

LAPORAN PRAKTIKUM

(JOBSHEET 7)



Disusun Oleh:

Nama : Masando Fami Ramadhan
NIM : 254107060011
Kelas : 1B
No. Absen : 14

PROGRAM STUDI SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2025

Daftar Isi

1. Tujuan	3
2. Praktikum	3
2.1. Percobaan 1: Studi Kasus Nilai Mahasiswa di SIAKAD - Perulangan FOR .	3
2.2. Percobaan 2: Studi Kasus Nilai Mahasiswa - Perulangan WHILE	10
2.3. Percobaan 3: Studi Kasus Transaksi di Kafe - Perulangan DO-WHILE	17
3. Tugas	21
3.1. Tugas 1	21
3.2. Tugas 2	25

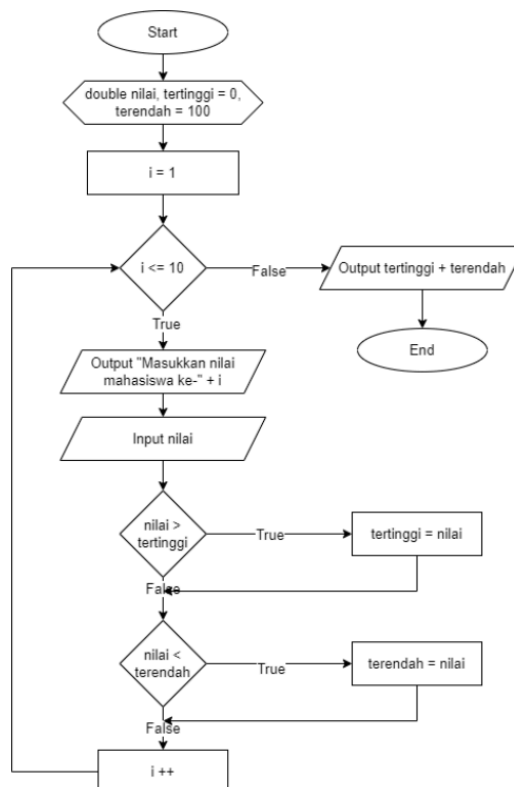
1. Tujuan

- Mahasiswa dapat menjelaskan format penulisan program perulangan (for, while, dan do-while)
- Mahasiswa dapat mengimplementasikan flowchart perulangan menggunakan bahasa pemrograman Java

2. Praktikum

2.1. Percobaan 1: Studi Kasus Nilai Mahasiswa di SIAKAD - Perulangan FOR

Di dalam Sistem Informasi Akademik (SIAKAD), dosen mengisi nilai mata kuliah Praktikum Dasar Pemrograman yang ditempuh oleh mahasiswa. Dosen tersebut ingin mencari nilai tertinggi dan terendah Kuis dari 10 mahasiswa di dalam satu kelas. Dosen tersebut harus memasukkan nilai dari setiap siswa, kemudian menentukan dan menampilkan nilai tertinggi dan terendah. Perhatikan flowchart berikut ini:



Buatlah program java berdasarkan studi kasus tersebut!

1. Masih dengan repo yang sama, buat folder baru bernamakan "Jobsheet7". Buat file baru di dalam folder tersebut dengan nama **SiakadFor{NoAbsen}.java**.
2. Tambahkan kode berikut:

```
import java.util.Scanner;

public class SiakadFor14 {
    public static void main(String[] args) {
        Scanner sc = 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 = sc.nextDouble();
            if (nilai > tertinggi) {
                tertinggi = nilai;
            }
            if (nilai < terendah) {
                terendah = nilai;
            }
        }

        System.out.println("Nilai tertinggi: " + tertinggi);
        System.out.println("Nilai terendah: " + terendah);

        sc.close();
    }
}
```

