

NAMA : SUTOKO

KELAS : SIB-1B

---

## JOB SHEET 7 PERULANGAN 1

### A. PERCOBAAN 1

```
1  import java.util.Scanner;
2  /**
3   * SiakadFor23
4   */
5  public class SiakadFor23 {
6
7      public static void main(String[] args) {
8          Scanner sc = new Scanner(System.in);
9
10         double nilai, tertinggi = 0, terendah = 100;
11
12         for (int i = 1; i <= 10; i++) {
13             System.out.println("Masukkan nilai mahasiswa ke-" + i + ":");
14             nilai = sc.nextDouble();
15             if (nilai > tertinggi) {
16                 tertinggi = nilai;
17             }
18             if (nilai < terendah) {
19                 terendah = nilai;
20             }
21         }
22         System.out.println("Nilai tertinggi: " + tertinggi);
23         System.out.println("Nilai terendah: " + terendah);
24         sc.close();
25     }
26 }
```

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

JAWABAN:

Dalam perulangan **for** pada program, ada tiga komponen utama, yaitu **inisialisasi**, **kondisi**, dan **inkrementasi**.

- **Inisialisasi:** `int i = 1;`  
Ini adalah bagian di mana variabel *i* diinisialisasi dengan nilai 1. Variabel ini digunakan untuk menghitung loop, dimulai dari 1.
- **Kondisi:** `i <= 10;`  
Loop akan berjalan selama kondisi ini terpenuhi, yaitu selama nilai *i* kurang dari atau sama dengan 10. Setelah *i* melebihi 10, loop akan berhenti.
- **Inkrementasi:** `i++`  
Pada setiap iterasi, nilai *i* akan bertambah 1, yang berarti loop akan berjalan 10 kali, dari 1 hingga 10.

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

JAWABAN:

- Alasan inisialisasi tertinggi = 0:  
Nilai tertinggi diinisialisasi dengan 0 karena kita menganggap bahwa nilai yang akan dimasukkan minimal 0 atau lebih. Dengan inisialisasi ini, nilai tertinggi yang pertama kali dimasukkan akan menggantikan 0 jika lebih besar.
- Alasan inisialisasi terendah = 100:  
Nilai terendah diinisialisasi dengan 100 karena diasumsikan nilai maksimal yang mungkin adalah 100. Jika nilai yang dimasukkan lebih kecil dari 100, variabel terendah akan segera diperbarui dengan nilai tersebut.

Apa yang terjadi jika tertinggi diinisialisasi 100 dan terendah diinisialisasi 0?

- Jika tertinggi diinisialisasi dengan 100, maka semua nilai input yang lebih kecil dari 100 tidak akan pernah mengganti nilai tertinggi. Akibatnya, nilai tertinggi yang terdeteksi selalu 100, kecuali jika ada nilai input yang lebih besar.
- Jika terendah diinisialisasi dengan 0, maka tidak akan ada nilai yang lebih rendah dari 0, sehingga nilai terendah akan tetap 0, bahkan jika ada nilai input yang lebih tinggi. Ini akan menghasilkan hasil yang salah.

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

JAWABAN:

Potongan kode membandingkan setiap nilai input dengan variabel tertinggi dan terendah. Nilai tertinggi dan terendah diperbarui jika input lebih besar atau lebih kecil dari variabel 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!

JAWABAN:

```
1 import java.util.Scanner;
2 /**
3  * SiakadFor23
4  */
5 public class SiakadFor23 {
6
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9
10        double nilai, tertinggi = 0, terendah = 100;
11        int jmlhLulus = 0, jmlhTidakLulus = 0, batasKelulusan = 60;
12
13        for (int i = 1; i <= 10; i++) {
14            System.out.println("Masukkan nilai mahasiswa ke-" + i + ":");
15            nilai = sc.nextDouble();
16
17            if (nilai > tertinggi) {
18                tertinggi = nilai;
19            }
20            if (nilai < terendah) {
21                terendah = nilai;
22            }
23            if (nilai >= batasKelulusan) {
24                jmlhLulus++;
25            } else {
26                jmlhTidakLulus++;
27            }
28        }
29        System.out.println("Nilai tertinggi: " + tertinggi);
30        System.out.println("Nilai terendah: " + terendah);
31
32        System.out.println("Jumlah mahasiswa yang lulus: " + jmlhLulus);
33        System.out.println("Jumlah mahasiswa yang tidak lulus: " + jmlhTidakLulus);
34
35        sc.close();
36    }
37 }
```

