

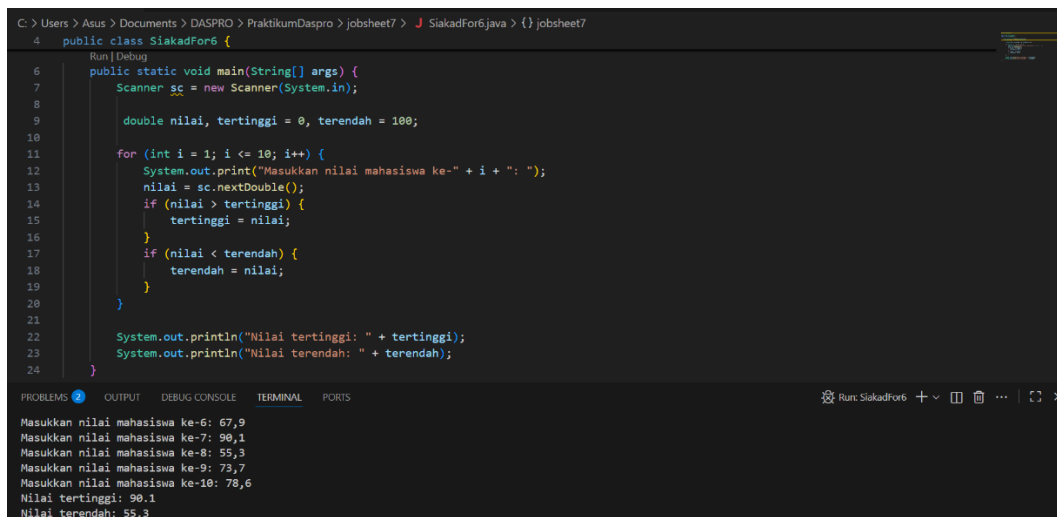
Nama : Deswita Khansa Rafifah

NIM : 254107020151

Kelas : TI\_1G

## Jobsheet 7

### Percobaan 1



```
C:\Users> Asus > Documents > DASPRO > PraktikumDaspro > jobsheet7 > J SiakadFor6.java > {} jobsheet7
4 public class SiakadFor6 {
    Run | Debug
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8
9         double nilai, tertinggi = 0, terendah = 100;
10
11        for (int i = 1; i <= 10; i++) {
12            System.out.print("Masukkan nilai mahasiswa ke-" + i + ": ");
13            nilai = sc.nextDouble();
14            if (nilai > tertinggi) {
15                tertinggi = nilai;
16            }
17            if (nilai < terendah) {
18                terendah = nilai;
19            }
20        }
21
22        System.out.println("Nilai tertinggi: " + tertinggi);
23        System.out.println("Nilai terendah: " + terendah);
24    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

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!

**Jawab:**

Pada perulangan FOR terdapat 3 komponen:

1. Inisialisasi → deklarasi dan inisialisasi variabel counter (variabel pengontrol perulangan).  
→ `int i = 1`
  2. Kondisi → batas atau syarat agar perulangan tetap dieksekusi.  
→ `i <= 10`
  3. Update → Perubahan nilai variabel counter pada setiap putaran perulangan (increment atau decrement)  
→ `i++`
2. Mengapa variabel **tertinggi** diinisialisasi 0 dan **terendah** diinisialisasi 100? Apa yang terjadi jika variabel tertinggi diinisialisasi 100 dan terendah diinisialisasi 0?

