**LAPORAN JOBSHEET 6**

**Dasar Pemrograman**

**Perulangan 1**

Dibina oleh Ibu Vivi Nur Wijayaningrum, S.Kom., M.Kom.

Logo

Description automatically generated

Disusun oleh :

Fitria Ramadhani Prihandiva

D-IV Sistem Informasi Bisnis 1C / 14

2241760055

**POLITEKNIK NEGERI MALANG**

**JURUSAN TEKNOLOGI INFORMASI**

**TAHUN 2023**

Nama : Fitria Ramadhani Prihandiva

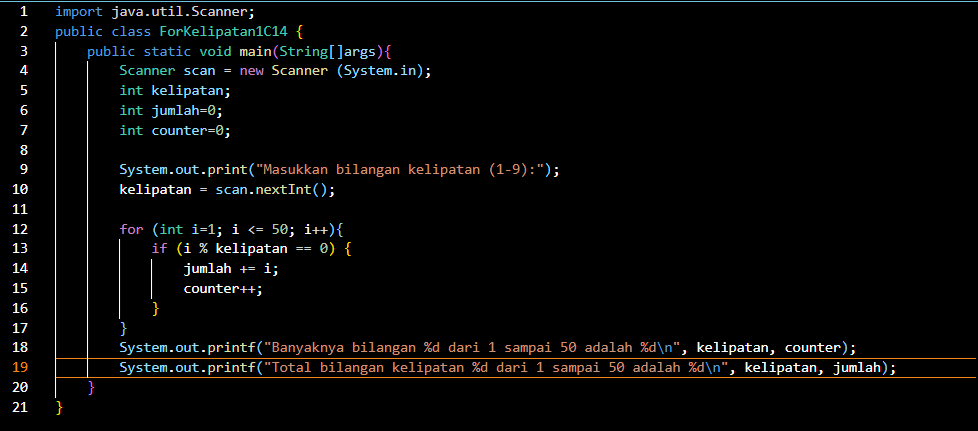
Kelas : SIB 1C

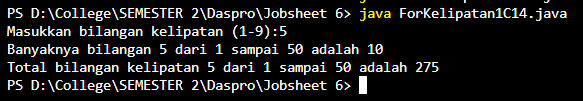
NIM : 2241760055

Mata Kuliah : Praktikum Dasar Pemrograman

**JOBSHEET 6**

**Percobaan 1 : Menghitung Bilangan Kelipatan Menggunakan FOR**





**Pertanyaan**

1. Terdapat tiga komponen perulangan pada sintaks FOR. Berdasarkan Percobaan 1 tersebut, sebutkan dan tunjukkan masing-masing komponen perulangan FOR pada kode program yang telah dibuat!

**for (int i = 1; i <= 50 ; i++)**

**int i = 1**

Merupakan inisialisasi, sebelum memulai perulangan kita harus menentukan dari mana perulangan tersebut akan dimulai. Pada sintaks diatas perulangan diberi nama **i** dengan tipe data integer dan diberi inisialisasi **1.** Yang artinya pengulangan dimulai dari angka 1

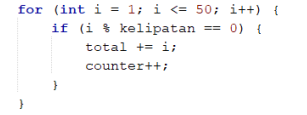
**i <= 50**

Merupakan syarat, perulangan For digunakan untuk Definite Loop yaitu perulangan yang telah diketahui batas akhirnya. Sehingga fungsi syarat tersebut adalah untuk membatasi sampai mana perulangan tersebut akan di jalankan. Pada sintaks diatas perulangan diakhiri dengan syarat **i** merupakan angka kurang dari sama dengan 50 (1-50). Selain angka tersebut maka sintaks tidak akan dieksekusi/dijalankan

**i++**

Merupakan update, dimana perintah ini digunakan pada fungsi yang dinilai TRUE maka angka angka yang dimasukkan (inputan) akan diupdate terus menerus (Looping) hingga batas syarat yang telah ditentukan. Sedangkan apabila telah mencapai batas maka akan dinilai FALSE karena tidak memenuhi syarat yang sudah di masukkan sebelumnya, sehingga program akan terhenti. Pada sintaks diatas perulangan yang dinilai benar akan di update dengan menambah +1 untuk tiap angka yang lolos(TRUE) pengecekan syarat.

