

Nama : Rofiq Aristiyawan
NIM : 254107020060
Kelas/Absen : TI-1G/27

Percobaan 1

Jawaban:

1. Struktur umum perulangan **for** di java:
for (inisialisai, kondisi, increment/decrement) {
kode yg diulang-ulang
}

Penjelasan:

- a. Inisialisasi → `int i = 1`

Bagian yg diproses sekali aja di awal, yaitu mulai hitungan dari 1 (Mahasiswa ke-1), dan variable `i` digunakan untuk mengontrol perulangan.

- b. Kondisi → `i <= 10`

Ini yg dicek setiap kali sebelum loop dijalankan, selama kondisi ini benar maka program didalam akan terus dijalankan, kalau `i` sudah lebih dari 10 → kondisi false → loop akan berhenti, jadi loop akan terus jalan 10 kali (dari 1 sampai 10).

- c. Increment → `i++`

Bagian yg dijalankan setiap akhir satu siklus loop, `i++` arti nya nilai `i` bertambah 1 setelah perulangan, fungsi nya untuk pindah ke mahasiswa berikutnya.

- d. Blok kode → `{...}`

Semua yg ada di dalam `{...}` adalah isi loop, program yg diulang-ulang, kalau dari kondisi diatas maka akan di jalankan sebanyak 10 kali.

2. Kenapa variable tertinggi di inisialisasi 0 dan terendah di inisialisasi 100?

Karena ketika program mulai belum ada nilai yg dimasukkan, jadi program butuh patokan awal. Tertinggi di inisialisasi 0 arti nya nilai tertinggi sementara adalah 0.

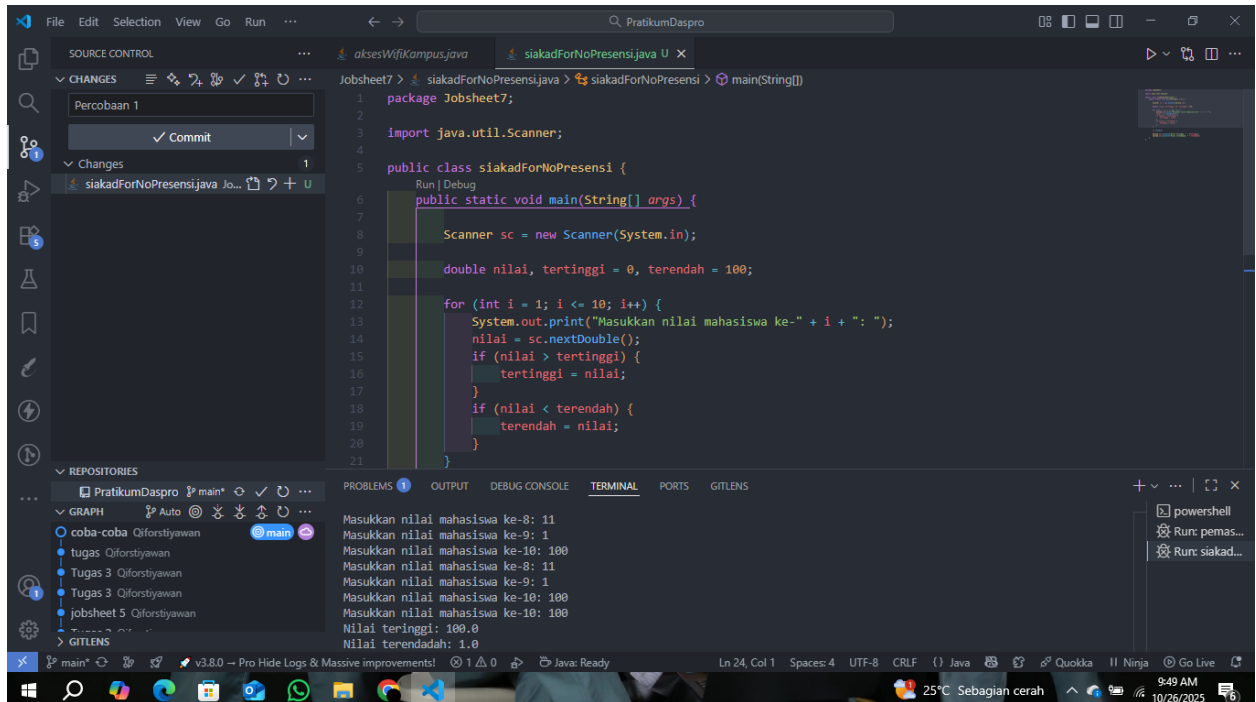
Ketika dimasukkan nilai, misalnya 40 maka akan dicek: `if (nilai > tertinggi) // apakah 40 > 0?`
Maka tertinggi = nilai // tertinggi menjadi 40.

