

## JOBSHEET 6

### SINTAKS PERULANGAN 1

#### 1. Kompetensi

Mahasiswa memahami serta mampu membuat program dalam bahasa Java menggunakan instruksi perulangan.

#### 2. Alat Dan Bahan:

1. PC/Laptop
2. JDK
3. Text editor (Sublime)

#### 3. Ulasan Teori:

*Loop* adalah suatu blok atau kelompok instruksi yang dilaksanakan secara berulang-ulang. Perulangan yang disebut juga repetition akan membuat efisiensi proses dibandingkan jika dioperasikan secara manual.

Perulangan yang dijelaskan pada *jobsheet* ini adalah :

- Perulangan dengan **for**
- Perulangan dengan **while**
- Perulangan dengan **do-while**

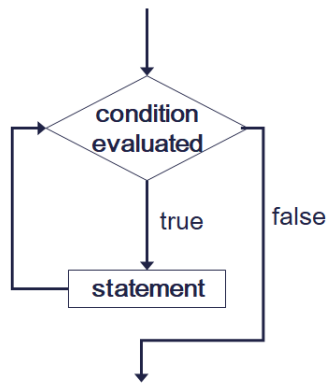
Penulisan sintaks ketiga jenis perulangan tersebut dibedakan sebagai berikut :

```
for(inisialisasi; kondisi; update)
{
    statement;
    ...
}
```

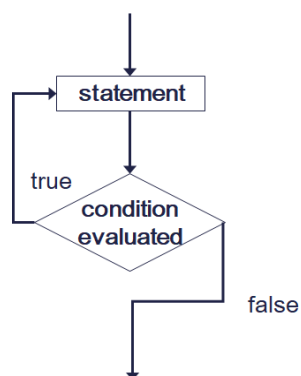
```
while(kondisi) {
    statement;
    ...
}
```

```
do {
    statement;
    ...
} while(kondisi);
```

Ketiga jenis *loop* tersebut sama-sama memiliki kondisi yang merupakan batasan suatu perulangan dilakukan. Cara kerja *loop* menggunakan **for** dan **while** dijelaskan pada Gambar 8.1. Batasan yang menjadi kondisi suatu perulangan didefinisikan dulu di awal, kemudian dilanjutkan dengan *statement* yang harus di-*looping*. Berbeda dengan **for** dan **while**, Gambar 8.2 menjelaskan tentang alur perulangan menggunakan **do-while**. Setelah inisialisasi dilakukan, akan di proses dulu *statement* yang harus dijalankan, baru kemudian dilakukan pembatasan *looping* dalam penulisan kondisi.



Gambar 6.1 Flowchart perulangan *for* dan *while*



Gambar 6.2 Flowchart perulangan *do-while*

#### 4. Langkah Praktikum:

1. Tulis ulang program untuk melakukan perulangan sebagai berikut :

##### a) Perulangan dengan **for**

```

public static void main(String[] args){
    Scanner sc = new Scanner(System.in);
    int angka, fac, i;
    System.out.println("===== PROGRAM MENGHITUNG NILAI FAKTORIAL =====");
    System.out.print("Masukkan suatu angka : ");
    angka = sc.nextInt();
    fac = 1;
    for(i=1; i<=angka; i++){
        fac = fac*i;
    }
    System.out.printf("nilai faktorial angka tersebut adalah: %d", fac);
}
    
```

##### b) Perulangan dengan **while**

```
public static void main(String[] args){
    Scanner sc = new Scanner(System.in);
    int angka, fac, i;
    System.out.println("===== PROGRAM MENGHITUNG NILAI FAKTORIAL =====");
    System.out.print("Masukkan suatu angka : ");
    angka = sc.nextInt();
    fac = 1;
    i = 1;
    while(i<=angka){
        fac = fac*i;
        i++;
    }
    System.out.printf("nilai faktorial angka tersebut adalah: %d", fac);
}
```

c) Perulangan dengan **do-while**

```
public static void main(String[] args){
    Scanner sc = new Scanner(System.in);
    int angka, fac, i;
    System.out.println("===== PROGRAM MENGHITUNG NILAI FAKTORIAL =====");
    System.out.print("Masukkan suatu angka : ");
    angka = sc.nextInt();
    fac = 1;
    i = 1;
    do{
        fac = fac*i;
        i++;
    } while(i<=angka);
    System.out.printf("nilai faktorial angka tersebut adalah: %d", fac);
}
```

2. Cocokkan hasil *running* program yang sudah Anda buat apakah sudah sesuai dengan tampilan berikut ini?

```
===== PROGRAM MENGHITUNG NILAI FAKTORIAL =====
Masukkan suatu angka : 5
nilai faktorial angka tersebut adalah: 120
```

