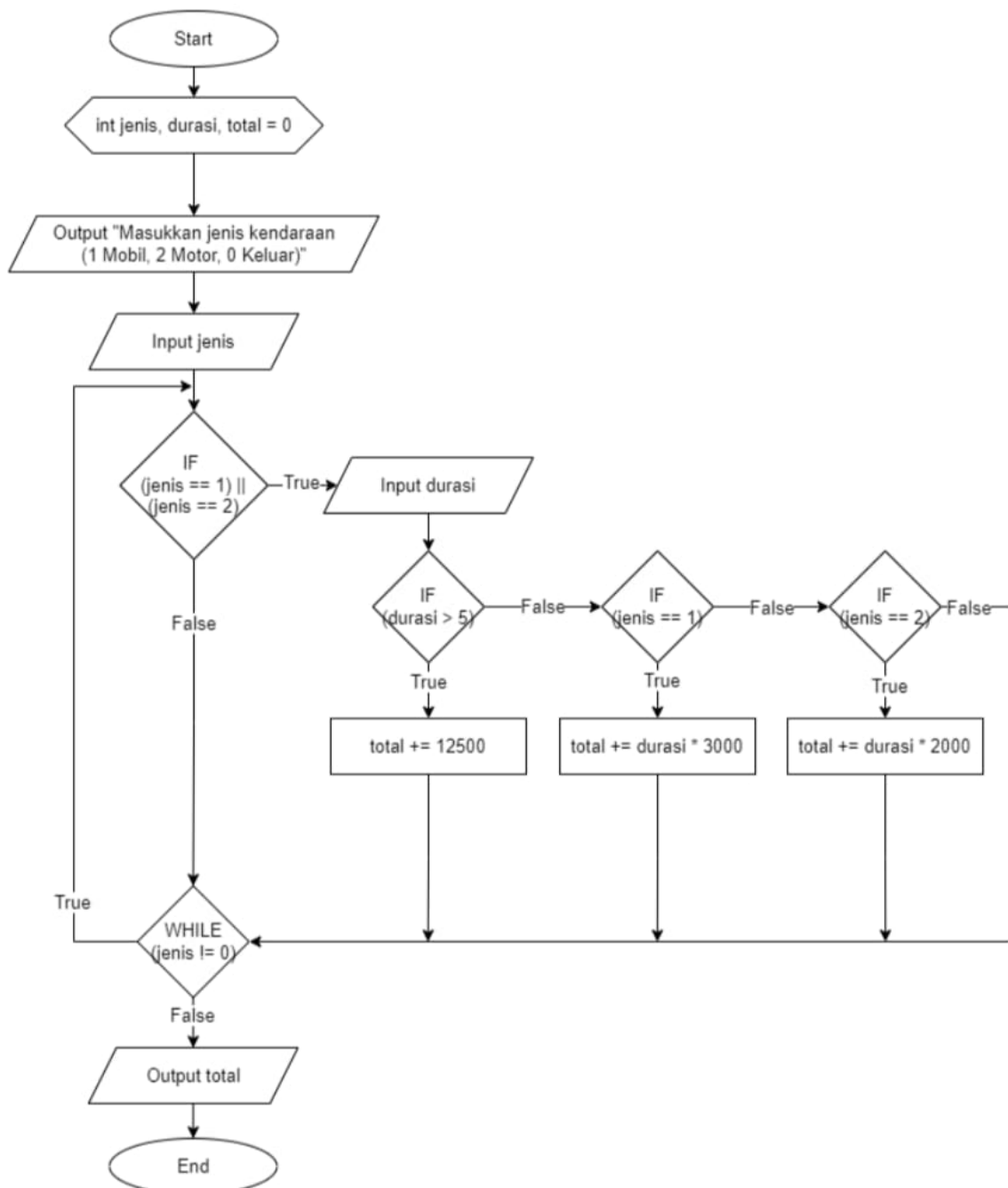
        System.out.println("\nTotal Tiket Terjual: " + totalTiketHarian + " tiket");
        System.out.println("Total Penjualan Harian: Rp " + totalPenjualanTiketPerHari);

    }
}
```

Hasilnya

```
Selamat datang!!  
Masukkan jumlah tiket per transaksi.  
  