**Jawab:**

Variabel tertinggi diinisialisasi dengan 0 karena ingin mencari nilai maksimum. Dengan memulai dari nilai paling kecil (0), setiap nilai yang diinput akan dibandingkan dan nilai yang lebih besar akan menggantikan nilai tertinggi. Variabel terendah diinisialisasi dengan 100 karena ingin mencari nilai minimum. Dengan memulai dari nilai paling besar (100, asumsi nilai maksimal), setiap nilai

yang diinput akan dibandingkan dan nilai yang lebih kecil akan menggantikan nilai terendah.

Jika dibalik (tertinggi = 100 dan terendah = 0), maka:

- Nilai tertinggi tidak akan pernah diperbarui jika semua nilai input lebih kecil dari 100, sehingga hasil akhir tertinggi akan tetap 100 (salah).
- Nilai terendah tidak akan pernah diperbarui jika semua nilai input lebih besar dari 0, sehingga hasil akhir terendah akan tetap 0 (salah).

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

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

**Jawab:**

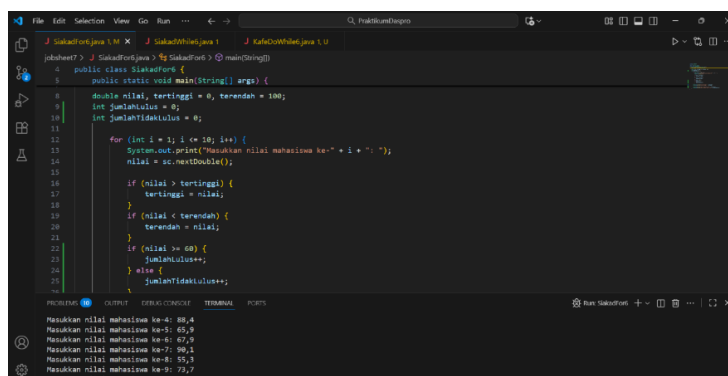
- Baris pertama membandingkan nilai input dengan nilai tertinggi saat ini. Jika lebih besar, maka nilai tersebut menjadi nilai tertinggi baru.
- Baris kedua membandingkan nilai input dengan nilai terendah saat ini. Jika lebih kecil, maka nilai tersebut menjadi nilai terendah baru.

**Alur kerja:**

1. Kondisi pertama mengecek apakah nilai yang diinput lebih besar dari nilai tertinggi yang tersimpan. Jika ya, maka variabel tertinggi diperbarui dengan nilai yang baru diinput.
2. Kondisi kedua mengecek apakah nilai yang diinput lebih kecil dari nilai terendah yang tersimpan. Jika ya, maka variabel terendah diperbarui dengan nilai yang baru diinput.
3. Kedua kondisi ini berjalan secara terpisah (bukan if-else) sehingga kedua pengecekan dilakukan untuk setiap input nilai.

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!

**Jawab:**

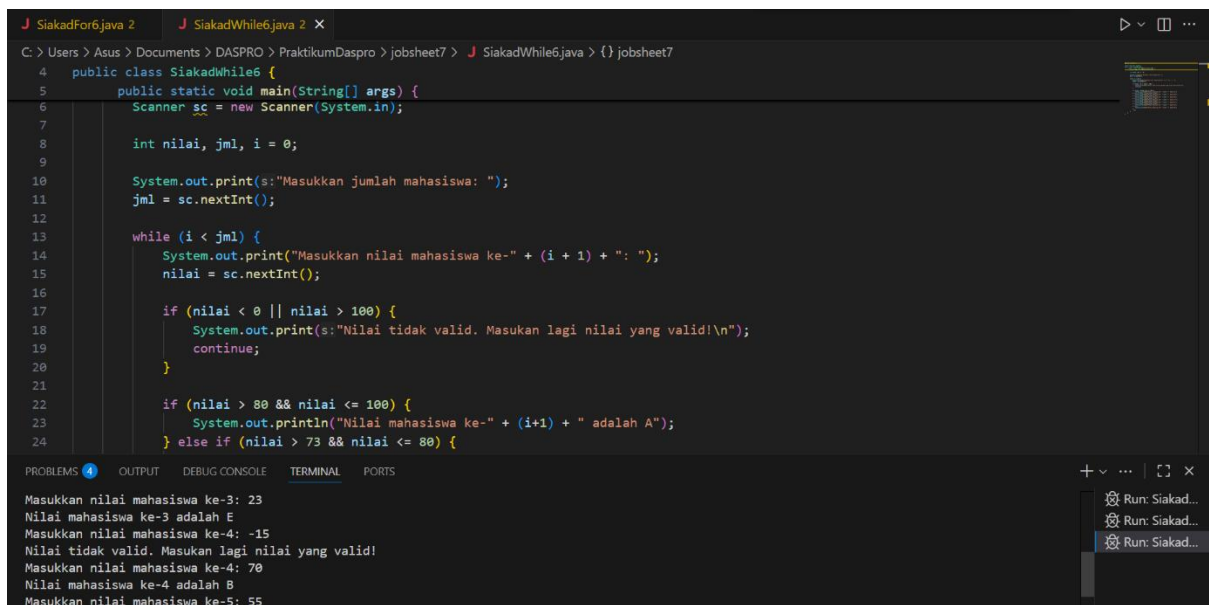


