

Perulangan 1

Jobsheet 7

Dosen Pengampu:

Vivi Nur Wijayaningrum, S.Kom, M.Kom



Disusun oleh:

Ahmad Rafid Riqkullah

T1 – 1G

254107020078

POLITEKNIK NEGERI MALANG

Jl. Soekarno Hatta No.9, Jatimulyo, Kec. Lowokwaru, Kota Malang, Jawa Timur

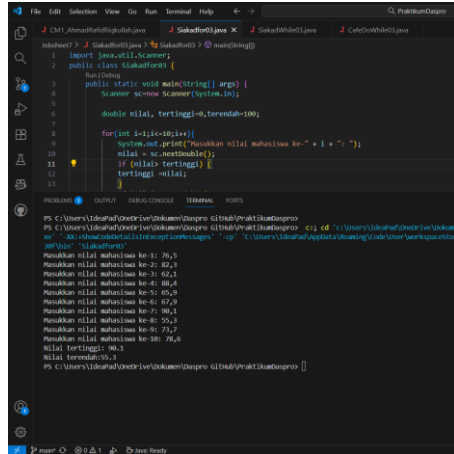
65141

TAHUN 2025-2026

2. Praktikum

2.1 Percobaan 1: Studi Kasus Nilai Mahasiswa di SIAKAD – Perulangan FOR Waktu Percobaan: 90 menit

2.1.2 Verifikasi Hasil Percobaan



```
1 import java.util.Scanner;
2 public class SIAKADFOR01 {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         double nilai, tertinggi=0, terendah=100;
6         for(int i=1; i<=10; i++){
7             System.out.print("Masukkan nilai mahasiswa ke-" + i + ": ");
8             nilai = sc.nextDouble();
9             if (nilai > tertinggi) {
10                 tertinggi = nilai;
11             }
12         }
13     }
14 }
```

OUTPUT

```
PS C:\Users\lida\OneDrive\Documents> cd C:\Users\lida\OneDrive\Documents
PS C:\Users\lida\OneDrive\Documents> javac SIAKADFOR01.java
PS C:\Users\lida\OneDrive\Documents> java SIAKADFOR01
Masukkan nilai mahasiswa ke-1: 76,5
Masukkan nilai mahasiswa ke-2: 82,5
Masukkan nilai mahasiswa ke-3: 82,1
Masukkan nilai mahasiswa ke-4: 89,4
Masukkan nilai mahasiswa ke-5: 85,9
Masukkan nilai mahasiswa ke-6: 87,9
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
PS C:\Users\lida\OneDrive\Documents>
```

```
1 import java.util.Scanner;
2 public class SIAKADFOR01 {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5
6         double nilai, tertinggi=0, terendah=100;
7
8         for(int i=1; i<=10; i++){
9             System.out.print("Masukkan nilai mahasiswa ke-" + i + ": ");
10            nilai = sc.nextDouble();
11            if (nilai > tertinggi) {
12                tertinggi = nilai;
13            }
14            if (nilai < terendah) {
15                terendah = nilai;
16            }
17        }
18        System.out.println("Nilai tertinggi: " + tertinggi);
19        System.out.println("Nilai terendah: " + terendah);
20
21        sc.close();
22    }
23 }
24 }
25 }
```

2.1.3 Pertanyaan

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

Jawab:

- **Inisialisasi: $\text{int } i = 1$:** variabel penghitung dimulai dari 1
- **Kondisi: $i \leq 10$:** perulangan akan terus jalan selama nilai i masih kurang dari atau sama dengan 10
- **Update: $i++$:** setiap perulangan, nilai i bertambah 1

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

Jawab:

- Tertinggi = 0 karena di awal kita anggap belum ada nilai, jadi kalau ada nilai lebih besar dari 0, otomatis langsung jadi nilai tertinggi.
- Terendah = 100 karena kita anggap batas maksimal nilai adalah 100, jadi kalau ada nilai lebih kecil dari 100, langsung dianggap terendah.

