

JOBSHEET 7

Nama : Muhammad Ibnu Zauzi

NIM : 24410720016

ABSEN : 18

KELAS : 1D

Percobaan 1

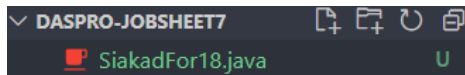
1. Buat repository baru pada akun Github Anda, beri nama daspro-jobsheet7



2. Lakukan cloning repository tersebut menggunakan perintah git clone dari terminal

```
HP@LAPTOP-VICTUS MINGW64 /c/MatKu1/DasarPemrograman/Pratikum
$ git clone https://github.com/ibnuzzi/daspro-jobsheet7.git
Cloning into 'daspro-jobsheet7'...
warning: You appear to have cloned an empty repository.
```

3. Buka folder repository tersebut menggunakan Visual Studio Code
4. Buat file baru, beri nama SiakadForNoAbsen.java



5. Buatlah struktur dasar program Java yang terdiri dari fungsi main().

```
public class SiakadFor18 {
    Run | Debug | Tabnine | Edit | Test | Explain | Document | Ask
    public static void main(String[] args) {

    }
}
```

6. Tambahkan library Scanner di bagian atas (luar) class

```
import java.util.Scanner;

public class SiakadFor18 {
```

7. Buat deklarasi Scanner dengan nama variabel sc di dalam fungsi main()

```
Scanner sc = new Scanner(System.in);
```

8. Deklarasikan variabel nilai, tertinggi, dan terendah bertipe double. Inisialisasi tertinggi dengan 0 dan terendah dengan 100

```
double nilai, tertinggi = 0, terendah = 100;
```

9. Buat struktur perulangan FOR dengan batas kondisi sesuai jumlah mahasiswa yaitu 10

```
for (int i = 1; i <= 10; i++) {

}
```

10. Di dalam perulangan FOR tersebut, tambahkan perintah untuk memasukkan nilai mahasiswa. Setelah itu, buat dua kondisi pemilihan secara terpisah untuk mengecek nilai tertinggi dan terendah dengan membandingkan nilai masukan dengan variabel tertinggi dan variabel terendah

```
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;  
    }  
}
```

11. Di luar perulangan FOR, tampilkan nilai tertinggi dan terendah

```
System.out.println("Nilai tertinggi: " + tertinggi);  
System.out.println("Nilai terendah: " + terendah);
```

12. Compile dan run program

```
Masukkan nilai mahasiswa ke-1: 70  
Masukkan nilai mahasiswa ke-2: 76  
Masukkan nilai mahasiswa ke-3: 78  
Masukkan nilai mahasiswa ke-4: 80  
Masukkan nilai mahasiswa ke-5: 88  
Masukkan nilai mahasiswa ke-6: 86  
Masukkan nilai mahasiswa ke-7: 90  
Masukkan nilai mahasiswa ke-8: 65  
Masukkan nilai mahasiswa ke-9: 60  
Masukkan nilai mahasiswa ke-10: 90  
Nilai tertinggi: 90.0  
Nilai terendah: 60.0
```

13. Commit dan push kode program ke Github

```
$ git add .  
  
HP@LAPTOP-VICTUS MINGW64 /c/MatKul/DasarP  
ain)  
$ git commit -m "percobaan-1"  
[main (root-commit) 06bf3d5] percobaan-1  
1 file changed, 22 insertions(+)  
create mode 100644 SiakadFor18.java  
  
HP@LAPTOP-VICTUS MINGW64 /c/MatKul/DasarP  
ain)  
$ git push  
Enumerating objects: 3, done.  
Counting objects: 100% (3/3), done.
```

Verifikasi Hasil Percobaan

```
Masukkan nilai mahasiswa ke-1: 76.5  
Masukkan nilai mahasiswa ke-2: 82.3  
Masukkan nilai mahasiswa ke-3: 62.1  
Masukkan nilai mahasiswa ke-4: 88.4  
Masukkan nilai mahasiswa ke-5: 65.9  
Masukkan nilai mahasiswa ke-6: 67.9  
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
```

Pertanyaan

1. `int i = 1`; sebagai **inisialisasi**, `i <= 10` sebagai **kondisi**, `i++` sebagai **update**, didalam for nantinya terdapat” **statement**”
2. Mengapa variabel tertinggi diinisialisasi 0 dan terendah diinisialisasi 100?
 - karena tidak ada nilai yang akan lebih kecil dari 0 dan tidak ada nilai yang akan lebih besar dari 100,nantinya setiap nilai input yang lebih besar dari 0 akan langsung menggantikan nilai var **tertinggi** sehingga dapat nilai terbesar dan setiap nilai input yang lebih kecil dari 100 akan langsung menggantikan var nilai **terendah**.

