

Nama : Khania Puji Auliya  
NIM : 254107020236  
Prodi : D-IV Teknik Informatika  
Kelas : 1G

### Percobaan 1

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”

Jawab:

1. Inisialisasi: I = 1  
Kondisi: I <= 10, nilai > tertinggi, nilai < terendah  
Update: i++
2. Variabel tertinggi diinisialisasi dengan 0 karena program harus membandingkan setiap nilai mahasiswa untuk mencari nilai yang paling tinggi. Dengan nilai awal 0, setiap nilai yang dimasukkan (karena pasti lebih besar atau sama dengan 0) dapat menggantikan nilai awal tersebut jika lebih tinggi. Sedangkan variabel terendah diinisialisasi dengan 100 karena program mencari nilai paling rendah. Dengan nilai awal 100, setiap nilai yang

dimasukkan (karena pasti lebih kecil atau sama dengan 100) dapat menggantikan nilai awal tersebut jika lebih rendah.

Jika posisi inisialisasi dibalik, yaitu `tertinggi = 100` dan `terendah = 0`, maka hasil perbandingan menjadi tidak sesuai. Nilai `tertinggi` tidak akan berubah karena tidak ada nilai yang lebih besar dari 100, dan `terendah` juga tidak akan berubah karena tidak ada nilai yang lebih kecil dari 0. Akibatnya, hasil akhir program akan salah karena selalu menampilkan nilai `tertinggi = 100` dan nilai `terendah = 0`, meskipun nilai sebenarnya berbeda.

### 3. Fungsi:

`if (nilai > tertinggi)`, Mengecek apakah nilai yang baru dimasukkan lebih besar daripada nilai `tertinggi` sebelumnya. Jika benar, maka nilai tersebut menjadi nilai `tertinggi` yang baru.

`if (nilai < terendah)`, Mengecek apakah nilai yang baru dimasukkan lebih kecil daripada nilai `terendah` sebelumnya. Jika benar, maka nilai tersebut menjadi nilai `terendah` yang baru.

Alur Kerja:

1. Program membaca satu nilai dari input pengguna.
2. Nilai tersebut dibandingkan dengan variabel `tertinggi`:
  - Jika lebih besar → nilai `tertinggi` diperbarui.
3. Nilai yang sama juga dibandingkan dengan variabel `terendah`:
  - Jika lebih kecil → nilai `terendah` diperbarui.
4. Langkah 1–3 diulangi untuk setiap nilai mahasiswa yang dimasukkan.
5. Setelah semua data selesai dimasukkan, variabel `tertinggi` menyimpan nilai paling tinggi, dan `terendah` menyimpan nilai paling rendah dari seluruh input.

```

File Edit Selection View Go Run Terminal Help
OPEN EDITORS
PRAKTIKUMDASPRO
jobsheet7 > J SisakadFor15.java > main(String[])
public class SisakadFor15 {
    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) {
                lulus++;
            } else {
                tidaklulus++;
            }
        }
        System.out.println("Nilai Tertinggi: " + tertinggi);
        System.out.println("Nilai Terendah: " + terendah);
        System.out.println("Jumlah Mahasiswa Lulus : " + lulus);
        System.out.println("Jumlah Mahasiswa Tidak Lulus : " + tidaklulus);
    }
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Masukkan nilai mahasiswa ke-1: 67  
 Masukkan nilai mahasiswa ke-2: 66  
 Masukkan nilai mahasiswa ke-3: 90  
 Masukkan nilai mahasiswa ke-4: 88  
 Masukkan nilai mahasiswa ke-5: 88  
 Masukkan nilai mahasiswa ke-6: 76  
 Masukkan nilai mahasiswa ke-7: 65  
 Masukkan nilai mahasiswa ke-8: 65  
 Masukkan nilai mahasiswa ke-9: 65  
 Masukkan nilai mahasiswa ke-10: 77  
 Nilai Tertinggi: 90.0  
 Nilai Terendah: 56.0  
 Jumlah Mahasiswa Lulus : 9  
 Jumlah Mahasiswa Tidak Lulus : 1

PS D:\PraktikumDaspro\PraktikumDaspro

4.

