# JOBSHEET 6

# SINTAKS PERULANGAN 1

## Kompetensi

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

## Alat Dan Bahan:

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

## 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 :

do {

statement;

…

} while(*kondisi*);

while(*kondisi*) {

statement;

…

}

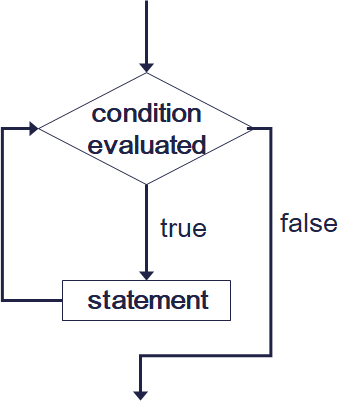
for(*inisialisasi; kondisi; update*) {

statement;

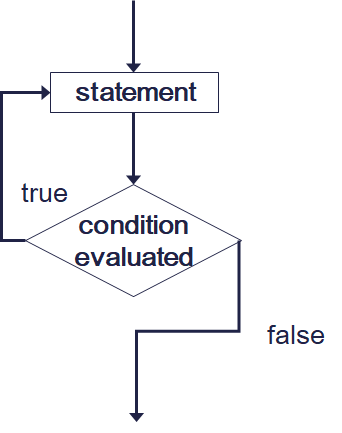
…

}

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.



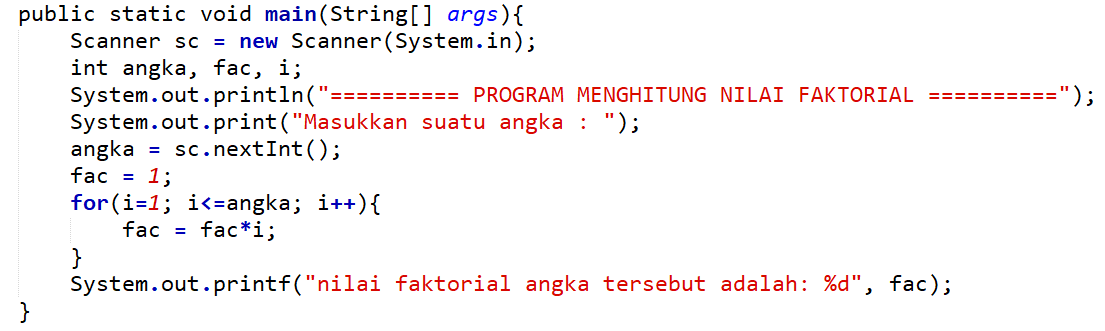
Gambar 6.1 *Flowchart* perulangan *for* dan *while*



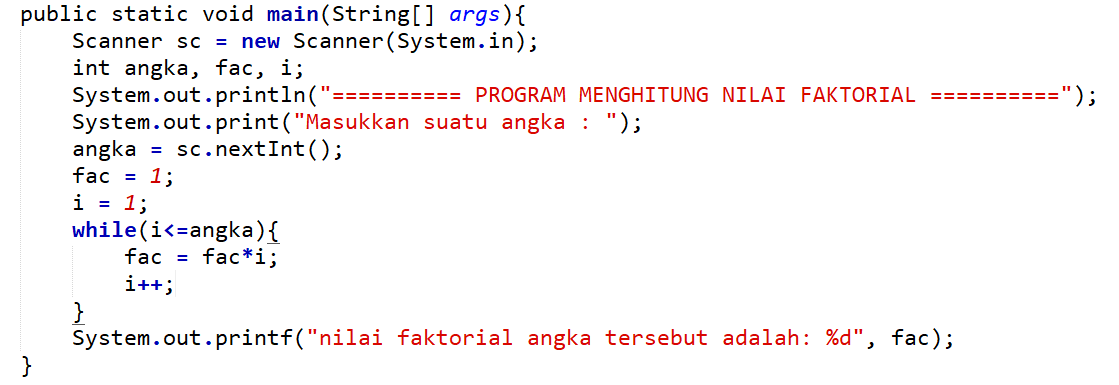
Gambar 6.2 *Flowchart* perulangan *do-while*