Sama seperti tertinggi, terendah di inisialisasi 100 arti nya nilai terendah sementara adalah 100.

Ketika dimasukkan nilai, misalnya 20 maka akan dicek: `if (nilai < terendah) // apakah 20 < 100?`
Maka terendah = nilai // terendah menjadi 20.

Kalau tertinggi = 100 dan terendah = 0, maka hasil akhir program akan salah karena kondisi pembandingan tidak pernah terpenuhi. Program tidak akan memperbarui nilai tertinggi maupun terendah.

- Menentukan nilai tertinggi dan terendah dari data yg diinputkan, jadi setiap memasukkan nilai maka dua if akan langsung membandingkan nilai input dengan nilai sementara yg tersimpan di variable tertinggi dan terendah.
- Before



The screenshot shows the initial state of the code in VS Code. The file explorer on the left shows the project structure. The main editor displays the code for 'siakadForNoPresensi.java'. The code is as follows:

```
package Jobsheet7;

import java.util.Scanner;

public class siakadForNoPresensi {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        double nilai, tertinggi = 0, terendah = 100;

        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;

            }

        }

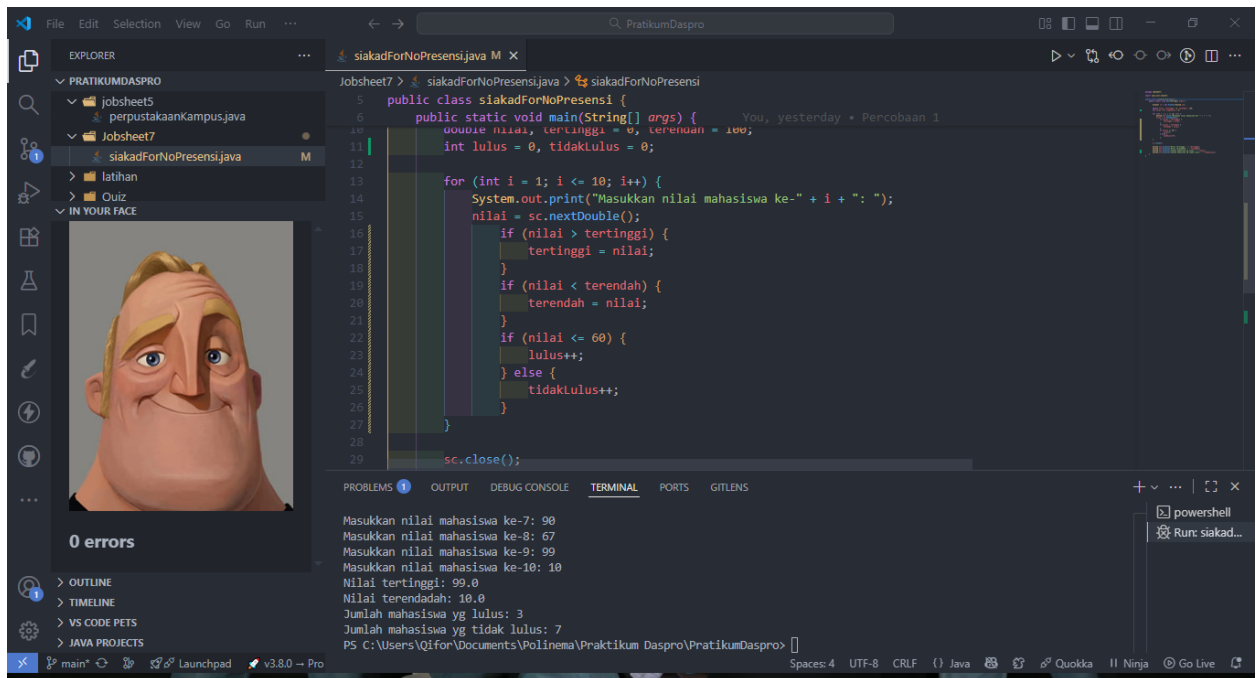
    }

}
```

The output window at the bottom shows the execution results:

```
Masukkan nilai mahasiswa ke-8: 11
Masukkan nilai mahasiswa ke-9: 1
Masukkan nilai mahasiswa ke-10: 100
Masukkan nilai mahasiswa ke-8: 11
Masukkan nilai mahasiswa ke-9: 1
Masukkan nilai mahasiswa ke-10: 100
Masukkan nilai mahasiswa ke-10: 100
Nilai tertinggi: 100.0
Nilai terendah: 1.0
```