2. Jelaskan alur kerja dari potongan kode program berikut!



**If ( i % kelipatan ==0) {**

**total +=1;**

**counter++;**

**}**

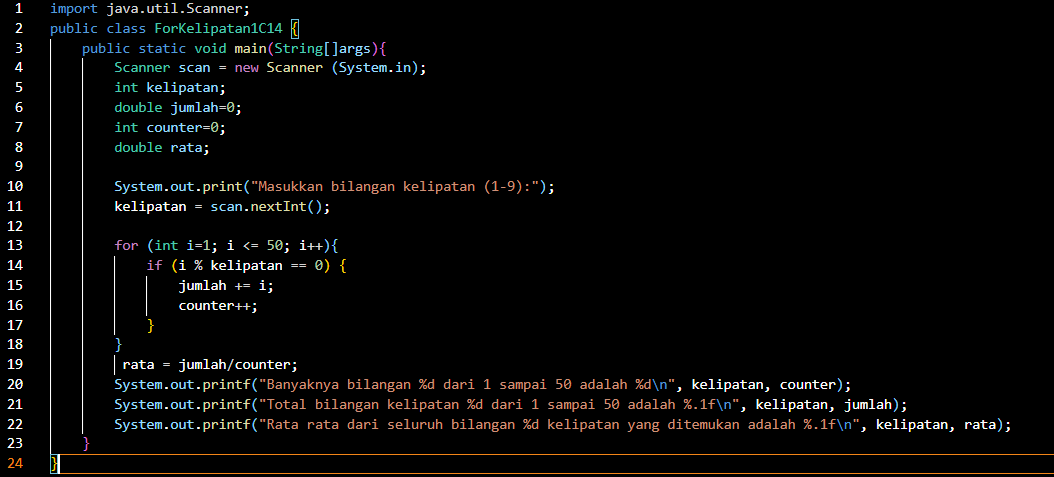
Sebelum memulai sintaks tersebut kita akan memasukan bilangan pada variabel **kelipatan** kemudian fungsi tersebut akan dieksekusi.

Jika **i** dalam sintaks diatas (Semua angka antara 1-50) di modulo oleh **kelipatan** yang kita masukkan sebelumnya akan menghasilkan 0 maka fungsi total dan counter akan dieksekusi.

**Total** = adalah jumlah total dari kelipatan yang ditemukan. Misalnya kita memasukkan 5 sebagai kelipatan maka akan ditemukan 5,10,15,20,25,30,35,40,45,50. Nah fungsi total akan menjumlahkan bilangan tersebut sehingga hasilnya 275.

**Counter** = adalah jumlah dari berapa kelipatan 5 dari 1 hingga 50. Misalnya kita memasukkan 5 sebagai kelipatan maka akan ditemukan 10 bilangan yang merupakan kelipatan dari 5 pada range 1-50.

3. Modifikasi kode program yang telah dibuat dengan menambahkan variabel baru untuk menghitung rata-rata dari seluruh bilangan kelipatan yang ditentukan!



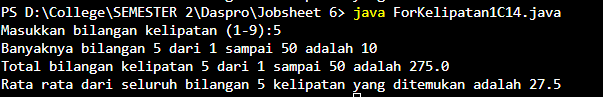
Saya memodifikasi dengan menambahkan variable bertipe **double** dengan nama **rata**. Kemudian merubah tipe data **jumlah** pada deklarasi menjadi **double**

Selanjutnya saya menambahkan perhitungan diluar sintaks **FOR** yaitu **rata= jumlah/counter.**