```
1 public class SisaKode {  
2     public static void main(String[] args) {  
3         double nilai, tertinggi = 0, terendah = 100;  
4         int jumlahLulus = 0;  
5         int jumlahTidakLulus = 0;  
6  
7         for (int i = 1; i <= 10; i++) {  
8             System.out.print("Masukkan nilai mahasiswa ke-" + i + " : ");  
9             nilai = sc.nextDouble();  
10  
11             if (nilai > tertinggi) {  
12                 tertinggi = nilai;  
13             }  
14             if (nilai < terendah) {  
15                 terendah = nilai;  
16             }  
17  
18             if (nilai >= 60) {  
19                 jumlahLulus++;  
20             } else {  
21                 jumlahTidakLulus++;  
22             }  
23         }  
24     }  
25 }
```

OUTPUT

```
Masukkan nilai mahasiswa ke-1 : 85.4  
Masukkan nilai mahasiswa ke-2 : 65.9  
Masukkan nilai mahasiswa ke-3 : 67.9  
Masukkan nilai mahasiswa ke-4 : 90.1  
Masukkan nilai mahasiswa ke-5 : 55.3  
Masukkan nilai mahasiswa ke-6 : 71.7  
Masukkan nilai mahasiswa ke-7 : 79.6
```

## Percobaan 2