After



The screenshot shows the modified code in VS Code. The file explorer on the left shows the project structure. The main editor displays the code for 'siakadForNoPresensi.java'. The code is as follows:

```
package Jobsheet7;

import java.util.Scanner;

public class siakadForNoPresensi {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        double nilai, tertinggi = 0, terendah = 100;

        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++;

            }

        }

        sc.close();

    }

}
```

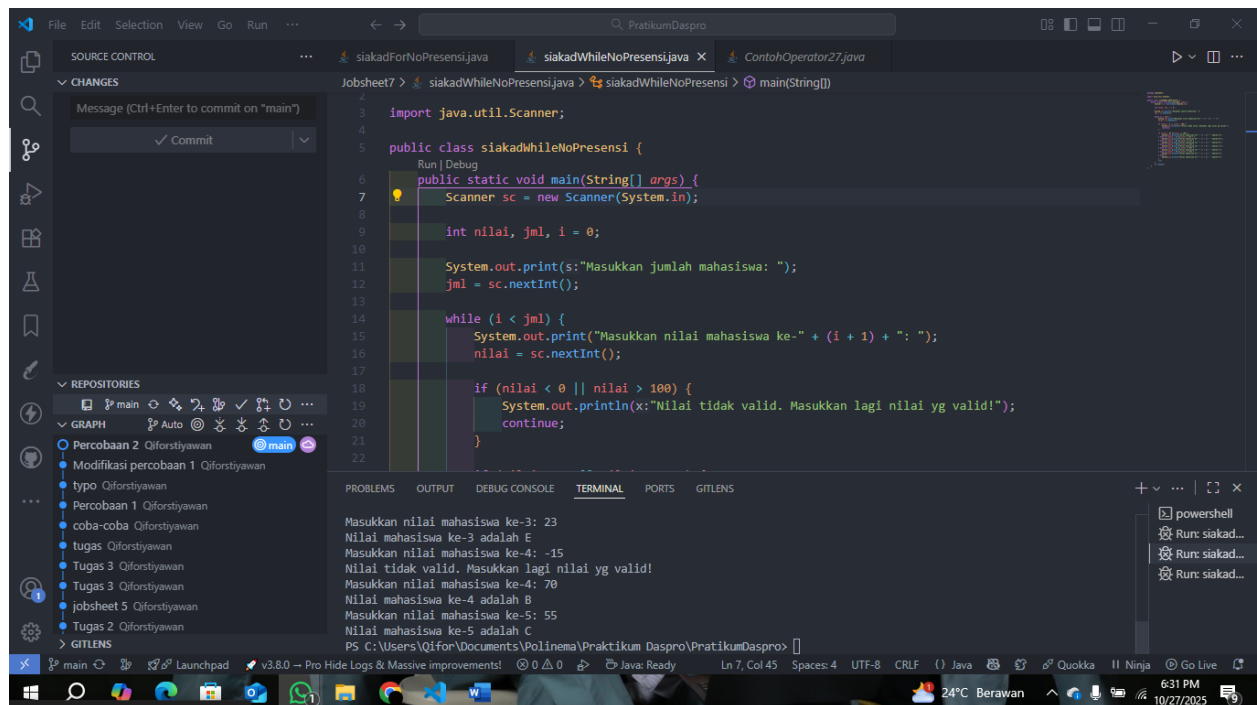
The output window at the bottom shows the execution results:

```
Masukkan nilai mahasiswa ke-7: 90
Masukkan nilai mahasiswa ke-8: 67
Masukkan nilai mahasiswa ke-9: 99
Masukkan nilai mahasiswa ke-10: 10
Nilai tertinggi: 99.0
Nilai terendah: 10.0
Jumlah mahasiswa yg lulus: 3
Jumlah mahasiswa yg tidak lulus: 7
PS C:\Users\Qifor\Documents\Polinema\Praktikum Daspro\PraktikumDaspro>
```

Percobaan 2

Jawaban:

- `nilai < 0 || nilai > 100`
Maksudnya, kondisi ini memeriksa apakah nilai yg diinput tidak valid, yaitu kurang dari 0 atau lebih dari 100. Jadi jika nilai lebih kecil dari 0 atau nilai lebih besar dari 100, maka nilai itu tidak valid.
 - `continue`
Digunakan untuk melewati sisa perintah di dalam loop saat ini, dan langsung lanjut ke iterasi berikutnya.
- Fungsi dari `i++` adalah menambah nilai variable `i` sebanyak 1. Dalam program itu, `i` berfungsi sebagai penghitung jumlah mahasiswa. kenapa ditulis di akhir karena logika loop nya adalah selama jumlah mahasiswa yg sudah diinput (`i`) belum mencapai batas (`jml`), terus ulang, dengan `i++` di akhir, selesai input, lanjut ke mahasiswa berikutnya dan perulangan berjalan pas sesuai jumlah mahasiswa.
Jika diawal, naik dulu baru input, mahasiswa pertama ke skip dan loop lebih cepat selesai.
- Jika `jml = 19`, maka perulangan akan berjalan sebanyak 19 kali dikarenakan kondisi awal: `i = 0`, kondisi berhenti: `i < jml`, yaitu perulangan akan berhenti ketika `i` tidak lagi kurang dari `jml`.
- Before



```
File Edit Selection View Go Run ...
SOURCE CONTROL
CHANGES
Message (Ctrl+Enter to commit on "main")
Commit
REPOSITORIES
main
GRAPH
Percobaan 2 Qiforstiyawan
Modifikasi percobaan 1 Qiforstiyawan
typo Qiforstiyawan
Percobaan 1 Qiforstiyawan
coba-coba Qiforstiyawan
tugas Qiforstiyawan
Tugas 3 Qiforstiyawan
Tugas 5 Qiforstiyawan
Tugas 2 Qiforstiyawan
GIT LENS
Jobsheet7 > siakadWhileNoPresensi.java > siakadWhileNoPresensi > main(String[])
import java.util.Scanner;

public class siakadWhileNoPresensi {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int nilai, jml, i = 0;

        System.out.print("Masukkan jumlah mahasiswa: ");
        jml = sc.nextInt();

        while (i < jml) {
            System.out.print("Masukkan nilai mahasiswa ke-" + (i + 1) + ": ");
            nilai = sc.nextInt();

            if (nilai < 0 || nilai > 100) {
                System.out.println("Nilai tidak valid. Masukkan lagi nilai yg valid!");
                continue;
            }
        }

        PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GIT LENS
        Masukkan nilai mahasiswa ke-3: 23
        Nilai mahasiswa ke-3 adalah E
        Masukkan nilai mahasiswa ke-4: -15
        Nilai tidak valid. Masukkan lagi nilai yg valid!
        Masukkan nilai mahasiswa ke-4: 70
        Nilai mahasiswa ke-4 adalah B
        Masukkan nilai mahasiswa ke-5: 55
        Nilai mahasiswa ke-5 adalah C
        PS C:\Users\Qifor\Documents\Polinema\Praktikum Daspro\PratikumDaspro>
```

After

```
package Jobsheet7;
import java.util.Scanner;

public class siakadWhileNoPresensi {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int nilai, jml, i = 0;

        System.out.print(s:"Masukkan jumlah mahasiswa: ");
        jml = sc.nextInt();

        while (i < jml) {
            System.out.print("Masukkan nilai mahasiswa ke-" + (i + 1) + ": ");
            nilai = sc.nextInt();

            if (nilai < 0 || nilai > 100) {
                System.out.println(x:"Nilai tidak valid. Masukkan lagi nilai yang valid!");
                continue;
            }
        }
    }
}
```

Masukkan jumlah mahasiswa: 5
Masukkan nilai mahasiswa ke-1: 90
Nilai mahasiswa ke-1 adalah A
Bagus, pertahankan nilainya!
Masukkan nilai mahasiswa ke-2: 54
Nilai mahasiswa ke-2 adalah C
Masukkan nilai mahasiswa ke-3: 46
Nilai mahasiswa ke-3 adalah D
Masukkan nilai mahasiswa ke-4: 53

Percobaan 3

Jawaban:

1. Karena do while selalu menjalankan blok minimal satu kali sebelum pengecekan kondisi , maka meskipun input pertama adalah 'batal', perulangan tetap berjalan sekali sebelum berhenti lewat break.
2. Kondisi berhenti pada perulangan do while tersebut adalah ketika pengguna memasukkan kata 'batal' sebagai nama pelanggan. Saat kondisi if (namaPelanggan.equalsIgnoreCase("batal")) terpenuhi, maka perintah break akan menghentikan perulangan meskipun while(true) bernilai benar.
3. Penggunaan true sebagai kondisi di do while (true) berfungsi untuk membuat loop yang secara otomatis terus-menerus berjalan sampai ada intervensi internal (seperti break) untuk berhenti.
4. Penggunaan true sebagai kondisi pada do while(true) berfungsi untuk menciptakan perulangan tak terbatas (infinite loop) yang hanya akan berhenti jika ada perintah dari dalam loop seperti break.

Pola ini sering digunakan pada program interaktif atau sistem berulang, di mana jumlah pengulangan belum dapat ditentukan di awal.