3. Salinlah program perulangan dengan menggunakan *break* berikut :

a) Perulangan dengan **for**

```
public static void main(String[] args){
    Scanner sc = new Scanner(System.in);
    int angka, b;
    System.out.println("===== PROGRAM LOOP DENGAN BREAK =====");
    for(b=0;true;) {
        System.out.print("Masukkan angka : ");
        angka = sc.nextInt();
        b += angka;
        if(b>50) break;
    }
    System.out.printf("Angka berhenti pada jumlah angka: %d", b);
}
```



b) Perulangan dengan **while**

```
public static void main(String[] args){
    Scanner sc = new Scanner(System.in);
    int angka, b;
    System.out.println("===== PROGRAM LOOP DENGAN BREAK =====");
    b = 0;
    while(true) {
        System.out.print("Masukkan angka : ");
        angka = sc.nextInt();
        b += angka;
        if(b>50) break;
    }
    System.out.printf("Angka berhenti pada jumlah angka: %d", b);
}
```

c) Perulangan dengan **do-while**

```
public static void main(String[] args){
    Scanner sc = new Scanner(System.in);
    int angka, b;
    System.out.println("===== PROGRAM LOOP DENGAN BREAK =====");
    b = 0;
    do {
        System.out.print("Masukkan angka : ");
        angka = sc.nextInt();
        b += angka;
        if(b>50) break;
    } while(true);
    System.out.printf("Angka berhenti pada jumlah angka: %d", b);
}
```

4. Cocokkan hasil *running* program *looping* menggunakan *break* yang sudah Anda buat apakah sudah sesuai dengan tampilan berikut ini?

```
===== PROGRAM LOOP DENGAN BREAK =====
Masukkan angka : 4
Masukkan angka : 3
Masukkan angka : 20
Masukkan angka : 32
Angka berhenti pada jumlah angka: 59
```

5. Salinlah program perulangan dengan menggunakan *continue* berikut :



```
public static void main(String[] args){
    Scanner sc = new Scanner(System.in);
    int angka, b, i, count;
    double avg;
    System.out.println("===== PROGRAM LOOP DENGAN CONTINUE =====");
    b = 0;
    count = 0;
    for(i=0; i<5; i++) {
        System.out.print("Masukkan angka : ");
        angka = sc.nextInt();
        if(angka>=40) continue;
        b += angka;
        count++;
    }
    avg = (double)b/count;
    System.out.printf("Rata-rata angka kurang dari 40: %.2f", avg);
}
```

6. Cocokkan hasil *running* program *looping* menggunakan *continue* yang sudah Anda buat apakah sudah sesuai dengan tampilan berikut ini?

```
===== PROGRAM LOOP DENGAN CONTINUE =====
Masukkan angka : 40
Masukkan angka : 60
Masukkan angka : 32
Masukkan angka : 25
Masukkan angka : 10
Rata-rata angka kurang dari 40: 22.33
```

## 5. Pertanyaan

1. Misalkan, Anda diminta membuat sebuah program Java yang meminta masukan sebuah bilangan bulat *n*. Kemudian, program menampilkan karakter '\*' di layar sebanyak *n* kali. Manakah di antara kedua potongan program di bawah ini yang lebih baik dan aman ? mengapa ?

<pre>/* misal: masukan user n sudah ditampung di variabel integer n */ int i = 0;  while (i &lt; n) {     System.out.print("*");     i++; }</pre>	<pre>/* misal: masukan user n sudah ditampung di variabel integer n */ int i = 0;  while (i != n) {     System.out.print("*");     i++; }</pre>
---	---

2. Apakah *output* dari ketiga potongan program di bawah ini:

<pre>int r = 1; int i = 1; int a = 2; int n = 4;  while (i &lt;= n) {     r = r * a;     i++; } System.out.print(r);</pre>	<pre>int n = 7; boolean stop = false;  int i = 1; while (!stop) {     if (i &gt;= n) {         stop = true;     } else {         if (i % 2 == 0) {             System.out.print("#");         } else {             System.out.print("*");         }         i++;     } }</pre>	<pre>int n = 1892; int sum = 0;  while (n &gt; 0) {     int digit = n % 10;     sum = sum + digit;     n = n / 10; } System.out.println(sum);</pre>
--	--	---

## 6. Tugas

1. **(SumGenap)** Buatlah program dengan menggunakan bahasa Java yang meminta masukan *user* sebuah bilangan bulat **N** ( $N > 0$ ). Program kemudian menampilkan penjumlahan N bilangan genap positif pertama (**bilangan genap  $\geq 0$** ).

Contoh:

- Jika *user* memasukkan  $N = 3$ , itu artinya :  $0 + 2 + 4 = 6$
- Jika *user* memasukkan  $N = 5$ , itu artinya :  $0 + 2 + 4 + 6 + 8 = 20$

Silakan Anda rancang sendiri untuk tampilan programnya (cukup tampilkan hasil penjumlahannya saja).

2. **(SumKuadrat)** Buatlah sebuah program dengan menggunakan bahasa Java yang meminta masukan *user* sebuah bilangan bulat **N** ( $N > 0$ ). Kemudian, program menampilkan penjumlahan **N** bilangan kuadrat pertama. Bilangan kuadrat adalah =  $\{1, 4, 9, 16, 25, 36, \dots, N^2\}$ .

Contoh:

- Jika *user* memasukkan  $N = 2$ , itu artinya :  $1 + 4 = 5$
- Jika *user* memasukkan  $N = 3$ , itu artinya :  $1 + 4 + 9 = 14$

3. **(OddDigit)** Buatlah sebuah program dengan menggunakan bahasa Java yang meminta masukan *user* sebuah bilangan bulat **N** ( $N > 0$ ). Program kemudian memeriksa setiap digit yang ada di angka tersebut, dan menampilkan berapa jumlah digit yang ganjil dari bilangan **N** tersebut.

Contoh:

- Jika *user* memasukkan  $N = 2345$ , jumlah digit yang ganjil = 2
- Jika *user* memasukkan  $N = 993312$ , jumlah digit yang ganjil = 5