3. *Compile* dan *run* program tersebut.

```
MyBook Z Series@FAMI-DESKTOP MINGW64
raktikum_Daspro/Jobsheet_7 (main)
❌ $ java -s -r SiakadFor14.java
Masukkan nilai mahasiswa ke-1: 76
Masukkan nilai mahasiswa ke-2: 82
Masukkan nilai mahasiswa ke-3: 87
Masukkan nilai mahasiswa ke-4: 76
Masukkan nilai mahasiswa ke-5: 85
Masukkan nilai mahasiswa ke-6: 79
Masukkan nilai mahasiswa ke-7: 81
Masukkan nilai mahasiswa ke-8: 90
Masukkan nilai mahasiswa ke-9: 68
Masukkan nilai mahasiswa ke-10: 68
Nilai tertinggi: 90.0
Nilai terendah: 68.0
```

Pertanyaan

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

- Dalam kode tersebut, sintaks perulangan for dituliskan sebagai berikut:

```
for (int i = 1; i <= 10; i++) {  
    ...  
}
```

for adalah keyword untuk sintaks perulangan for di java. Sedangkan **int i** adalah inisialisasi variabel counter. Pada kode tersebut, nilai **i** pada iterasi pertama akan dimulai dari **1**. **1 <= 10** adalah kondisi sebagai batas/syarat perulangan tetap dieksekusi. Dalam hal ini, nilai **i** harus lebih kecil dari/sama dengan **10**. Terakhir **i++** adalah sintaks *increment* untuk mengupdate nilai variabel counter setiap iterasi.

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

- Variabel **tertinggi** diinisialisasi dengan **0** (kemungkinan nilai paling rendah) dan **terendah** diinisialisasi dengan **100** (kemungkinan nilai paling tinggi) untuk memastikan bahwa nilai pertama yang dimasukkan oleh pengguna akan segera menggantikan nilai inisialisasi tersebut, dan proses perbandingan nilai akan berjalan dengan benar untuk seluruh data.

Jika nilai inisialisasi tersebut dibalik, program akan menghasilkan output yang salah karena nilai-nilai yang dimasukkan tidak akan pernah bisa menggantikan nilai inisialisasi. Dengan kata lain, nilai **tertinggi** akan selalu menjadi **100**, sedangkan nilai **terendah** akan selalu menjadi **0**.

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

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

- Bagian kode tersebut akan membandingkan nilai sekarang (dalam iterasi) dan nilai tertinggi/terendah terakhir apakah nilai tersebut lebih besar/kecil dari nilai terakhir atau tidak. Jika iya, maka update nilai tertinggi/terendah menjadi nilai pada iterasi itu. Dengan mengulangnya sampai iterasi terakhir, kita akan mendapatkan nilai tertinggi dan terendah yang benar dari nilai-nilai tersebut.
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!

```

import java.util.Scanner;

public class SiakadFor14 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        double nilai, tertinggi = 0, terendah = 100;
        int lulus = 0, tidakLulus = 0;

        for (int i = 1; i <= 10; i++) {
            System.out.print("Masukkan nilai mahasiswa ke-" +
i + ": ");
            nilai = sc.nextDouble();

            if (nilai > tertinggi) {
                tertinggi = nilai;
            }

            if (nilai < terendah) {
                terendah = nilai;
            }

            if (nilai >= 60.0) {
                lulus++;
            } else {
                tidakLulus++;
            }
        }

        System.out.println("\n--- Hasil Analisis Nilai ---");
        System.out.println("Nilai tertinggi: " + tertinggi);
        System.out.println("Nilai terendah: " + terendah);
        System.out.println("-----");
        System.out.println("LULUS: " + lulus);
        System.out.println("TIDAK LULUS: " + tidakLulus);

        sc.close();
    }
}