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

- Setiap nilai input yang lebih kecil dari 100 (yang seharusnya menjadi nilai tertinggi) tidak akan pernah menggantikan nilai **tertinggi** karena 100 lebih besar dari semua nilai tersebut dan nilai **tertinggi** akan tetap 100.
 - Setiap nilai input yang lebih besar dari 0 (yang seharusnya menjadi nilai terendah) tidak akan pernah menggantikan nilai **terendah** karena 0 lebih kecil dari semua nilai tersebut dan nilai **terendah** akan tetap 0.
3. fungsi dari alur kerja adalah
 - if pertama membandingkannya dulu dengan **nilai** yang diinputkan dan isi var nilai **tertinggi** adalah 0,apakah **nilai** yang diinputkan lebih dari var **tertinggi** jika nilainya **true**, nantinya setiap nilai input nilai yang lebih besar dari 0 akan langsung menggantikan nilai var **tertinggi** sehingga dapat nilai terbesar
 - if kedua membandingkannya dulu dengan **nilai** yang diinputkan dan isi var nilai **terendah** adalah 100, apakah **nilai** yang diinputkan kurang dari var **terendah** jika nilainya **true**, nantinya setiap nilai input yang lebih kecil dari 100 akan langsung menggantikan var nilai **terendah**.

```
double nilai, tertinggi = 0, terendah = 100, nilaiMinimal = 60;
int mahasiswa = 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 > nilaiMinimal) {
        mahasiswa += 1;
    }
}

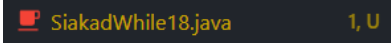
System.out.println("Nilai tertinggi: " + tertinggi);
System.out.println("Nilai terendah: " + terendah);
System.out.println("Jumlah mahasiswa lulus " + mahasiswa);
System.out.println("Jumlah mahasiswa tidak lulus " + (10 - mahasiswa));
```

4.

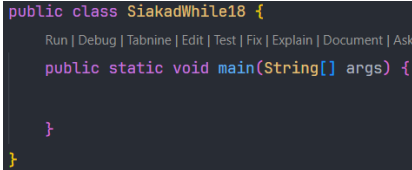
```
$ git add .
HP@LAPTOP-VICTUS MINGW64 /c/MatKu1/DasarPemrograman)
$ git commit -m "pertanyaan"
[main 3ce5234] pertanyaan
1 file changed, 7 insertions(+), 1 deletion(-)
HP@LAPTOP-VICTUS MINGW64 /c/MatKu1/DasarPemrograman)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
```
- 5.

Percobaan 2

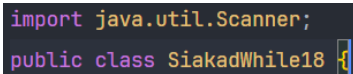
1. Buat file baru, beri nama SiakadWhileNoAbsen.java



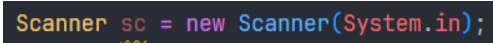
2. Buatlah struktur dasar program Java yang terdiri dari fungsi main().



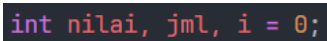
3. Tambahkan library Scanner di bagian atas (luar) class



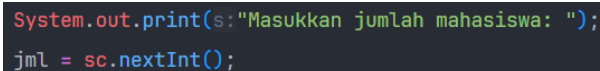
4. Buat deklarasi Scanner dengan nama variabel sc di dalam fungsi main()



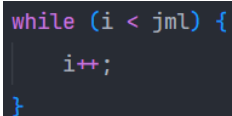
5. Deklarasikan variabel nilai, jml, dan i (untuk perulangan) bertipe integer. Inisialisasi i dengan 0 sebagai nilai awal perulangan



6. Tuliskan kode program untuk menerima input banyaknya mahasiswa yang disimpan ke variabel jml. Dengan demikian, batas perulangan akan dinamis sesuai masukan dari pengguna melalui keyboard.