5. Commit dan push kode program ke Github  
<https://github.com/sutoko395/daspro-jobsheet7.git>

## B. PERCOBAAN 2

```
1  import java.util.Scanner;
2  /**
3   * SiakadWhile23
4   */
5  public class SiakadWhile23 {
6
7      public static void main(String[] args) {
8          Scanner sc = new Scanner(System.in);
9          int nilai, jml, i;
10         i = 0;
11
12         System.out.println("Masukkan jumlah mahasiswa: ");
13         jml = sc.nextInt();
14
15         while (i < jml) {
16             System.out.println("Masukkan nilai mahasiswa ke-" + (i+1) + ":");
17             nilai = sc.nextInt();
18             if (nilai < 0 || nilai > 100) {
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 <= 73) {
25                 System.out.println("Nilai mahasiswa ke-" + (i+1) + "adalah B+");
26             } else if (nilai > 65 && nilai <= 73) {
27                 System.out.println("Nilai mahasiswa ke-" + (i+1) + "adalah B");
28             } else if (nilai > 60 && nilai <= 65) {
29                 System.out.println("Nilai mahasiswa ke-" + (i+1) + "adalah C+");
30             } else if (nilai > 50 && nilai <= 60) {
31                 System.out.println("Nilai mahasiswa ke-" + (i+1) + "adalah C");
32             } else if (nilai > 39 && nilai <= 50) {
33                 System.out.println("Nilai mahasiswa ke-" + (i+1) + "adalah D");
34             } else
35                 System.out.println("Nilai mahasiswa ke-" + (i+1) + "adalah E");
36             i++;
37
38             sc.close();
39         }
40     }
41 }
42
43
44
```

1. Pada potongan kode berikut, tentukan maksud dan kegunaan dari sintaks berikut: a. nilai < 0 || nilai > 100 b. continue

JAWABAN:

- a. nilai < 0 || nilai > 100:

Memeriksa apakah nilai yang dimasukkan tidak valid (di luar rentang 0 hingga 100). Jika kondisi ini true, maka perintah selanjutnya tidak akan dijalankan, dan pengguna diminta memasukkan nilai ulang.

b. continue:

Berfungsi untuk langsung melanjutkan ke iterasi berikutnya dalam perulangan, melewati kode di bawahnya. Dalam konteks ini, digunakan jika nilai yang dimasukkan tidak valid.

2. Mengapa sintaks i++ dituliskan di akhir perulangan WHILE? Apa yang terjadi jika posisinya dituliskan di awal perulangan WHILE?

JAWABAN:

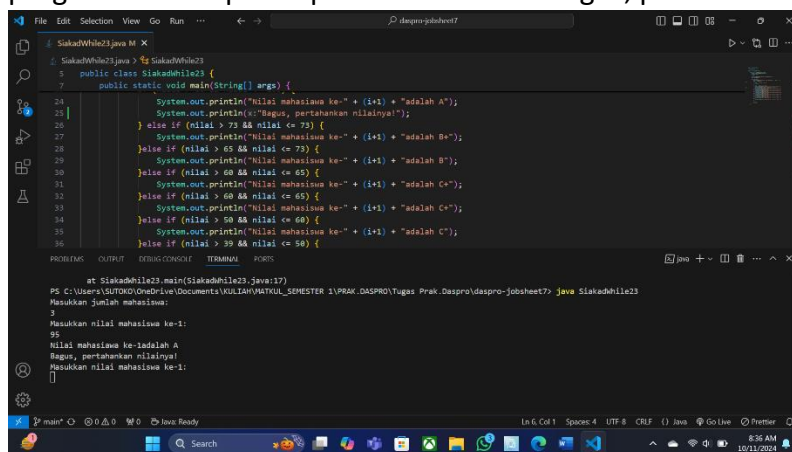
- Mengapa di akhir: Diletakkan di akhir agar variabel i hanya bertambah ketika input nilai valid. Jika i++ ditulis di awal, bahkan ketika nilai tidak valid (di luar 0-100), perulangan akan tetap menghitungnya, yang mengakibatkan jumlah iterasi tidak sesuai dengan jumlah mahasiswa.
- Jika ditulis di awal: Perulangan akan mengabaikan validasi input dan tetap menambah nilai i, yang menyebabkan mahasiswa dengan nilai tidak valid tetap dihitung, sehingga perulangan selesai lebih cepat dari yang seharusnya.