```



```
- $ java SiakadFor14.java

  JAWA, JAWA, JAWA!

✓ Compilation successful.

> Running SiakadFor14...

Masukkan nilai mahasiswa ke-1: 78
Masukkan nilai mahasiswa ke-2: 82
Masukkan nilai mahasiswa ke-3: 56
Masukkan nilai mahasiswa ke-4: 45
Masukkan nilai mahasiswa ke-5: 78
Masukkan nilai mahasiswa ke-6: 75
Masukkan nilai mahasiswa ke-7: 61
Masukkan nilai mahasiswa ke-8: 87
Masukkan nilai mahasiswa ke-9: 59
Masukkan nilai mahasiswa ke-10: 25

--- Hasil Analisis Nilai ---
Nilai tertinggi: 87.0
Nilai terendah: 25.0
-----
Jumlah mahasiswa LULUS: 6
Jumlah mahasiswa TIDAK LULUS: 4
```

5. *Commit* dan *push* kode program ke Github

```
- feat: jobsheet 7 - percobaan 1
  fami0110 committed 2 minutes ago
```

2.2. Percobaan 2: Studi Kasus Nilai Mahasiswa - Perulangan WHILE

Seorang dosen ingin memasukkan nilai beberapa mahasiswa ke dalam SIAKAD untuk ditentukan kategori nilai hurufnya. Program harus meminta dosen untuk memasukkan nilai setiap mahasiswa. Jika dosen memasukkan nilai yang tidak valid (negatif atau lebih dari 100), program harus mengabaikan input tersebut dan meminta dosen untuk melakukan input ulang. Selanjutnya, nilai yang valid dikelompokkan ke dalam kategori huruf A ($80 < \text{nilai} \leq 100$), B+ ($73 < \text{nilai} \leq 80$), B ($65 < \text{nilai} \leq 73$), C+ ($60 < \text{nilai} \leq 65$), C ($50 < \text{nilai} \leq 60$), D ($39 < \text{nilai} \leq 50$), dan E ($\text{nilai} \leq 39$).

Berdasarkan studi kasus tersebut, buat program menggunakan bahasa pemrograman Java!

1. Masih dengan repo dan folder yang sama, buat file baru di dalam folder tersebut dengan nama `SiakadWhile{NoAbsen}.java`.
2. Tambahkan kode berikut:

```
import java.util.Scanner;

public class SiakadWhile14 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int nilai, jml, i = 0;

        System.out.print("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("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();
}
}

```

3. *Compile* dan *run* program tersebut.

```
$ java SiakadWhile14.java
```

 **JAWA, JAWA, JAWA!** 

✓ Compilation successful.

> Running SiakadWhile14...

```
Masukkan jumlah mahasiswa: 3
Masukkan nilai mahasiswa ke-1: -1
Nilai tidak valid. Masukkan lagi nilai yang valid!
Masukkan nilai mahasiswa ke-1: -1
Nilai tidak valid. Masukkan lagi nilai yang valid!
Masukkan nilai mahasiswa ke-1: 80
Nilai mahasiswa ke-1 adalah B+
Masukkan nilai mahasiswa ke-2: 67
Nilai mahasiswa ke-2 adalah B
Masukkan nilai mahasiswa ke-3: 90
Nilai mahasiswa ke-3 adalah A
```

Pertanyaan

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

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

- Sesuai dengan perintah pada studi kasus, bagian kode tersebut berfungsi untuk memvalidasi nilai yang dimasukkan oleh pengguna. Jika nilai yang dimasukkan tidak valid, maka variabel counter **i** tidak akan ter-update (increment), dan program akan meminta pengguna untuk memasukkan nilai lagi.
2. Mengapa sintaks **i++** dituliskan di akhir perulangan WHILE? Apa yang terjadi jika posisinya dituliskan di awal perulangan WHILE?
 - Sintaks tersebut diletakkan di akhir perulangan while karena ia berfungsi sebagai updater untuk variabel counter yang memastikan perulangan tersebut berhenti setelah jumlah iterasi yang diinginkan tercapai.