```
4 public class SiakadWhile6 {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         int nilai, jml, i = 0;
9
10        System.out.print("Masukkan jumlah mahasiswa: ");
11        jml = sc.nextInt();
12
13        while (i < jml) {
14            System.out.print("Masukkan nilai mahasiswa ke-" + (i + 1) + ": ");
15            nilai = sc.nextInt();
16
17            if (nilai < 0 || nilai > 100) {
18                System.out.print("Nilai tidak valid. Masukkan lagi nilai yang valid!\n");
19                continue;
20            }
21
22            if (nilai > 80 && nilai <= 100) {
23                System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah A");
24            } else if (nilai > 73 && nilai <= 80) {
```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

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

Run: Siakad...  
Run: Siakad...  
Run: Siakad...

## 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`

### Jawab:

- a.  $\text{nilai} < 0 \parallel \text{nilai} > 100 \rightarrow$  Sintaks tersebut digunakan untuk mengecek apakah nilai yang dimasukkan tidak valid. Pernyataan akan bernilai TRUE jika nilai kurang dari 0 atau lebih dari 100. Jika TRUE, program akan menganggap input tidak valid dan meminta pengguna memasukkan nilai ulang.
  - b. `continue`  $\rightarrow$  Perintah `continue` digunakan untuk melewati (skip) satu iterasi saat ini dan langsung melanjutkan ke iterasi berikutnya pada perulangan WHILE. Pada program, `continue` dipakai ketika nilai tidak valid sehingga proses kategori nilai tidak dijalankan untuk input tersebut.
2. Mengapa sintaks `i++` dituliskan di akhir perulangan WHILE? Apa yang terjadi jika posisinya dituliskan di awal perulangan WHILE?

### Jawab:

Sintaks `i++` diletakkan di akhir perulangan WHILE karena proses input dan pengecekan nilai harus dilakukan terlebih dahulu sebelum indeks mahasiswa dinaikkan. Dengan

menempatkan `i++` di akhir, setiap mahasiswa diproses secara berurutan mulai dari mahasiswa ke-0 hingga mahasiswa terakhir.

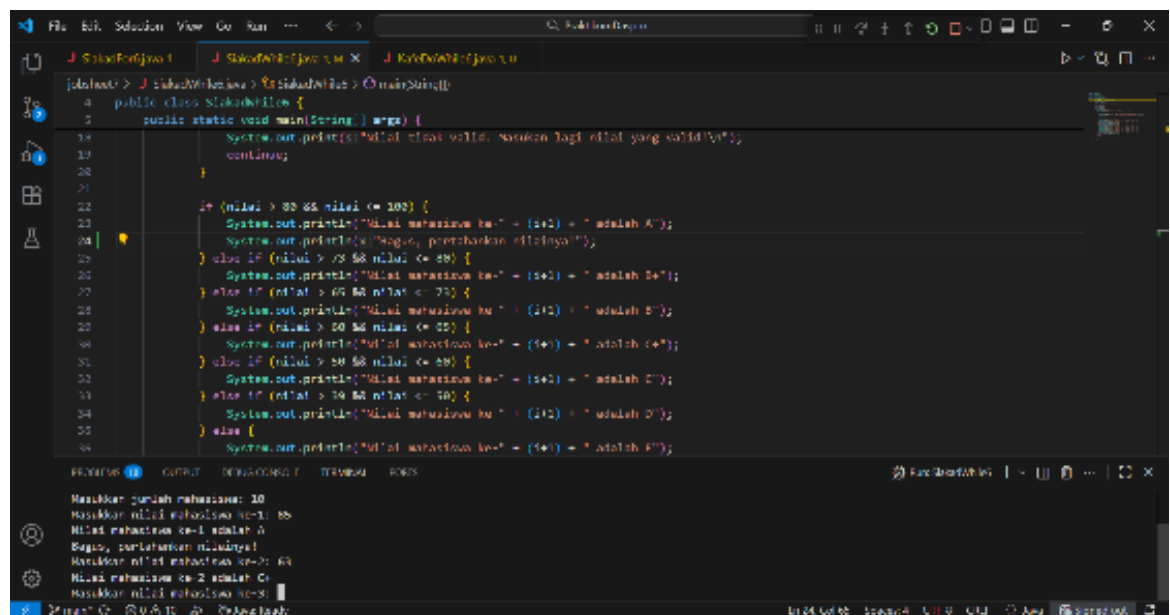
Namun, jika `i++` dituliskan di awal perulangan, maka nilai `i` akan menjadi 1 sebelum input pertama diproses. Hal ini menyebabkan data mahasiswa ke-0 tidak pernah dibaca, dan perulangan langsung memproses mahasiswa mulai dari indeks 1 hingga indeks terakhir. Jumlah perulangan tetap sama, tetapi urutan data bergeser satu posisi, sehingga mahasiswa pertama hilang dan tidak pernah diinput.

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

**Jawab:** Perulangan akan berjalan 19 kali. Variabel `i` dimulai dari 0 dan akan terus berulang selama `i < 19`, yaitu untuk `i = 0, 1, 2, 3`, sampai 18.

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

**Jawab:**

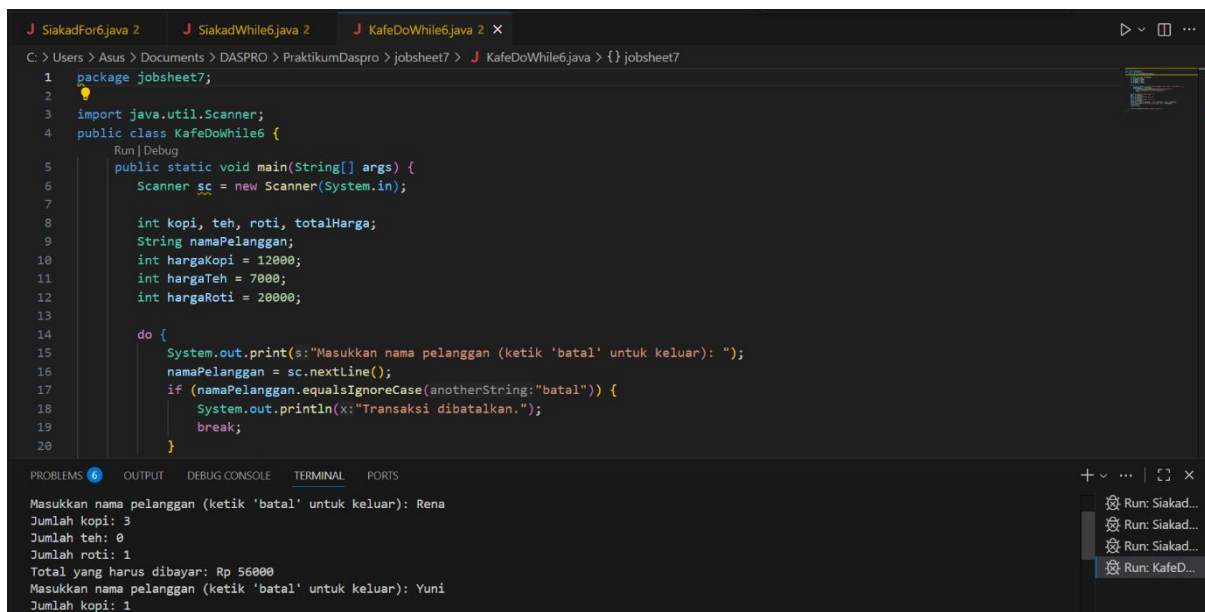