7. Buat struktur perulangan WHILE dengan batas kondisi sesuai jumlah mahasiswa yaitu 5. Perhatikan simbol yang digunakan adalah < karena perulangan variabel i dimulai dari 0, bukan 1



8. Di dalam perulangan WHILE tersebut, tambahkan perintah untuk memasukkan nilai mahasiswa. Setelah itu, buat kondisi pemilihan IF untuk mengecek valid atau tidaknya nilai yang dimasukkan, dengan syarat nilai harus berada pada rentang 0 hingga 100. Kemudian tambahkan kondisi pemilihan IF-ELSE IF-ELSE untuk menampilkan kategori nilai huruf berdasarkan ketentuan



9. Compile dan run program

```
Masukkan jumlah mahasiswa: 5
Masukkan nilai mahasiswa ke-1: 80
Nilai mahasiswa ke-1 adalah B
Masukkan nilai mahasiswa ke-2: 88
Nilai mahasiswa ke-2 adalah A
Masukkan nilai mahasiswa ke-3: 76
Nilai mahasiswa ke-3 adalah B
Masukkan nilai mahasiswa ke-4: 65
Nilai mahasiswa ke-4 adalah C+
Masukkan nilai mahasiswa ke-5: 75
Nilai mahasiswa ke-5 adalah B
```

10. Commit dan push kode program ke Github

```
$ git add .
HP@LAPTOP-VICTUS MINGW64 /c/MatKu1/Dasa
ain)
$ git commit -m "percobaan-2"
[main 4bfbe58] percobaan-2
1 file changed, 38 insertions(+)
create mode 100644 SiakadWhile18.java
HP@LAPTOP-VICTUS MINGW64 /c/MatKu1/Dasa
ain)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
```

Pertanyaan

1. a. `nilai < 0 || nilai > 100`

Maksud :

Merupakan kondisi logika yang mengecek apakah nilai variable nilai berada di luar rentang 0 hingga 100.

Kegunaan :

- Kondisi akan bernilai true jika salah satu atau kedua kondisi berikut terpenuhi:
 - Nilai kurang dari 0
 - Nilai lebih dari 100
- Jika tidak ada kondisi yang terpenuhi, maka kondisi akan bernilai false.

b. `continue`

Maksud :

digunakan untuk **melewati (skip)** iterasi saat ini (1 iterasi saja) dan melanjutkan ke iterasi berikutnya.

Kegunaan :

Continue digunakan untuk menghentikan proses lebih lanjut pada iterasi saat ini jika ditemukan nilai yang tidak valid dan jika nilai tidak berada dalam rentang 0 hingga 100, maka program akan mencetak pesan kesalahan dan langsung melanjutkan ke iterasi berikutnya untuk meminta input nilai yang baru.

2. Karena nilai `i` hanya akan bertambah 1 setelah semua kondisi grade diperiksa, termasuk ketika nilai tidak valid sehingga program akan terus meminta input untuk mahasiswa ke-1 hingga nilai yang dimasukkan valid.
Jika ditulis diawal maka ketika nilai tidak valid maka akan tetap melanjutkan ke mahasiswa selanjutnya.
3. sebanyak 19 kali untuk memproses data dari 19 mahasiswa.

- ```

if (nilai > 80 && nilai ≤ 100) {
 System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah A");
 System.out.println(x:"Bagus, pertahankan nilainya");
}

```
4.

```

$ git add .

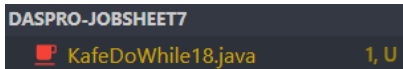
HP@LAPTOP-VICTUS MINGW64 /c/MatKu1/DasarPemrograman
$ git commit -m "pertanyaan"
[main f02a1b7] pertanyaan
1 file changed, 4 insertions(+), 2 deletions(-)

HP@LAPTOP-VICTUS MINGW64 /c/MatKu1/DasarPemrograman
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.

```
- 5.

### Percobaan 3

1. Buat file baru, beri nama KafeDoWhileNoAbsen.java



2. Buatlah struktur dasar program Java yang terdiri dari fungsi main().

```

public class KafeDoWhile18 {
 Tabnine | Edit | Test | Explain | Document | Ask | Run | Debug
 public static void main(String[] args) {

 }
}

```

3. Tambahkan library Scanner di bagian atas (luar) class

```
import java.util.Scanner;
```

4. Buat deklarasi Scanner dengan nama variabel sc di dalam fungsi main()

```
Scanner sc = new Scanner(System.in);
```

5. Deklarasikan variabel kopi, teh, dan roti bertipe integer untuk menampung banyaknya item yang dibeli pelanggan, serta namaPelanggan bertipe String. Deklarasi dan inisialisasi hargaKopi dengan 12000, hargaTeh dengan 7000, hargaRoti dengan 20000.

```
int kopi, teh, roti, hargaKopi = 12000, hargaTeh = 7000, hargaRoti = 20000, totalHarga;
String namaPelanggan;
```