Apabila sintaks tersebut diletakkan di awal perulangan while, maka yang terjadi adalah ada iterasi yang hilang yakni mahasiswa ke-1. Selain itu, apabila pengguna pada iterasi tersebut memasukkan nilai yang salah, alih-alih memasukkan ulang mahasiswa pada iterasi itu, program malah menyuruh kita untuk memasukkan nilai mahasiswa selanjutnya. Dengan kata lain, proses validasi akan menjadi tidak berguna apabila **i++** ditaruh di awal perulangan while
 3. Apabila jumlah mahasiswa yang dimasukkan adalah **19**, berapa kali perulangan WHILE akan berjalan?
 - Perulangan while akan dijalankan 19 kali, dimulai dari 0-18.

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

```
import java.util.Scanner;

public class SiakadWhile14 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int nilai, jml, i = 0;

        System.out.print("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("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("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();
}
}

```

- \$ java SiakadWhile14.java

 JAWA, JAWA, JAWA! 

✓ Compilation successful.

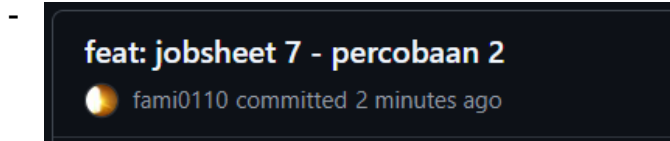
> Running SiakadWhile14...

```

Masukkan jumlah mahasiswa: 3
Masukkan nilai mahasiswa ke-1: 97
Nilai mahasiswa ke-1 adalah A
Bagus, pertahankan nilainya!
Masukkan nilai mahasiswa ke-2: 1
Nilai mahasiswa ke-2 adalah E
Masukkan nilai mahasiswa ke-3: 39
Nilai mahasiswa ke-3 adalah E

```

5. *Commit* dan *push* kode program ke Github



2.3. Percobaan 3: Studi Kasus Transaksi di Kafe - Perulangan DO-WHILE

Di sebuah kafe, kasir ingin memproses transaksi beberapa pelanggan. Pelanggan dapat membeli lebih dari satu item (kopi dengan harga Rp 12.000, teh dengan harga Rp 7.000, dan roti dengan harga Rp 20.000), dan kasir akan terus memasukkan jumlah pembelian untuk setiap pelanggan. Jika ada pelanggan yang memutuskan untuk membatalkan transaksi (dengan memasukkan "batal"), maka kasir akan menghentikan input transaksi dan program berhenti.

Berdasarkan studi kasus tersebut, buat program menggunakan bahasa pemrograman Java!

1. Masih dengan repo dan folder yang sama, buat file baru di dalam folder tersebut dengan nama `SiakadDoWhile{NoAbsen}.java`.
2. Tambahkan kode berikut:

```
import java.util.Scanner;

public class KafeDoWhile14 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int kopi, teh, roti;
        String namaPelanggan;

        final int hargaKopi = 12000;
        final int hargaTeh = 7000;
        final int hargaRoti = 20000;
        long totalHarga;

        do {
            System.out.print("Masukkan nama pelanggan (ketik 'batal' untuk keluar): ");
            namaPelanggan = sc.next();

            if (namaPelanggan.equalsIgnoreCase("batal")) {
                System.out.println("Transaksi dibatalkan.");
                break;
            }
        }
```

```

    }

    System.out.print("Jumlah kopi: ");
    kopi = sc.nextInt();
    System.out.print("Jumlah teh: ");
    teh = sc.nextInt();
    System.out.print("Jumlah roti: ");
    roti = sc.nextInt();

    totalHarga = (long)kopi * hargaKopi + (long)teh *
hargaTeh + (long)roti * hargaRoti;
    System.out.println("Total yang harus dibayar: Rp
" + totalHarga);
    sc.nextLine();
} while (true);

System.out.println("Semua transaksi selesai.");
sc.close();
}
}