## Langkah Praktikum:

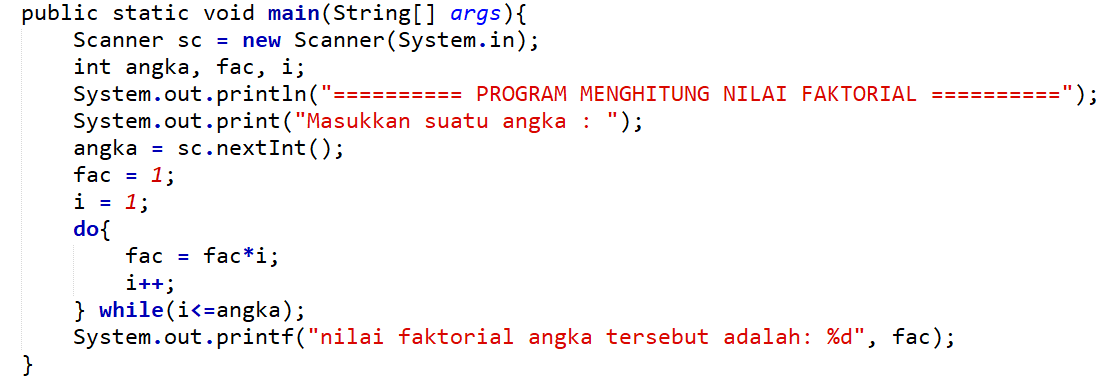
1. Tulis ulang program untuk melakukan perulangan sebagai berikut :
2. Perulangan dengan **for**