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

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

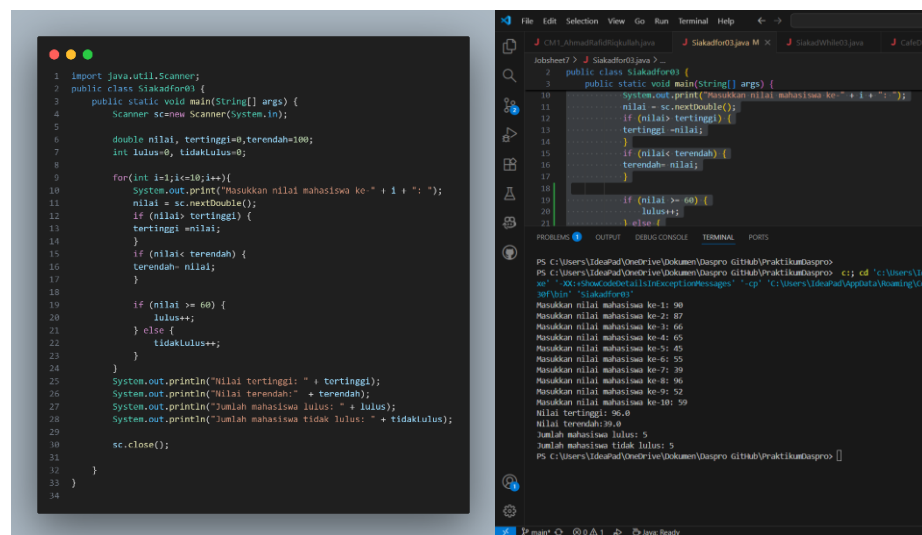
Jawab:

Untuk mencari nilai tertinggi dan terendah dari semua nilai yang dimasukkan.

- Kalau nilai yang baru lebih besar dari nilai tertinggi sebelumnya → ganti nilai tertinggi.
- Kalau nilai yang baru lebih kecil dari nilai terendah sebelumnya → ganti nilai terendah.

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:



The screenshot shows an IDE with two panels. The left panel displays the Java code for a program that calculates the highest and lowest scores of students and counts the number of students who passed (score ≥ 60) and failed (score < 60). The right panel shows the output of the program, which includes the highest and lowest scores and the counts of students who passed and failed.

```
1 import java.util.Scanner;  
2 public class Sikadfor03 {  
3     public static void main(String[] args) {  
4         Scanner sc = new Scanner(System.in);  
5  
6         double nilai, tertinggi=0, terendah=100;  
7         int lulus=0, tidaklulus=0;  
8  
9         for(int i=1; i<=10; i++){  
10            System.out.print("Masukkan nilai mahasiswa ke-" + i + ": ");  
11            nilai = sc.nextDouble();  
12            if (nilai > tertinggi) {  
13                tertinggi = nilai;  
14            }  
15            if (nilai < terendah) {  
16                terendah = nilai;  
17            }  
18  
19            if (nilai >= 60) {  
20                lulus++;  
21            } else {  
22                tidaklulus++;  
23            }  
24        }  
25        System.out.println("Nilai tertinggi: " + tertinggi);  
26        System.out.println("Nilai terendah: " + terendah);  
27        System.out.println("Jumlah mahasiswa lulus: " + lulus);  
28        System.out.println("Jumlah mahasiswa tidak lulus: " + tidaklulus);  
29  
30        sc.close();  
31    }  
32 }  
33 }  
34 }
```

Output:

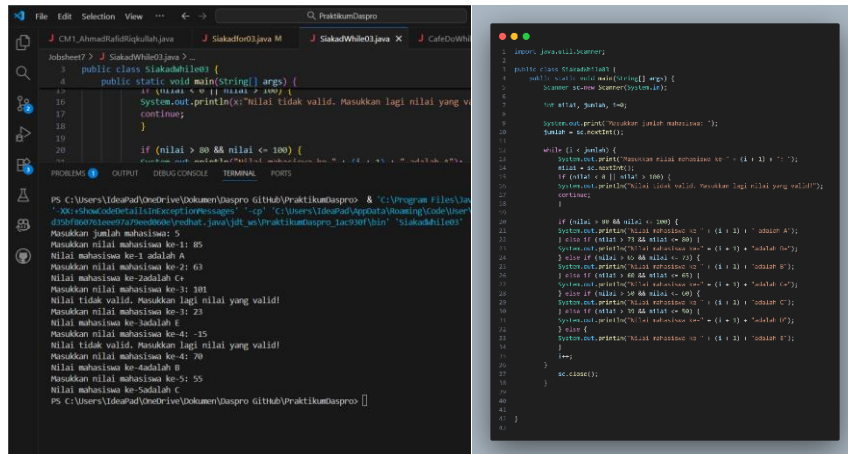
```
PS C:\Users\idead\OneDrive\Documents> java -cp . Sikadfor03  
Masukkan nilai mahasiswa ke-1: 90  
Masukkan nilai mahasiswa ke-2: 87  
Masukkan nilai mahasiswa ke-3: 66  
Masukkan nilai mahasiswa ke-4: 65  
Masukkan nilai mahasiswa ke-5: 45  
Masukkan nilai mahasiswa ke-6: 35  
Masukkan nilai mahasiswa ke-7: 39  
Masukkan nilai mahasiswa ke-8: 96  
Masukkan nilai mahasiswa ke-9: 52  
Masukkan nilai mahasiswa ke-10: 59  
Nilai tertinggi: 96.0  
Nilai terendah: 35.0  
Jumlah mahasiswa lulus: 5  
Jumlah mahasiswa tidak lulus: 5  
PS C:\Users\idead\OneDrive\Documents>
```

5. Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan 1”

CafeDoWhile03.java	Percobaan 3	3 days ago
PenjualanTiketBuskop.java	Tugas 1	34 minutes ago
SikadWhile03.java	Modifikasi Percobaan 2	1 hour ago
Sikadfor03.java	Modifikasi Percobaan 1	1 hour ago

2.2 Percobaan 2: Studi Kasus Nilai Mahasiswa di SIAKAD – Perulangan WHILE Waktu Percobaan: 90 menit

2.2.2 Langkah-langkah Percobaan



2.2.3 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. `nilai < 0 || nilai > 100`

b. `continue`

Jawab:

- a. Kondisi ini memeriksa apakah nilai yang dimasukkan tidak valid, yaitu:

- kurang dari 0, atau
- lebih dari 100

Kalau salah satu kondisi benar, maka dianggap nilai tidak valid.

- b. Perintah `continue` digunakan untuk melewati sisa kode dalam perulangan dan langsung lanjut ke perulangan berikutnya.

Jadi kalau nilai yang dimasukkan tidak valid, program tidak akan lanjut memproses nilai itu, tapi langsung minta input nilai lagi.

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

Jawab:

- `i++` di akhir perulangan berarti penghitung (*i*) baru akan bertambah setelah semua proses di dalam perulangan selesai.
- Kalau `i++` ditulis di awal, maka nilai *i* akan langsung bertambah sebelum data diproses

- Perulangannya akan berjalan **sebanyak 19 kali**, karena kondisi perulangan adalah `while (i < jumlah)` dan `jumlah = 19`

- [illegible]

- | | | |
|------------------------------|------------------------|----------------|
| ☐ CafeDoWhile03.java | Percobaan 3 | 3 days ago |
| ☐ PenjualanTiketBioskop.java | Tugas 1 | 34 minutes ago |
| ☐ StakadWhile03.java | Modifikasi Percobaan 2 | 1 hour ago |
| ☐ Stakadfor03.java | Modifikasi Percobaan 1 | 1 hour ago |

Karena pada perulangan DO-WHILE, program akan menjalankan isi perulangan terlebih dahulu minimal sekali, baru setelah itu mengecek kondisi. Jadi walaupun langsung mengetik “batal”, perulangan tetap dijalankan satu kali untuk menampilkan pesan “Transaksi dibatalkan.”