```

3. *Compile* dan *run* program tersebut.

```

$ java KafeDoWhile14.java

    🍵 JAWA, JAWA, JAWA! 🍵

✅ Compilation successful.

> Running KafeDoWhile14...

Masukkan nama pelanggan (ketik 'batal' untuk keluar):sando
Jumlah kopi: 1
Jumlah teh: 1
Jumlah roti: 2
Total yang harus dibayar: Rp 59000
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?
 - Perulangan akan dilakukan **satu kali**. Karena seperti sifat DO-WHILE “lakukan dulu, baru cek”. Sehingga ketika pelanggan memasukkan “batal” pada iterasi pertama, maka iterasi kedua tidak akan dijalankan.
2. Sebutkan kondisi berhenti yang digunakan pada perulangan DO-WHILE tersebut!
 - Kondisi berakhir pada perulangan tersebut (saat pengguna memasukkan “batal”) terdapat pada bagian kode berikut:

```
if (namaPelanggan.equalsIgnoreCase("batal")) {  
    System.out.println("Transaksi dibatalkan.");  
    break;  
}
```

3. Apa fungsi dari penggunaan nilai **true** pada kondisi DO-WHILE?
 - Karena pada perulangan tersebut kondisi di-set **true**, maka perulangan akan dijalankan selamanya, **SELAMA** pengguna tidak memasukkan “batal” pada saat input nama pelanggan.
4. Mengapa perulangan DO-WHILE tersebut tetap berjalan meskipun tidak ada komponen inisialisasi dan update?
 - Perulangan tersebut tetap berjalan karena komponen inisialisasi dan update yang mengatur logikanya berada di dalam perulangan itu sendiri. Oleh karena itu, yang penting kondisi perulangannya telah diatur secara eksplisit menjadi **true** dan kondisi dan logika kapan harus berhenti sudah ada di dalam perulangan (dalam hal ini ketika pengguna memasukkan nilai “batal”). Kalau tidak ada kondisi berhenti, maka program akan berjalan terus-menerus.

5. *Commit* dan *push* kode program ke Github

-

feat: jobsheet 7 - percobaan 3



fami0110 committed now

3. Tugas

3.1. 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 > 4 tiket, pelanggan mendapatkan diskon 10%.
- Jika pelanggan membeli > 10 tiket, pelanggan mendapatkan diskon 15%.
- Jika input jumlah tiket tidak valid (negatif), program akan mengabaikan input tersebut dan meminta input ulang.

Catatan: Perulangan dapat menggunakan for, while, atau do-while. Penambahan break atau continue jika diperlukan.

```

import java.util.Scanner;

public class Tugas1 {
    public static void main(String[] args) {
        // Deklarasi variabel
        Scanner sc = new Scanner(System.in);
        int hargaKotor = 0, totalTiketTerjual = 0;
        double diskon = 0, totalBayar = 0, totalPenjualanBersih
= 0;

        // Input jumlah pelanggan
        System.out.print("Masukkan total pelanggan hari ini: ");
        int jumlahPelanggan = sc.nextInt();

        int i = 0;

        while (i < jumlahPelanggan) {
            System.out.println("\n--- Pelanggan ke-" + (i + 1) +
" ---");
            System.out.print("Masukkan jumlah tiket yang dibeli:
");
            int jumlahTiket = sc.nextInt();

            // Validasi input
            if (jumlahTiket < 0) {
                System.out.println("✗ Jumlah tiket tidak
valid. Silakan masukkan ulang.");
                continue;
            }

            hargaKotor = jumlahTiket * 50000;

            // Hitung Diskon
            if (jumlahTiket > 10) {
                diskon = 0.15;
            } else if (jumlahTiket > 4) {
                diskon = 0.10;
            }

```

```

        // Hitung total harga setelah diskon
        totalBayar = (double) hargaKotor - ((double)
hargaKotor * diskon);