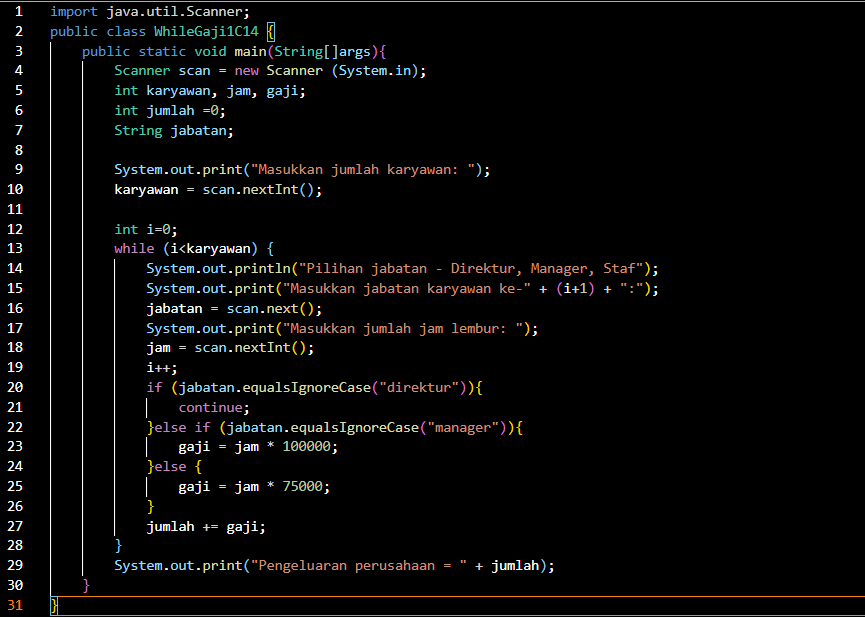
Selanjutnya saya merubah **%d\n** menjadi **%.1f\n** agar bisa memunculkan angka desimal dengan satu angka saja dibelakang koma.

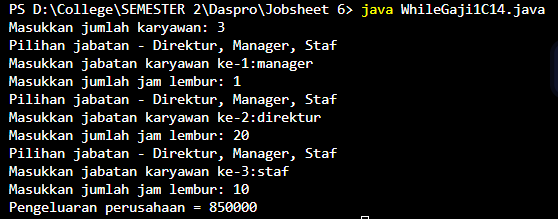
Menambahkan satu lagi sintaks **printf.** Untuk menampilkan rata rata yang telah terhitung.

Berikut adalah hasil output nya:



**Percobaan 2 : Menghitung Gaji Lembur Karyawan Menggunakan WHILE dan Continue**





1. Tunjukkan bagian kode program yang digunakan sebagai syarat untuk menghentikan perulangan WHILE! Berapa kali perulangan dilakukan?

Syarat ada pada sintaks **while (i<karyawan)** perulangan terjadi sebanyak kita memasukkan jumlah pada variabel **karyawan**. Dengan Syarat jumlah **karyawan** harus lebih besar dari **i**. Sedangkan **i** telah diinisialisasi dengan jumlah 0. Sehingga dapat disimpulkan bahwa input jumlah **karyawan** hanya bisa dijalankan apabila berjumlah positif (diatas 0)



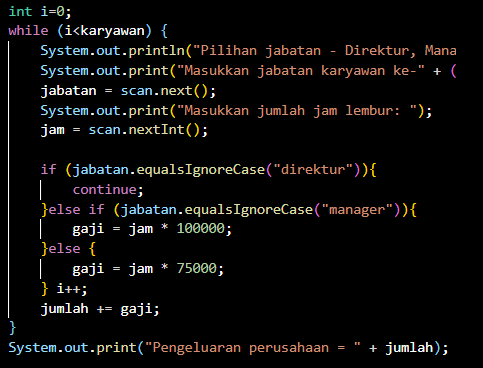
2. Pada potongan kode berikut,



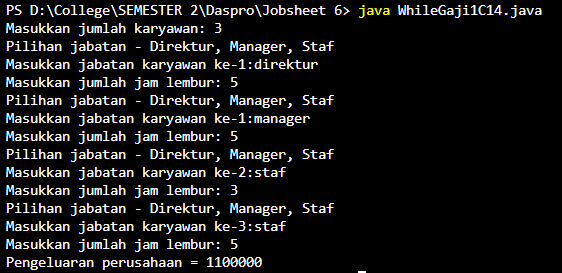
Apa yang sebenarnya terjadi jika variabel **jabatan** berisi nilai “DIREKTUR”? Apa peran CONTINUE yang dituliskan di dalam sintaks perulangan?