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

Jawab:

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

Artinya, kalau nama pelanggan yang dimasukkan adalah “batal” (huruf besar atau kecil sama saja), maka program akan menghentikan perulangan dengan perintah break.

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

Jawab:

Nilai true membuat perulangan berjalan terus-menerus (infinite loop) sampai ada perintah khusus seperti break yang menghentikannya.

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

Jawab:

Karena perulangan DO-WHILE tidak bergantung pada variabel penghitung seperti FOR atau WHILE biasa. Selama kondisi di bagian while(true) masih bernilai benar, maka perulangan tetap dijalankan. Proses pengulangan dikontrol secara manual dengan perintah break, bukan dengan variabel penghitung (i++, dsb).

3. Tugas Waktu Percobaan : 120 Menit

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”

Catatan: Perulangan dapat menggunakan for, while, atau do-while.

Penambahan break atau continue jika diperlukan

```

1  import java.util.Scanner;
2
3  public class PenjualanTiketBioskop {
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6
7          int hargaTiket = 50000, totalTiket = 0;
8          double totalPenjualan = 0;
9          double diskon = 0;
10
11         System.out.print("Masukkan jumlah pelanggan hari ini: ");
12         int jumlahPelanggan = sc.nextInt();
13
14         for (int i = 1; i <= jumlahPelanggan; i++) {
15             System.out.print("Masukkan jumlah tiket yang dibeli pelanggan ke-" + i + ": ");
16             int jumlahTiket = sc.nextInt();
17
18             // untuk ngecek
19             if (jumlahTiket < 0) {
20                 System.out.println("Jumlah tiket tidak valid! Silakan masukkan ulang.");
21                 i--;
22                 continue;
23             }
24
25             // pilih Diskon
26             if (jumlahTiket > 10) {
27                 diskon = 0.15; // 15%
28             } else if (jumlahTiket > 4) {
29                 diskon = 0.10; // 10%
30             }
31
32             // Hitung
33             double totalHarga = jumlahTiket * hargaTiket * (1 - diskon);
34             totalTiket += jumlahTiket;
35             totalPenjualan += totalHarga;
36
37             System.out.println("Total harga untuk pelanggan ke-" + i + ": Rp " + totalHarga);
38         }
39
40         // Hasil akhir
41         System.out.println("\n=== LAPORAN PENJUALAN HARIAN ===");
42         System.out.println("Total tiket terjual: " + totalTiket);
43         System.out.println("Total penjualan tiket: Rp " + totalPenjualan);
44
45         sc.close();
46     }
47 }
48
49

```

```

Masukkan jumlah pelanggan hari ini: 4
Masukkan jumlah tiket yang dibeli pelanggan ke-1: 4
Total harga untuk pelanggan ke-1: Rp 200000.0
Masukkan jumlah tiket yang dibeli pelanggan ke-2: 5
Total harga untuk pelanggan ke-2: Rp 225000.0
Masukkan jumlah tiket yang dibeli pelanggan ke-3: 10
Total harga untuk pelanggan ke-3: Rp 450000.0
Masukkan jumlah tiket yang dibeli pelanggan ke-4: 11
Total harga untuk pelanggan ke-4: Rp 467500.0

```

```

=== LAPORAN PENJUALAN HARIAN ===

```

```

Total tiket terjual: 30

```

```

Total penjualan tiket: Rp 1342500.0

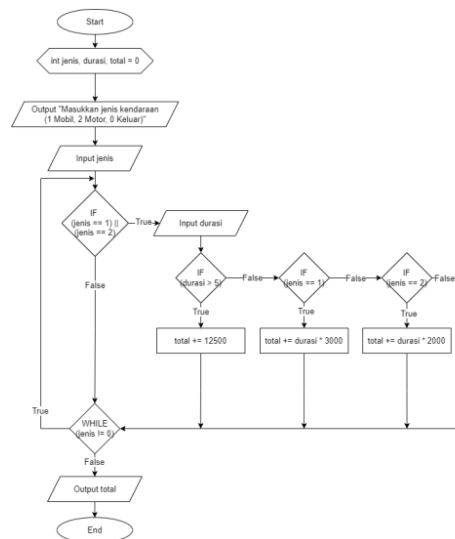
```

```

PS C:\Users\IdeaPad\OneDrive\Dokumen\Daspro GitHub\Praktikum>