```
File Edit Solution View Go Run ...<> F:\Koding\Java\...<
J: SikadWinJaw... X J: SikadWinJaw... X J: SikadWinJaw... X
package < >
public class SikadWinJaw {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Masukin nama valid, max 100 karakter ya!");
        continue;
    }
    while (true) {
        if (nilai > 85 && nilai <= 100) {
            System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah A");
            System.out.println("Bagus, pertahankan nilainya!");
        } else if (nilai > 75 && nilai <= 85) {
            System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah B+");
        } else if (nilai > 65 && nilai <= 75) {
            System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah B");
        } else if (nilai > 55 && nilai <= 65) {
            System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah C+");
        } else if (nilai > 45 && nilai <= 55) {
            System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah C");
        } else if (nilai > 35 && nilai <= 45) {
            System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah D");
        } else {
            System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah E");
        }
        i++;
    }
}
```

PROGRAM OUTPUT DEBUGGER TEST RUN

Masukkan jumlah mahasiswa: 3  
Masukkan nilai mahasiswa ke-1: 88  
Nilai mahasiswa ke-1 adalah A  
Bagus, pertahankan nilainya!  
Masukkan nilai mahasiswa ke-2: 65  
Nilai mahasiswa ke-2 adalah C  
Masukkan nilai mahasiswa ke-3: 45  
Nilai mahasiswa ke-3 adalah D

## Percobaan 3