        // Update total keseluruhan
        totalTiketTerjual += jumlahTiket;
        totalPenjualanBersih += totalBayar;

        System.out.printf("Harga Kotor: Rp %d\n",
hargaKotor);
        System.out.printf("Diskon: %.0f%%\n", (diskon *
100));
        System.out.printf("Total Bayar: Rp %.0f\n",
totalBayar);

        // Update counter
        i++;
    }

    // Tampilkan hasil akhir
    System.out.println("\n===== LAPORAN PENJUALAN =====");
    System.out.println("Total Tiket Terjual: " +
totalTiketTerjual + " tiket");
    System.out.printf("Pendapatan Bersih: Rp %.2f\n",
totalPenjualanBersih);
    System.out.println("=====");

    sc.close();
}
}

```

```
$ java Tugas1.java
```

```
☕ JAWA, JAWA, JAWA! ☕
```

```
✅ Compilation successful.
```

```
> Running Tugas1...
```

```
Masukkan total pelanggan hari ini: 3
```

```
--- Pelanggan ke-1 ---
```

```
Masukkan jumlah tiket yang dibeli: 3
```

```
Harga Kotor: Rp 150000
```

```
Diskon: 0%
```

```
Total Bayar: Rp 150000
```

```
--- Pelanggan ke-2 ---
```

```
Masukkan jumlah tiket yang dibeli: 5
```

```
Harga Kotor: Rp 250000
```

```
Diskon: 10%
```

```
Total Bayar: Rp 225000
```

```
--- Pelanggan ke-3 ---
```

```
Masukkan jumlah tiket yang dibeli: 11
```

```
Harga Kotor: Rp 550000
```

```
Diskon: 15%
```

```
Total Bayar: Rp 467500
```

```
===== LAPORAN PENJUALAN =====
```

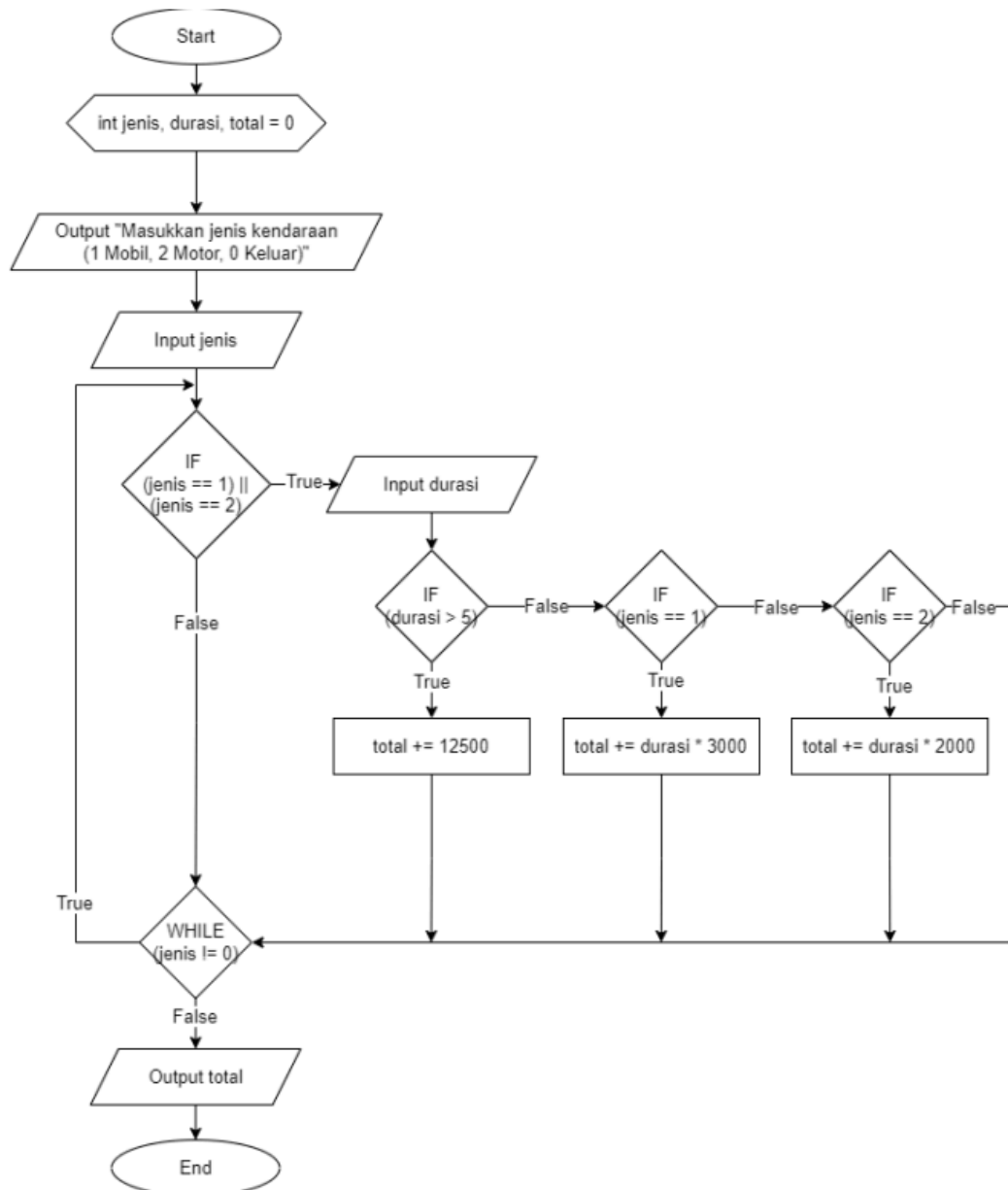
```
Total Tiket Terjual: 19 tiket
```