6. Buat struktur perulangan DO-WHILE dengan kondisi true

```
do {

} while (true);
```

7. Di dalam perulangan DO-WHILE tersebut, tambahkan perintah untuk memasukkan namaPelanggan. Kemudian tambahkan kondisi IF untuk mengecek isi variabel namaPelanggan. Selanjutnya, tambahkan perintah untuk memasukkan banyaknya item yang dibeli pelanggan untuk setiap menu, apabila masukan nama pelanggan bukan "batal". Hitung total harga pembelian dan tampilkan hasilnya.

```

do {
 System.out.println("Masukkan nama pelanggan (ketik 'batal' untuk keluar): ");
 namaPelanggan = sc.nextLine();
 if (namaPelanggan.equalsIgnoreCase("batal")) {
 System.out.println("Transaksi dibatalkan.");
 break;
 }

 System.out.print("Jumlah kopi: ");
 kopi = sc.nextInt();
 System.out.print("Jumlah teh: ");
 teh = sc.nextInt();
 System.out.print("Jumlah roti: ");
 roti = sc.nextInt();

 totalHarga = (kopi * hargaKopi) + (teh * hargaTeh) + (roti * hargaRoti);
 System.out.println("Total yang harus dibayar: Rp " + totalHarga);
 sc.nextLine();
} while (true);

```

#### 8. Compile dan run program

```

Masukkan nama pelanggan (ketik 'batal' untuk keluar):
Muhammad Ibnu Zauzi
Jumlah kopi: 3
Jumlah teh: 2
Jumlah roti: 4
Total yang harus dibayar: Rp 130000
Masukkan nama pelanggan (ketik 'batal' untuk keluar):
batal
Transaksi dibatalkan.

```

#### 9. Commit dan push kode program ke Github

```

$ git add .
HP@LAPTOP-VICTUS MINGW64 /c/MatKul/Dasa
ain)
$ git commit -m "percobaan3"
[main ca4b86a] percobaan3
1 file changed, 28 insertions(+)
create mode 100644 KafeDowhile18.java
HP@LAPTOP-VICTUS MINGW64 /c/MatKul/Dasa
ain)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.

```

### Langkah-langkah Percobaan

```

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
Transaksi dibatalkan.

```

### Pertanyaan

1. Perulangan DO-WHILE hanya akan berjalan sebanyak 1 kali dan menampilkan output Transaksi dibatalkan karena disitu terdapat perintah **break** untuk memberhentikan perulangan DO-WHILE, meskipun kondisi dalam while(true) .
2. Kondisinya menggunakan break

3. Agar kondisi didalam perulangan akan terus menerus berjalan karena diberikan kondisi **true**, perulangan tersebut akan berhenti Ketika perintah break berhasil dieksekusi, jadi selagi perintah break tidak terpenuhi maka perulangan akan terus berjalan.
4. Karena dalam kondisi while terdapat nilai true . Ini artinya, kondisi akan selalu bernilai benar maka perulangan akan terus berjalan , kecuali ada perintah break yang menghentikannya.

## Tugas

```
do {
 System.out.print(s:"Masukkan nama pelanggan (ketik selesai untuk keluar): ");
 namaPelanggan = sc.next();
 sc.nextLine();
 if (namaPelanggan.equalsIgnoreCase(anotherString:"selesai")) {
 break;
 }

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

 if (tiket < 0) {
 System.out.println(x:"Jumlah tiket tidak valid (negatif). Masukkan lagi!");
 continue;
 }

 if (tiket > 10) {
 diskon = 0.15;
 } else if (tiket > 4) {
 diskon = 0.1;
 } else {
 diskon = 0;
 }

 hargaDiskon = hargaTiket * diskon;
}
```

1.

```
do {
 System.out.print(s:"Masukkan jenis kendaraan (1 Mobil, 2 Motor, 0 Keluar):");
 jenis = sc.nextInt();

 if (jenis == 1 || jenis == 2) {
 System.out.print(s:"Masukkan durasi parkir (dalam jam): ");
 durasi = sc.nextInt();

 if (durasi > 5) {
 total += 12500;
 } else {
 if (jenis == 1) {
 total += durasi * 3000;
 } else if (jenis == 2) {
 total += durasi * 2000;
 }
 }
 }
} while (jenis != 0);
```

2.

```
System.out.println("Total pembayaran parkir: Rp " + total);
```