```

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”

```

1 import java.util.Scanner;
2
3 public class ParkirWhile_Bonar {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         int jenis = -1, durasi, total = 0;
8
9         System.out.println("=== Program Penghitung Pembayaran Parkir ===");
10        System.out.println("Tarif: Mobil = Rp 3.000/jam | Motor = Rp 2.000/jam");
11        System.out.println("Durasi > 5 jam dikenai tarif tetap Rp 12.500");
12        System.out.println("-----");
13
14        while (true) {
15            System.out.print("\nMasukkan jenis kendaraan (1-Mobil, 2-Motor, 0-Keluar): ");
16            jenis = sc.nextInt();
17
18            if (jenis == 0) {
19                // selesai, keluar dari loop
20                break;
21            }
22
23            // cek jenis valid: harus 1 atau 2
24            if (jenis == 1 || jenis == 2) {
25                // input durasi
26                System.out.print("Masukkan durasi parkir (jam): ");
27                durasi = sc.nextInt();
28
29                if (durasi > 5) {
30                    total += 12500;
31                    System.out.println("Durasi > 5 jam = Tarif tetap Rp 12.500");
32                } else {
33                    if (jenis == 1) { // mobil
34                        total += durasi * 3000;
35                        System.out.println("Mobil - Tarif: Rp " + (durasi * 3000));
36                    } else { // motor
37                        total += durasi * 2000;
38                        System.out.println("Motor - Tarif: Rp " + (durasi * 2000));
39                    }
40                }
41            } else {
42                // bukan 0, bukan 1, bukan 2 = input invalid
43                System.out.println("Jenis kendaraan tidak valid! Masukkan 1 atau 2, atau 0 untuk keluar.");
44            }
45        }
46
47        // Output
48        System.out.println("\n=== TOTAL PEMBAYARAN HARI INI ===");
49        System.out.println("Total pendapatan parkir: Rp " + total);
50        System.out.println("Selesai");
51
52        sc.close();
53    }
54 }
55

```

```

=== Program Penghitung Pembayaran Parkir ===
Tarif: Mobil = Rp 3.000/jam | Motor = Rp 2.000/jam
Durasi > 5 jam dikenai tarif tetap Rp 12.500
-----

```

```

Masukkan jenis kendaraan (1-Mobil, 2-Motor, 0-Keluar): 1
Masukkan durasi parkir (jam): 2
Mobil - Tarif: Rp 6000

```

```

Masukkan jenis kendaraan (1-Mobil, 2-Motor, 0-Keluar): 2
Masukkan durasi parkir (jam): 2
Motor - Tarif: Rp 4000

```

```

Masukkan jenis kendaraan (1-Mobil, 2-Motor, 0-Keluar): 1
Masukkan durasi parkir (jam): 6
Durasi > 5 jam ? Tarif tetap Rp 12.500

```

```

Masukkan jenis kendaraan (1-Mobil, 2-Motor, 0-Keluar): 2
Masukkan durasi parkir (jam): 6
Durasi > 5 jam ? Tarif tetap Rp 12.500

```

```

Masukkan jenis kendaraan (1-Mobil, 2-Motor, 0-Keluar): 0

```

```

=== TOTAL PEMBAYARAN HARI INI ===
Total pendapatan parkir: Rp 35000
Selesai

```

PS C:\Users\IdeaPad\OneDrive\Documents\Daspro Github\PraktikumDaspro>