

KAMIS, 2 OKTOBER 2025

**LAPORAN HASIL PRAKTIKUM**

Jobsheet 5



Disusun Oleh:

Rayhan Jofan Halim

Kelas 1H/TI

254107020230

**PROGRAM STUDI D-IV TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG  
2025**

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

Proses:

```
1 package Jobsheet7;
2
3 public class SiakadFor24 {
4
5     public static void main(String[] args) {
6
7         java.util.Scanner sc = new java.util.Scanner(System.in);
8         double nilai, tertinggi = 0, terendah = 100;
9         for (int i = 1; i <= 10; i++) {
10             System.out.print("Masukkan nilai ke-" + i + ": ");
11             nilai = sc.nextDouble();
12             if (nilai > tertinggi) {
13                 tertinggi = nilai;
14             }
15             if (nilai < terendah) {
16                 terendah = nilai;
17             }
18         }
19         System.out.println("Nilai tertinggi adalah: " + (int) tertinggi);
20         System.out.println("Nilai terendah adalah: " + (int) terendah);
21     }
22 }
23
24
```

Output:

```
PS C:\PratikumDaspro> c:; cd 'c:\PratikumDaspro\CodeDetailsInExceptionMessages' '-cp
d762151ea9e2cd8db38a56\redhat.java\j
Masukkan nilai ke-1: 76,5
Masukkan nilai ke-2: 82,3
Masukkan nilai ke-3: 62,1
Masukkan nilai ke-4: 88,4
Masukkan nilai ke-5: 65,9
Masukkan nilai ke-6: 67,9
Masukkan nilai ke-7: 90,1
Masukkan nilai ke-8: 55,3
Masukkan nilai ke-9: 73,7
Masukkan nilai ke-10: 78,6
Nilai tertinggi adalah: 90.1
Nilai terendah adalah: 55.3
```

### 2.1.3 Pertanyaan

1. Sebutkan dan tunjukkan masing-masing komponen perulangan FOR pada kode program Percobaan 1!
2. Mengapa variabel tertinggi diinisialisasi 0 dan terendah diinisialisasi 100? Apa yang terjadi jika variabel tertinggi diinisialisasi 100 dan terendah diinisialisasi 0?
3. Jelaskan fungsi dan alur kerja dari potongan kode berikut!

```

if (nilai > tertinggi) {
    tertinggi = nilai;
}
if (nilai < terendah) {
    terendah = 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!
5. Commit dan push hasil modifikasi Anda ke Github dengan pesan "Modifikasi Percobaan 1"

### Jawaban:

1.

- Inisialisasi (Initialization): `int i = 1`

Ini adalah bagian yang dieksekusi pertama kali dan hanya sekali. Bagian ini menginisialisasi variabel counter (penghitung) `i` dengan nilai awal 1.

- Kondisi (Condition): `i <= 10`

Ini adalah kondisi yang akan dicek sebelum setiap iterasi (perulangan) dimulai. Jika kondisi ini bernilai `true`, perulangan akan dijalankan. Jika bernilai `false`, perulangan akan berhenti. Dalam kasus ini, perulangan akan terus berjalan selama nilai `i` masih kurang dari atau sama dengan 10

- Pembaruan (Update/Increment): `i++`

Ini adalah bagian yang dieksekusi di akhir setiap iterasi. `i++` adalah singkatan dari `i = i + 1`, yang artinya menambah nilai `i` sebesar 1 setiap kali satu putaran perulangan selesai.

2. `tertinggi = 0` (Nilai Terendah yang Mungkin): Variabel tertinggi diatur ke nilai terendah yang mungkin diterima (0). Tujuannya adalah agar nilai pertama yang Anda masukkan (misalnya 85) pasti akan lebih besar dari 0 (if (`85 > 0`)). Dengan demikian, nilai 85 tersebut akan langsung menggantikan 0 sebagai tertinggi sementara. Jika tertinggi diinisialisasi terlalu besar (misal 1000), nilai input normal tidak akan pernah melewatinya.

`terendah = 100` (Nilai Tertinggi yang Mungkin): Variabel terendah diatur ke nilai tertinggi yang mungkin diterima (100). Tujuannya adalah agar nilai pertama yang Anda masukkan (misalnya 85) pasti akan lebih kecil dari 100 (if (`85 < 100`)). Dengan demikian, nilai 85 tersebut juga akan langsung menggantikan 100 sebagai terendah sementara.