```
1 package jobsheet7;
2
3 import java.util.Scanner;
4 public class KafeDoWhile6 {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         int kopi, teh, roti, totalHarga;
9         String namaPelanggan;
10        int hargaKopi = 12000;
11        int hargaTeh = 7000;
12        int hargaRoti = 20000;
13
14        do {
15            System.out.print("Masukkan nama pelanggan (ketik 'batal' untuk keluar): ");
16            namaPelanggan = sc.nextLine();
17            if (namaPelanggan.equalsIgnoreCase("batal")) {
18                System.out.println("Transaksi dibatalkan.");
19                break;
20            }
21        } while (true);
22
23        System.out.println("Masukkan jumlah kopi: ");
24        kopi = sc.nextInt();
25        System.out.println("Masukkan jumlah teh: ");
26        teh = sc.nextInt();
27        System.out.println("Masukkan jumlah roti: ");
28        roti = sc.nextInt();
29
30        totalHarga = (kopi * hargaKopi) + (teh * hargaTeh) + (roti * hargaRoti);
31        System.out.println("Total yang harus dibayar: Rp " + totalHarga);
32
33        System.out.print("Masukkan nama pelanggan (ketik 'batal' untuk keluar): ");
34        namaPelanggan = sc.nextLine();
35        if (namaPelanggan.equalsIgnoreCase("batal")) {
36            System.out.println("Transaksi dibatalkan.");
37            break;
38        }
39    }
40 }
```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Masukkan nama pelanggan (ketik 'batal' untuk keluar): Rena  
Jumlah kopi: 3  
Jumlah teh: 0  
Jumlah roti: 1  
Total yang harus dibayar: Rp 56000  
Masukkan nama pelanggan (ketik 'batal' untuk keluar): Yuni  
Jumlah kopi: 1

## Pertanyaan

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

**Jawab:** Perulangan DO-WHILE selalu dijalankan minimal satu kali, karena proses dijalankan terlebih dahulu sebelum kondisi dicek. Meskipun input pertama adalah “batal”, perulangan tetap dieksekusi 1 kali.

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

**Jawab:** Kondisi berhenti pada perulangan DO-WHILE terjadi ketika pengguna memasukkan kata “batal”. Input tersebut menjadi tanda bahwa transaksi dihentikan sehingga program keluar dari perulangan.

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

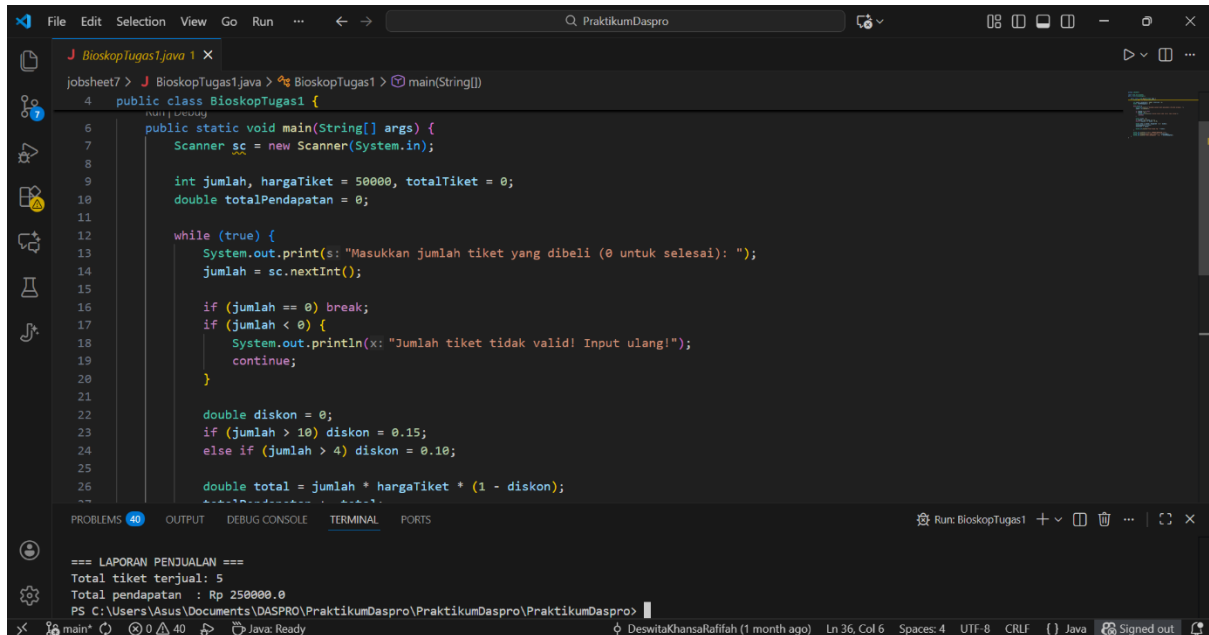
**Jawab:** Penggunaan nilai true pada kondisi DO-WHILE membuat perulangan berjalan tanpa batas karena kondisi tersebut selalu dianggap benar. Dengan demikian, program akan terus meminta input dari pengguna secara berulang tanpa henti. Perulangan hanya akan berhenti jika pengguna memberikan perintah tertentu yang memicu penghentian, seperti memasukkan kata “batal”.

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

**Jawab:** Perulangan DO-WHILE pada program ini tidak bergantung pada counter seperti i. Loop terus berjalan karena kondisi while(true) menciptakan loop tak terbatas. Perulangan hanya berhenti jika pengguna memasukkan nilai sentinel (“batal”), yang memicu perintah break. Oleh karena itu, meskipun tidak ada inisialisasi variabel penghitung atau update di dalam DO-WHILE, loop tetap aktif selama belum masuk kondisi berhenti.

# TUGAS

## Tugas 1



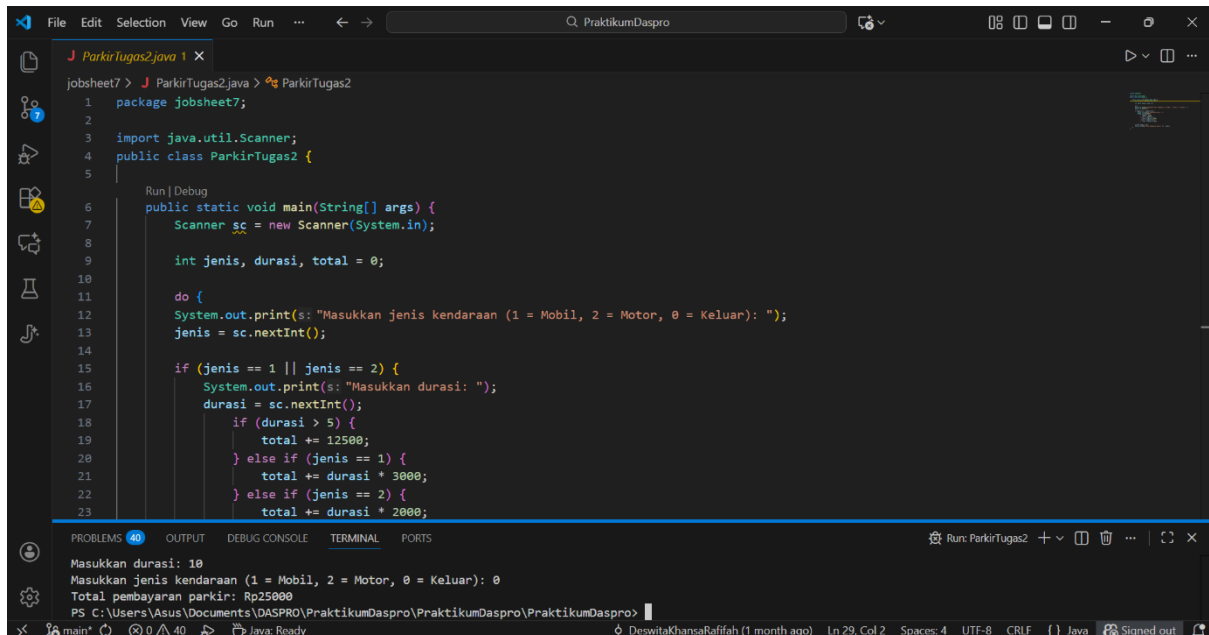
The screenshot shows an IDE window titled "PraktikumDaspro" with a file named "BioskopTugas1.java". The code defines a class "BioskopTugas1" with a "main" method. The program prompts the user to enter the number of tickets, validates the input, calculates a discount based on the quantity, and then calculates the total revenue. The terminal output shows the program running successfully with the input 5 tickets, resulting in a total revenue of Rp 250000.0.