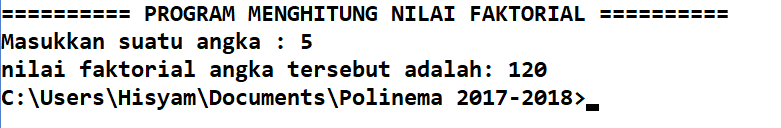
1. Perulangan dengan **while**



1. Perulangan dengan **do-while**

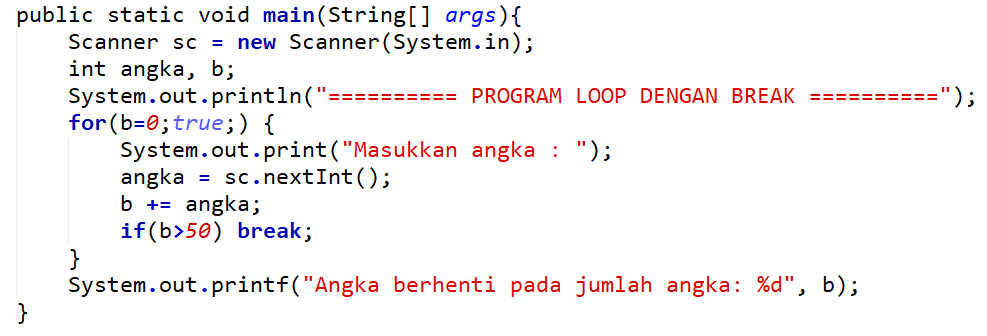


1. Cocokkan hasil *running* program yang sudah Anda buat apakah sudah sesuai dengan tampilan berikut ini?

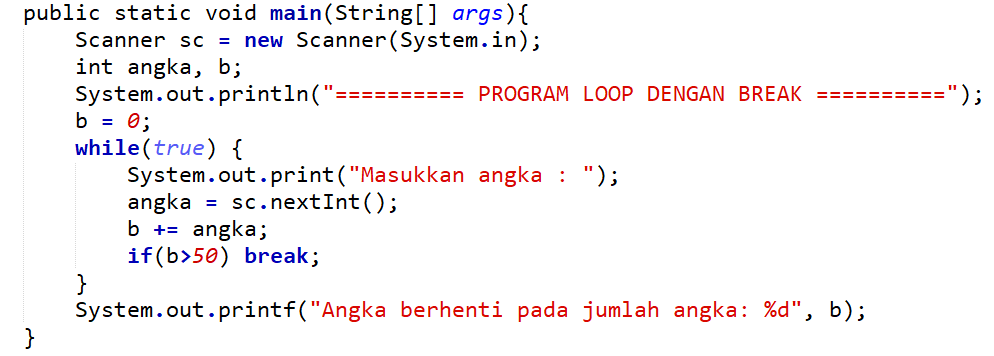


3. Salinlah program perulangan dengan menggunakan *break* berikut :

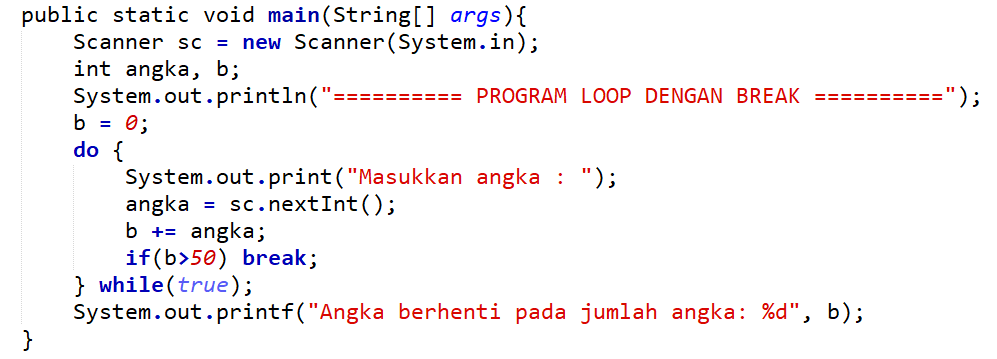
1. Perulangan dengan **for**



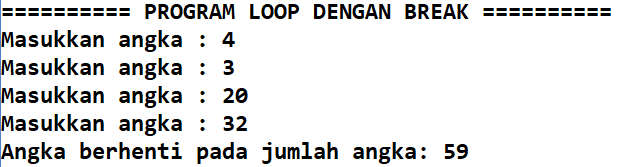
1. Perulangan dengan **while**



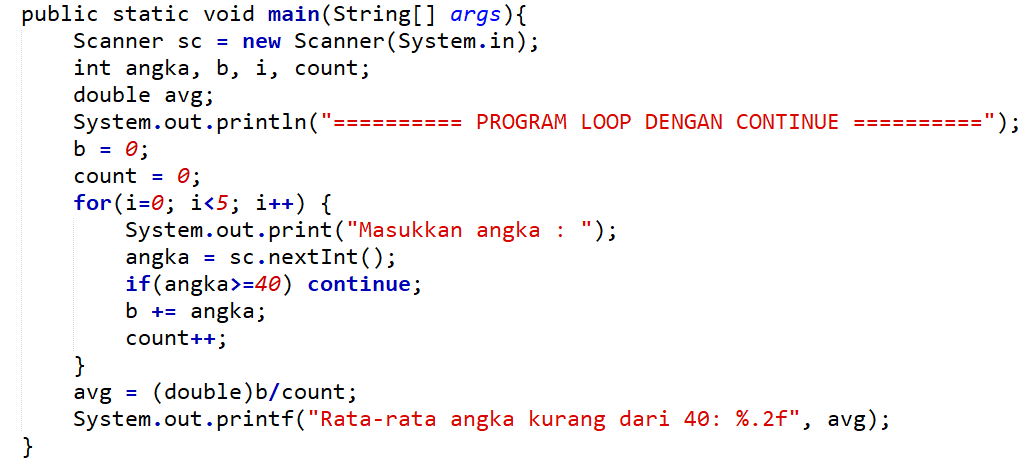
1. Perulangan dengan **do-while**



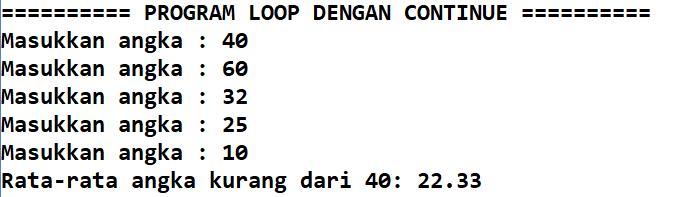
1. Cocokkan hasil *running* program *looping* menggunakan *break* yang sudah Anda buat apakah sudah sesuai dengan tampilan berikut ini?



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



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



## 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 ?

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

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

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

## 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 ≥ 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).

1. **(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, ...., **N**2}.

Contoh:

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

1. **(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