```
Pendapatan Bersih: Rp 842500.00
```

```
=====
```


3.2. Tugas 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!

```

import java.util.Scanner;

public class Tugas2 {
    public static void main(String[] args) {

        // Deklarasi variabel
        Scanner sc = new Scanner(System.in);
        int jenis, durasi, total = 0;

        // Masuk ke perulangan
        do {
            System.out.println("Masukkan jenis kendaraan:");
            System.out.println("1. Mobil");
            System.out.println("2. Motor");
            System.out.println("0. Keluar");
            jenis = sc.nextInt();

            if (jenis == 1 || jenis == 2) {
                System.out.print("Masukkan durasi parkir (jam):");
                durasi = sc.nextInt();

                // Pemilihan biaya parkir berdasarkan durasi
                // atau jenis kendaraan
                if (durasi > 5) {
                    total += 12500;
                } else {
                    if (jenis == 1) {
                        total += durasi * 3000;
                    } else if (jenis == 2) {
                        total += durasi * 2000;
                    }
                }
            } else if (jenis != 0) {
                System.out.println("Jenis kendaraan tidak valid.");
            }

        } while (jenis != 0);
    }
}

```

```
        System.out.println("Total biaya parkir hari ini: " +  
total);  
  
        sc.close();  
    }  
}
```

```
$ java Tugas2.java
```

☕ JAWA, JAWA, JAWA! ☕

✓ Compilation successful.

> Running Tugas2...

Masukkan jenis kendaraan:

- 1. Mobil
- 2. Motor
- 0. Keluar

1

Masukkan durasi parkir (jam): 2

Masukkan jenis kendaraan:

- 1. Mobil
- 2. Motor
- 0. Keluar

2

Masukkan durasi parkir (jam): 3

Masukkan jenis kendaraan:

- 1. Mobil
- 2. Motor
- 0. Keluar

0

Total biaya parkir hari ini: 12000

Link Praktikum: https://github.com/fami0110/Praktikum_Daspro