```

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

```

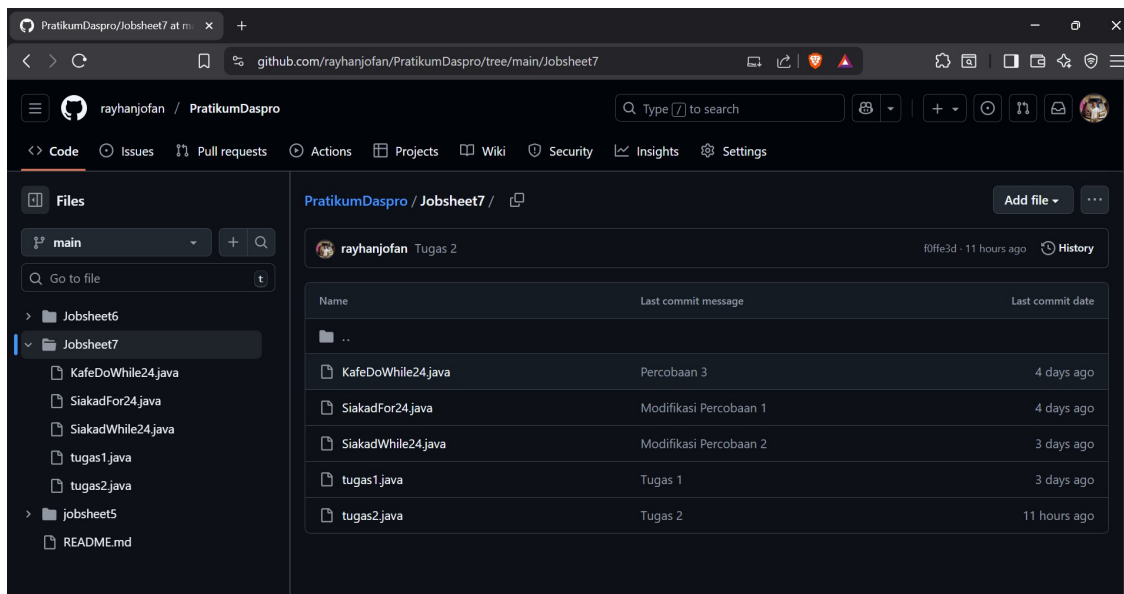
3.

Fungsi `if` pertama Program memeriksa apakah nilai input saat ini (`nilai`) lebih besar dari nilai tertinggi yang sudah tersimpan. Jika ya, nilai input baru tersebut akan menggantikan nilai tertinggi yang lama. `if` kedua program memeriksa apakah nilai input saat ini (`nilai`) lebih kecil dari nilai terendah yang sudah tersimpan. Jika ya, nilai input baru tersebut akan menggantikan nilai terendah yang lama.

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!

```
1  for (int i = 1; i <= 10; i++) {
2      System.out.print("Masukkan nilai ke-" + i + ": ");
3      nilai = sc.nextDouble();
4      if (nilai > tertinggi) {
5          tertinggi = nilai;
6      }
7      if (nilai < terendah) {
8          terendah = nilai;
9      }
10     if (nilai >= 60) {
11         System.out.println("Status: Lulus");
12         lulus++;
13     } else {
14         System.out.println("Status: Tidak Lulus");
15         tidakLulus++;
16     }
17 }
```

5. Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan1”



## 2.2 Percobaan 2: Studi Kasus Nilai Mahasiswa di SIAKAD – Perulangan WHILE

Proses:

```
1 package Jobsheet7;
2
3 public class SiakadWhile24 {
4     public static void main(String[] args) {
5
6         java.util.Scanner sc = new java.util.Scanner(System.in);
7
8         int nilai, jml, i = 0;
9
10        System.out.print("Masukan Jumlah Mahasiswa: ");
11        jml = sc.nextInt();
12        while (i < jml) {
13            System.out.print("Masukan Nilai Mahasiswa ke-" + (i + 1) + ": ");
14            nilai = sc.nextInt();
15
16            if (nilai < 0 || nilai > 100) {
17                System.out.println("Nilai tidak valid. Masukan lagi nilai yang valid!");
18                continue;
19            }
20            if (nilai > 80 && nilai <= 100) {
21                System.out.println("Nilai Mahasiswa ke-" + (i + 1) + " : A");
22            } else if (nilai > 73 && nilai <= 80) {
23                System.out.println("Nilai Mahasiswa ke-" + (i + 1) + " : B+");
24            } else if (nilai > 65 && nilai <= 73) {
25                System.out.println("Nilai Mahasiswa ke-" + (i + 1) + " : B");
26            } else if (nilai > 60 && nilai <= 65) {
27                System.out.println("Nilai Mahasiswa ke-" + (i + 1) + " : C+");
28            } else if (nilai > 50 && nilai <= 60) {
29                System.out.println("Nilai Mahasiswa ke-" + (i + 1) + " : C");
30            } else if (nilai > 39 && nilai <= 50) {
31                System.out.println("Nilai Mahasiswa ke-" + (i + 1) + " : D");
32            } else {
33                System.out.println("Nilai Mahasiswa ke-" + (i + 1) + " : E");
34            }
35            i++;
36        }
37    }
38 }
39
```

Output:

```
Masukan Jumlah Mahasiswa: 5
Masukan Nilai Mahasiswa ke-1: 85
Nilai Mahasiswa ke-1 : A
Masukan Nilai Mahasiswa ke-2: 63
Nilai Mahasiswa ke-2 : C+
Masukan Nilai Mahasiswa ke-3: 101
Nilai tidak valid. Masukan lagi nilai yang valid!
Masukan Nilai Mahasiswa ke-3: 23
Nilai Mahasiswa ke-3 : E
Masukan Nilai Mahasiswa ke-4: -15
Nilai tidak valid. Masukan lagi nilai yang valid!
Masukan Nilai Mahasiswa ke-4: 70
Nilai Mahasiswa ke-4 : B
Masukan Nilai Mahasiswa ke-5: 55
Nilai Mahasiswa ke-5 : C
```

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;  
}  
a. Nilai < 0 || nilai > 100  
- Maksud: Kondisi ini bernilai true jika nilai tersebut lebih kecil dari 0 atau jika nilai tersebut lebih besar dari 100

- Kegunaan: Untuk mendeteksi dan menandai input yang dianggap tidak sah atau di luar batas yang ditetapkan untuk nilai mahasiswa. Operator `||` memastikan blok kode di dalamnya akan dieksekusi jika salah satu dari kondisi tersebut terpenuhi.

b. Continue

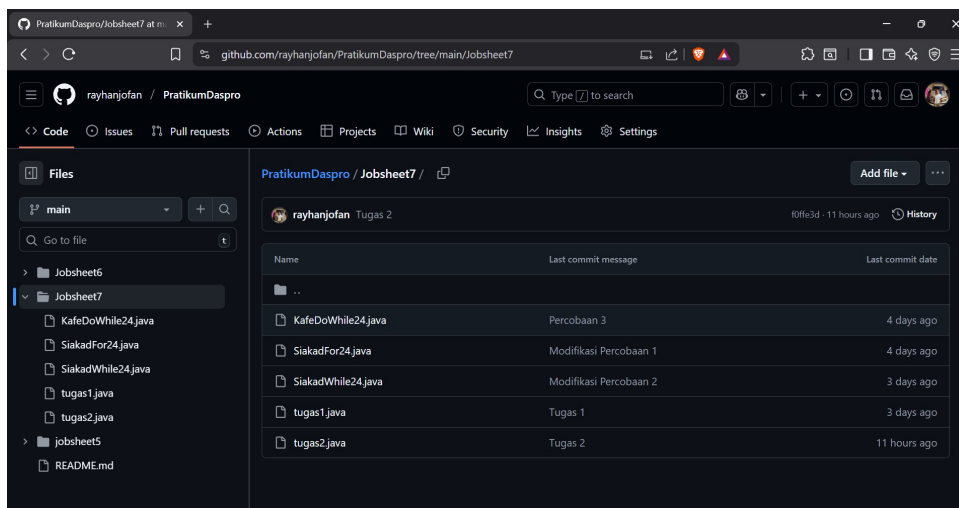
- Maksud: Memberi perintah kepada program untuk melewati atau mengabaikan sisa instruksi di dalam body perulangan untuk iterasi saat ini.

- Kegunaan: Dalam konteks ini, setelah menampilkan pesan kesalahan ("Nilai tidak valid..."), continue akan segera menghentikan iterasi yang sedang berjalan dan melompat ke bagian `i++` dari loop. Hal ini memastikan bahwa nilai yang tidak valid (di luar 0-100) tidak akan diproses oleh logika pencari nilai tertinggi/terendah dan logika kelulusan.

2. Mengapa sintaks `i++` dituliskan di akhir perulangan WHILE? Apa yang terjadi jika posisinya dituliskan di awal perulangan WHILE?
  - Jawaban: Sintaks `i++` dituliskan di akhir perulangan while untuk memastikan bahwa seluruh proses untuk satu mahasiswa selesai sebelum counter perulangan bertambah. Jika `i++` dipindahkan ke awal perulangan, tepat setelah while, maka logika perulangan akan rusak saat terjadi input nilai yang tidak valid.
3. Apabila jumlah mahasiswa yang dimasukkan adalah 19, berapa kali perulangan WHILE akan berjalan?
  - Jawaban: Perulangan while akan berjalan sebanyak 19 kali
4. Modifikasi kode program sehingga apabila terdapat mahasiswa yang mendapat nilai A, program menampilkan pesan tambahan "Bagus, pertahankan nilainya!"

```
1 if (nilai > 80 && nilai <= 100) {
2     System.out.println("Nilai Mahasiswa ke-" + (i + 1) + " : A");
3     System.out.println("Bagus, pertahankan nilainya!");
}
```

5. Commit dan push hasil modifikasi Anda ke Github dengan pesan "Modifikasi Percobaan2"



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

Proses:

```
1 package Jobsheet7;
2
3 public class KafeDowhile24 {
4
5     public static void main(String[] args) {
6
7         java.util.Scanner sc = new java.util.Scanner(System.in);
8         int kopi, teh, roti;
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
18            if (namaPelanggan.equalsIgnoreCase("batal")) {
19                System.out.println("Program dibatalkan.");
20                break;
21            }
22
23            System.out.print("Jumlah kopi: ");
24            kopi = sc.nextInt();
25
26            System.out.print("Jumlah teh: ");
27            teh = sc.nextInt();
28
29            System.out.print("Jumlah roti: ");
30            roti = sc.nextInt();
31
32            int totalHarga = (kopi * hargaKopi) + (teh * hargaTeh) + (roti * hargaRoti);
33            System.out.println("Total yang harus dibayar: Rp " + totalHarga);
34            sc.nextLine(); // Clear the buffer
35
36        } while (true);
37    }
38 }
39
40
```

Output:

```
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
Jumlah teh: 4
Jumlah roti: 2
Total yang harus dibayar: Rp 80000
Masukkan nama pelanggan (ketik 'batal' untuk keluar): BATAL
Program dibatalkan.
```

### Percobaan 3

#### Pertanyaan!

1. Pada penggunaan DO-WHILE ini, apabila nama pelanggan yang dimasukkan pertama kali adalah “batal”, maka berapa kali perulangan dilakukan?  
- Jawaban: Perulangan DO-WHILE akan berjalan satu kali
2. Sebutkan kondisi berhenti yang digunakan pada perulangan DO-WHILE tersebut!



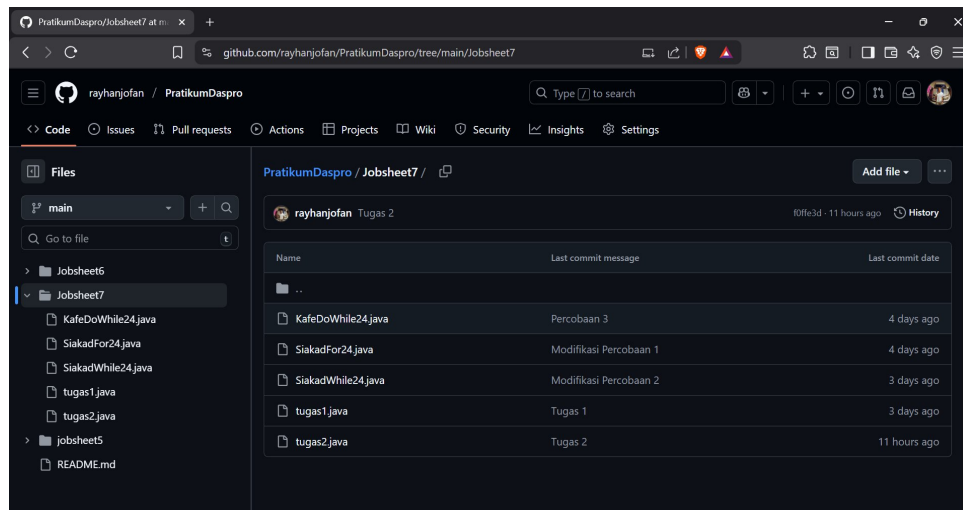
- Jawaban: Kondisi berhenti yang sebenarnya menghentikan perulangan adalah penggunaan kata kunci break; di dalam blok if.
3. Apa fungsi dari penggunaan nilai true pada kondisi DO-WHILE?
- Jawaban: Untuk menciptakan perulangan tak terbatas (infinite loop) secara sengaja.
4. Mengapa perulangan DO-WHILE tersebut tetap berjalan meskipun tidak ada komponen inisialisasi dan update?
- Jawaban: Karena perulangan ini berjalan berdasarkan kondisi kelanjutan yang disengaja (true), bukan berdasarkan kondisi hitungan yang harus mencapai batas tertentu.

## 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.

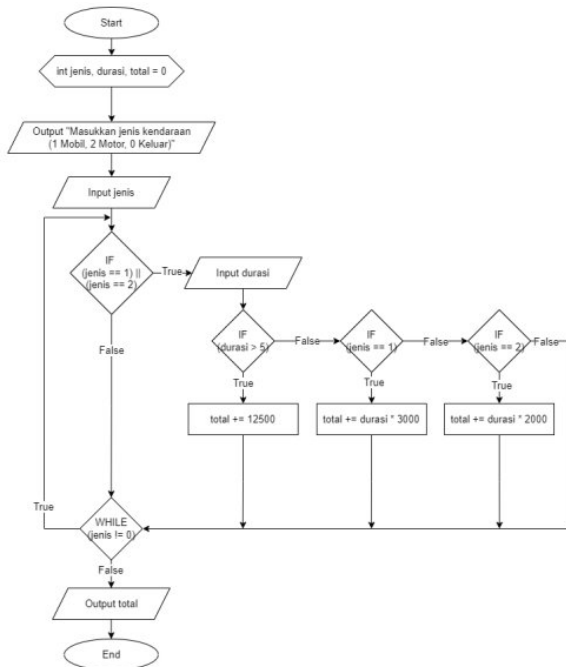
Commit dan push program Anda ke Github dengan pesan “Tugas 1”

Jawab :





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”

Jawab :