```

package jobsheet7;
import java.util.Scanner;

public class SisakadFor15 {
    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) {
                lulus++;
            } else {
                tidaklulus++;
            }
        }
        System.out.println("Nilai Tertinggi: " + tertinggi);
        System.out.println("Nilai Terendah: " + terendah);
        System.out.println("Jumlah Mahasiswa Lulus : " + lulus);
        System.out.println("Jumlah Mahasiswa Tidak Lulus : " + tidaklulus);
    }
}

```

5.

## Percobaan 2

- 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  $< 0 \parallel$  nilai  $> 100$
  - b. Continue
2. Mengapa sintaks `i++` dituliskan di akhir perulangan WHILE? Apa yang terjadi jika posisinya dituliskan di awal perulangan WHILE?
  3. Apabila jumlah mahasiswa yang dimasukkan adalah 19, berapa kali perulangan WHILE akan berjalan?
  4. Modifikasi kode program sehingga apabila terdapat mahasiswa yang mendapat nilai A, program menampilkan pesan tambahan "Bagus, pertahankan nilainya"!
  5. Commit dan push hasil modifikasi Anda ke Github dengan pesan "Modifikasi Percobaan 2"

Jawab:

1. Nilai  $< 0 \parallel$  nilai  $> 100$ : Pemilihan IF untuk mengecek valid atau tidaknya nilai yang dimasukkan, apakah nilai yang dimasukkan kurang dari 0 OR nilai lebih dari 100.  
Kegunaan: Validasi input  
Continue: menghentikan satu iterasi (dalam loop) jika kondisi tertentu terjadi, dan melanjutkan pada iterasi berikutnya dalam loop.  
Kegunaan: untuk melewati (skip) iterasi saat ini (1 iterasi saja)
2. Sintaks `i++` ditulis di akhir perulangan while agar nilai `i` diperbarui *setelah* kondisi perulangan diperiksa dan blok kode di dalamnya dieksekusi, sehingga memastikan kondisi perulangan dapat berubah dan perulangan dapat berakhir. Jika `i++` dipindahkan ke awal, perulangan akan berjalan secara tidak terhingga (infinite loop)  
karena `i` diperbarui *sebelum* kondisi perulangan pertama kali diperiksa, membuat kondisi tetap benar dan tidak pernah mencapai titik berhenti.
3. Perulangan WHILE akan berjalan selama 19 kali, karena perulangan WHILE akan berjalan selama kondisinya benar.

```

File Edit Selection View Go Run Terminal Help
OPEN EDITORS
PRAKTIKUMDASPRO
jobsheet7 > J_SiajadWhile15.java 1 x
public class SiajadWhile15 {
    public static void main(String[] args) {
        int nilai = sc.nextInt();
        if (nilai < 0 || nilai > 100) {
            System.out.print("Nilai tidak valid. Masukkan lagi nilai yang valid!\n");
            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++;
    }
}

```

PS D:\PraktikumDaspro\PraktikumDaspro> 8 "C:\IRK-24.0\IRK-24.0\bin\java.exe" "-Djava.awt.headless=true" -showCodeDetailsInExceptionMessages "-cp" "C:\Users\Evie Fitriana\AppData\Roaming\CodeBlocks\PraktikumDaspro\jobsheet7\SiajadWhile15"

Masukkan jumlah mahasiswa: 3  
 Masukkan nilai mahasiswa ke-1: 94  
 Nilai mahasiswa ke-1 adalah A  
 Bagus, pertahankan nilainya!  
 Masukkan nilai mahasiswa ke-2: 88  
 Nilai mahasiswa ke-2 adalah A  
 Bagus, pertahankan nilainya!  
 Masukkan nilai mahasiswa ke-3: 67  
 Nilai mahasiswa ke-3 adalah B

4.