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

JAWABAN:

Jika jumlah mahasiswa yang dimasukkan adalah 19, maka perulangan **WHILE** akan berjalan **19 kali**, selama tidak ada input yang tidak valid. Setiap iterasi akan menerima input untuk satu mahasiswa.

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



```
1 public class SiakadHile23 {
2     public static void main(String[] args) {
3         // ...
4         while (jumlah < 19) {
5             // ...
6             System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah A");
7             System.out.println("Bagus, pertahankan nilainya!");
8             // ...
9             if (nilai > 75 && nilai <= 79) {
10                 System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah B+");
11             }
12             // ...
13             if (nilai > 65 && nilai <= 69) {
14                 System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah B");
15             }
16             // ...
17             if (nilai > 55 && nilai <= 59) {
18                 System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah C+");
19             }
20             // ...
21             if (nilai > 45 && nilai <= 49) {
22                 System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah C");
23             }
24             // ...
25             if (nilai > 35 && nilai <= 39) {
26                 System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah D");
27             }
28             // ...
29             if (nilai > 25 && nilai <= 29) {
30                 System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah E");
31             }
32             // ...
33             if (nilai > 15 && nilai <= 19) {
34                 System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah F");
35             }
36             // ...
37             if (nilai > 5 && nilai <= 9) {
38                 System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah G");
39             }
40             // ...
41             if (nilai > 0 && nilai <= 4) {
42                 System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah H");
43             }
44             // ...
45             if (nilai < 0 || nilai > 100) {
46                 System.out.println("Nilai tidak valid, masukkan nilai yang benar!");
47                 continue;
48             }
49             // ...
50             jumlah++;
51             i++;
52         }
53     }
54 }
```

5. Commit dan push kode program ke Github

<https://github.com/sutoko395/daspro-jobsheet7.git>

### C. PERCOBAAN 3

```
1  import java.util.Scanner;
2  /**
3   * KafeDowWhile23
4   */
5  public class KafeDowWhile23 {
6
7      public static void main(String[] args) {
8          Scanner sc = new Scanner(System.in);
9
10         int kopi, teh, roti;
11         String namaPelanggan;
12         double totalHarga, hargaKopi = 12000, hargaTeh = 7000, hargaRoti = 20000;
13
14         do {
15             System.out.println("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             System.out.println("Jumlah kopi: ");
22             kopi = sc.nextInt();
23             System.out.println("Jumlah Teh");
24             teh = sc.nextInt();
25             System.out.println("Jumlah roti");
26             roti = sc.nextInt();
27
28             totalHarga = (kopi * hargaKopi) + (teh * hargaTeh) + (roti * hargaRoti);
29             System.out.println("Total yang harus dibayar : Rp" + totalHarga);
30             sc.nextLine();
31         }while (true);
32
33         System.out.println("Semua transaksi selesai.");
34
35         sc.close();
36     }
37 }
```

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

JAWABAN:

Jika nama pelanggan yang dimasukkan pertama kali adalah "batal", maka perulangan dilakukan 1 kali. Loop do-while akan menjalankan blok kode setidaknya sekali sebelum memeriksa kondisi berhenti.

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

JAWABAN:

Kondisi berhenti yang digunakan dalam perulangan do-while adalah ketika pengguna memasukkan nama pelanggan "batal". Jika kondisi ini terpenuhi, program akan menampilkan pesan "Transaksi dibatalkan" dan keluar dari loop menggunakan perintah break.