```
public class BioskopTugas1 {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        int jumlah, hargaTiket = 50000, totalTiket = 0;  
        double totalPendapatan = 0;  
  
        while (true) {  
            System.out.print(s: "Masukkan jumlah tiket yang dibeli (0 untuk selesai): ");  
            jumlah = sc.nextInt();  
  
            if (jumlah == 0) break;  
            if (jumlah < 0) {  
                System.out.println(x: "Jumlah tiket tidak valid! Input ulang!");  
                continue;  
            }  
  
            double diskon = 0;  
            if (jumlah > 10) diskon = 0.15;  
            else if (jumlah > 4) diskon = 0.10;  
  
            double total = jumlah * hargaTiket * (1 - diskon);  
            totalPendapatan += total;  
            totalTiket += jumlah;  
        }  
  
        System.out.println("==== LAPORAN PENJUALAN ====");  
        System.out.println("Total tiket terjual: " + totalTiket);  
        System.out.println("Total pendapatan : Rp " + totalPendapatan);  
    }  
}
```

Terminal Output:

```
==== LAPORAN PENJUALAN ===  
Total tiket terjual: 5  
Total pendapatan : Rp 250000.0  
PS C:\Users\Asus\Documents\Daspro\PraktikumDaspro>
```

## Tugas 2



The screenshot shows an IDE window titled "PraktikumDaspro" with a file named "ParkirTugas2.java". The code defines a class "ParkirTugas2" with a "main" method. The program prompts the user to enter the type of vehicle (1 for Mobil, 2 for Motor, 0 for Keluar) and the duration of parking. It then calculates the total parking fee based on the vehicle type and duration. The terminal output shows the program running successfully with the input 1 for Mobil and 10 for duration, resulting in a total parking fee of Rp 25000.

```
package jobsheet7;  
  
import java.util.Scanner;  
public class ParkirTugas2 {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        int jenis, durasi, total = 0;  
  
        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: ");  
                durasi = sc.nextInt();  
  
                if (durasi > 5) {  
                    total += 12500;  
                } else if (jenis == 1) {  
                    total += durasi * 3000;  
                } else if (jenis == 2) {  
                    total += durasi * 2000;  
                }  
            }  
        } while (jenis != 0);  
  
        System.out.println("Total pembayaran parkir: Rp " + total);  
    }  
}
```

Terminal Output:

```
Masukkan durasi: 10  
Masukkan jenis kendaraan (1 = Mobil, 2 = Motor, 0 = Keluar): 0  
Total pembayaran parkir: Rp 25000  
PS C:\Users\Asus\Documents\Daspro\PraktikumDaspro>
```