```

package jobsheet7;
import java.util.Scanner;
public class SiajadWhile15 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int nilai, jumlah, i = 0;
        System.out.print("Masukkan jumlah mahasiswa: ");
        jumlah = sc.nextInt();
        while (i < jumlah) {
            System.out.print("Masukkan nilai mahasiswa ke-" + (i + 1) + ": ");
            nilai = sc.nextInt();
            if (nilai < 0 || nilai > 100) {
                System.out.print("Nilai tidak valid. Masukkan lagi nilai yang valid!\n");
                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 {
                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah E");
            }
            i++;
        }
    }
}

```

5.

### Percobaan 3

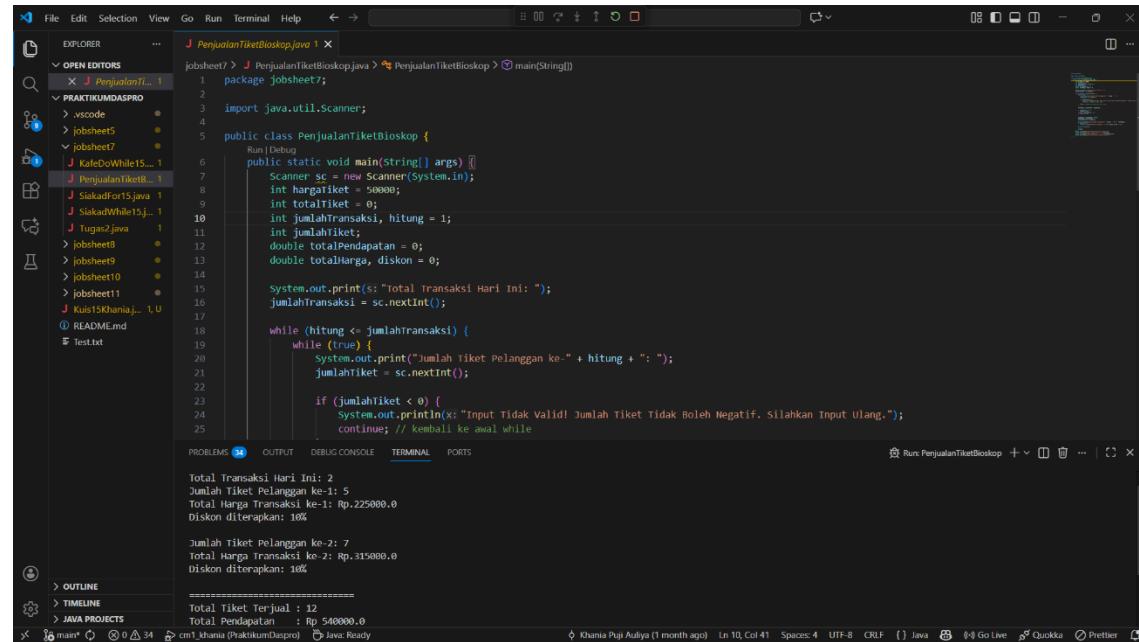
1. Pada penggunaan DO-WHILE ini, apabila nama pelanggan yang dimasukkan pertama kali adalah “batal”, maka berapa kali perulangan dilakukan?
2. Sebutkan kondisi berhenti yang digunakan pada perulangan DO-WHILE tersebut!
3. Apa fungsi dari penggunaan nilai true pada kondisi DO-WHILE?

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

Jawab:

1. Pada penggunaan DO-WHILE jika pertama kali dimasukkan “batal” maka perulangan akan dilakukan satu kali, karena pengecekan kondisi dilakukan setelah blok dijalankan. Jadi, meskipun nama yang ditulis “batal” program akan tetap menjalankannya, karena “batal” dilakukan setelah input.
  2. Kondisi berhenti jika pengguna menginput “batal”, karena jika input “batal” perulangan akan dihentikan menggunakan BREAK.
  3. Penggunaan true pada kondisi DO-WHILE yaitu digunakan untuk membuat perulangan tak terbatas (infinite loop), yang artinya perulangan tidak akan berhenti dengan sendirinya, dan hanya berhenti jika ada perintah break di dalam program.
  4. Perulangan DO-WHILE tetap berjalan meskipun tidak ada inisialisasi dan update karena kondisi while nya selalu true, perulangan tidak membutuhkan variabel counter seperti pada for, perulangan berhenti bukan karena update counter, tetapi karena break di dalam blok do.

## TUGAS



The screenshot shows a Java development environment with the following details:

- File Explorer:** Shows files like `Tugas2.java`, `jobsheet7.java`, `KafeDoWhile15.java`, etc.
- Code Editor:** Displays the `Tugas2.java` file containing a class that handles vehicle types and parking fees.
- Terminal:** Shows command-line interactions for testing the application.
- Output:** Shows the total payment for parking.

```
J Tugas2.java 1
jobsheet7 > J Tugas2.java > Tugas2 > main(String[])
package jobsheet7;
import java.util.Scanner;
public class Tugas2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int jenis, durasi;
        int total = 0;
        int biaya;
        System.out.print("Masukkan jenis kendaraan (1. Mobil, 2. Motor, 0. Keluar): ");
        jenis = sc.nextInt();
        while (jenis != 0) {
            String namakendaraan = "";
            if (jenis == 1) {
                namakendaraan = "Mobil";
            } else if (jenis == 2) {
                namakendaraan = "Motor";
            } else {
                System.out.println("Jenis kendaraan tidak valid!");
                System.out.print("Masukkan jenis kendaraan (1. Mobil, 2. Motor, 0. Keluar): ");
                jenis = sc.nextInt();
            }
        }
    }
}
Run Tugas2 + v | Run Tugas2 | ... | x
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Masukkan jenis kendaraan (1. Mobil, 2. Motor, 0. Keluar): 1  
Masukkan durasi parkir untuk Mobil (jam): 2  
Mobil parkir selama 2 jam = Rp 6000

Masukkan jenis kendaraan berikutnya (1. Mobil, 2. Motor, 0. Keluar): 2  
Masukkan durasi parkir untuk Motor (jam): 4  
Motor parkir selama 4 jam = Rp 8000

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

Total pembayaran parkir seluruh kendaraan: Rp 14000