Masukkan jumlah tiket: 5  
Harga Tiket: Rp 250000.0  
Diskon: Rp 25000.0  
Total yang harus dibayar: Rp 225000.0  
  
Masukkan jumlah tiket: 12  
Harga Tiket: Rp 600000.0  
Diskon: Rp 90000.0  
Total yang harus dibayar: Rp 510000.0  
  
Masukkan jumlah tiket: 3  
Harga Tiket: Rp 150000.0  
Diskon: Rp 0.0  
Total yang harus dibayar: Rp 150000.0  
  
Masukkan jumlah tiket: 0  
Kasir ditutup.  
  
Total Tiket Terjual: 20 tiket  
Total Penjualan Harian: Rp 885000.0
```

2. Perhatikan flowchart berikut!



Sebuah tempat parkir ingin membuat program untuk menghitung total pembayaran parkir dari beberapa kendaraan. Tarif parkir adalah Rp 3.000 per jam untuk mobil dan Rp 2.000 per jam untuk motor. Namun, jika durasi parkir lebih dari 5 jam, diberikan tarif tetap sebesar Rp 12.500 untuk semua kendaraan. Program akan terus meminta masukan selama input bukan 0. Implementasikan flowchart tersebut ke dalam bentuk kode program Java! Commit dan push program Anda ke Github dengan pesan “Tugas 2”

Jawaban:

```
import java.util.Scanner;

public class TempatParkir {

    Run | Debug | Run main | Debug main
    public static void main(String[] args) {

        int jenis;
        int durasi;
        int total = 0;

        Scanner sc = new Scanner(System.in);

        do {

            System.out.print(s:"Masukkan jenis kendaraan (1 Mobil, 2 Motor, 0 Keluar): ");

            jenis = sc.nextInt();

            if (jenis == 1 || jenis == 2) {

                System.out.print(s:"Masukkan durasi parkir (jam): ");
                durasi = sc.nextInt();

                if (durasi > 5) {
                    total += 12500;
                    System.out.println(x:"Biaya anda adalah Rp 12.500 (Tarif tetap untuk parkir lebih dari 5 jam)");
                } else {
                    switch (jenis) {
                        case 1:
                            int biaya = durasi * 3000;
                            total += biaya;
                            System.out.println("Biaya: Rp " + biaya);
                            break;
                        case 2:
                            biaya = durasi * 2000;
                            total += biaya;
                            System.out.println("Biaya: Rp " + biaya);
                            break;
                    }
                }
            }

        } while (jenis != 0);

        System.out.println("Total pendapatan parkir: Rp " + total);
    }
}
```

Hasilnya

```
Masukkan jenis kendaraan (1 Mobil, 2 Motor, 0 Keluar): 2
Masukkan durasi parkir (jam): 4
Biaya: Rp 8000
Masukkan jenis kendaraan (1 Mobil, 2 Motor, 0 Keluar): 2
Masukkan durasi parkir (jam): 6
Biaya anda adalah Rp 12.500 (Tarif tetap untuk parkir lebih dari 5 jam)
Masukkan jenis kendaraan (1 Mobil, 2 Motor, 0 Keluar): 1
Masukkan durasi parkir (jam): 2
Biaya: Rp 6000
Masukkan jenis kendaraan (1 Mobil, 2 Motor, 0 Keluar): 1
Masukkan durasi parkir (jam): 6
Biaya anda adalah Rp 12.500 (Tarif tetap untuk parkir lebih dari 5 jam)
Masukkan jenis kendaraan (1 Mobil, 2 Motor, 0 Keluar): 0
Total pendapatan parkir: Rp 39000
```