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

JAWABAN:

Penggunaan nilai true pada kondisi do-while membuat loop menjadi infinite loop (loop tanpa henti) yang hanya dapat dihentikan dengan perintah break di dalam blok. Hal ini memungkinkan kode dalam loop untuk dieksekusi terus-menerus hingga kondisi tertentu (dalam hal ini, input nama pelanggan) terpenuhi.

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

JAWABAN:

Perulangan do-while tetap berjalan meskipun tidak ada komponen inisialisasi dan update karena do-while menjalankan blok kode minimal sekali sebelum memeriksa kondisi. Dalam contoh ini, perulangan terus berjalan karena tidak ada batasan yang mengatur pengulangan berdasarkan nilai i atau kondisi lainnya. Perulangan dihentikan hanya ketika ada input yang sesuai (yaitu "batal").

## D. 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.

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

```
1  import java.util.Scanner;
2  /**
3   * PenjualanTiketBioskop23
4   */
5  public class PenjualanTiketBioskop23 {
6
7      public static void main(String[] args) {
8          Scanner sc = new Scanner(System.in);
9
10         int hargaTiket = 50000, totalTiket = 0;
11         double totalPenjualan = 0;
12         boolean selesai = false;
13
14         while (!selesai) {
15             System.out.print("Masukkan jumlah tiket yang terjual (atau -1 untuk selesai): ");
16             int jumlahTiket = sc.nextInt();
17
18             if (jumlahTiket == -1) {
19                 selesai = true;
20             } else if (jumlahTiket < 0) {
21                 System.out.println("Input tidak valid. Silakan masukkan jumlah tiket yang benar.");
22                 continue;
23             } else {
24                 totalTiket += jumlahTiket;
25
26                 double hargaTotal = jumlahTiket * hargaTiket;
27                 if (jumlahTiket > 10) {
28                     hargaTotal *= 0.85;
29                 } else if (jumlahTiket > 4) {
30                     hargaTotal *= 0.90;
31                 }
32
33                 totalPenjualan += hargaTotal;
34                 System.out.println("Penjualan untuk " + jumlahTiket + " tiket adalah: Rp " + hargaTotal);
35             }
36         }
37
38         System.out.println("\nTotal tiket yang terjual hari ini: " + totalTiket);
39         System.out.println("Total penjualan tiket hari ini: Rp " + totalPenjualan);
40
41         sc.close();
42     }
43 }
```



2. 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!

```
1  import java.util.Scanner;
2  /**
3   * PembayaranParkir23
4   */
5  public class PembayaranParkir23 {
6
7      public static void main(String[] args) {
8          Scanner sc = new Scanner(System.in);
9
10         int jenis, durasi;
11         double total = 0;
12
13         do {
14             System.out.println("Masukkan jenis kendaraan (1: Mobil, 2: Motor, 0: Keluar): ");
15             jenis = sc.nextInt();
16
17             if (jenis == 0) {
18                 break;
19             }
20
21             if (jenis == 1 || jenis == 2) {
22                 System.out.println("Masukkan durasi parkir (dalam jam): ");
23                 durasi = sc.nextInt();
24
25                 if (durasi < 0) {
26                     System.out.println("Durasi parkir tidak boleh negatif.");
27                     continue;
28                 }
29                 if (durasi > 5) {
30                     total += 12500;
31                 } else {
32                     if (jenis == 1) {
33                         total += durasi * 3000;
34                     } else if (jenis == 2) {
35                         total += durasi * 2000;
36                     }
37                 }
38             } else if (jenis != 0) {
39                 System.out.println("Input tidak valid. Silakan masukkan 1 untuk Mobil, 2 untuk Motor, atau 0 untuk keluar.");
40             }
41         } while (true);
42
43         System.out.println("Total pembayaran parkir: Rp " + total);
44         System.out.println("Transaksi selesai.");
45         sc.close();
46     }
47 }
48
```

LINK REPOSITORY SUTOKO(23)

<https://github.com/sutoko395/daspro-jobsheet7.git>