Pada saat jabatan **direktur** dimasukkan maka tidak ada perhitungan **gaji** yang di eksekusi. Penggunaan fungsi **Continue** adalah bertujuan untuk melanjutkan pada sintaks selanjutnya. Sehingga apabila **direktur** dimasukkan maka program akan langsung berlanjut mengeksekusi sintaks yang selanjutnya yaitu jabatan **manager.**

3. Mengapa komponen update **i++** diletakkan di posisi tengah, tidak di bagian akhir statement? Pindahkan **i++** di bagian akhir, lalu jalankan kembali program dengan memasukkan “direktur” sebagai jabatan karyawan pertama. Apa yang terjadi? Jelaskan!



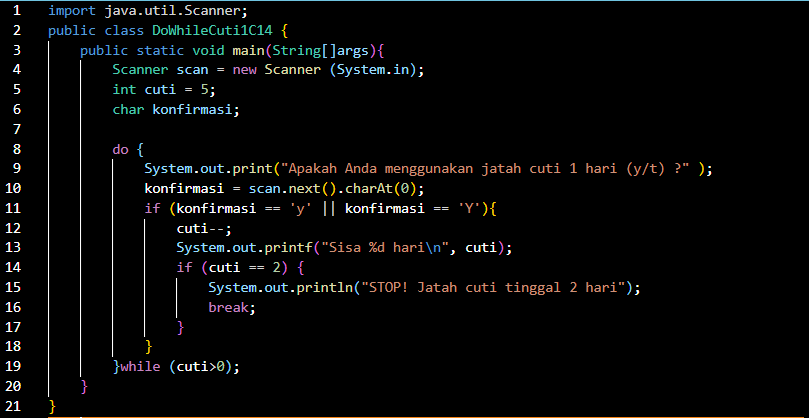
Saya mencoba memindahkan **i++** pada akhir sintaks **while**. Outputnya sebagai berikut

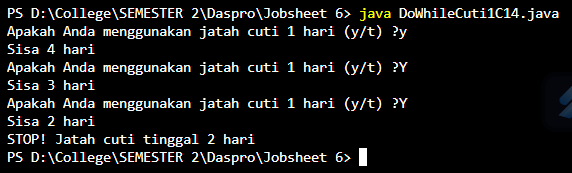


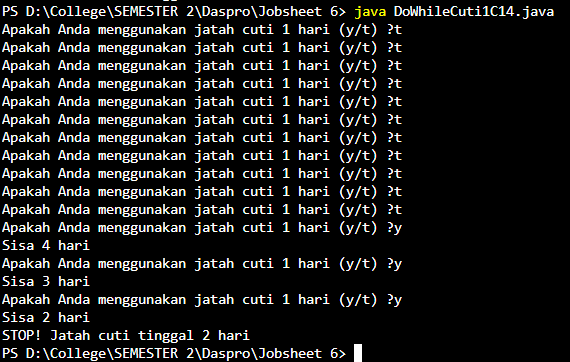
Perulangan terjadi sebanyak 4x dan ketika **jabatan direktur** dimasukkan maish tertulis (karyawan ke 1) dan setelah selesai, perulangan masih tertulis (karyawan ke 1). Baru ketika memasukkan **manager** dan **staf** maka penulisan karyawan ke- terupdate menjadi karyawan ke 2 dan seterusnya. Hingga terjadi 3x pengulangan.

Alasan diletakkan ditengah adalah agar pengulangan direktur tetap terhitung meski bertemu continue jumlah **i** tetap terupdate.

**Percobaan 3 : Menghitung Jatah Cuti Menggunakan DO-WHILE dan BREAK**





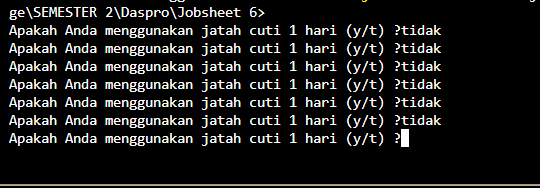


**Pertanyaan**

1. Apa kegunaan sintaks BREAK di dalam sintaks perulangan?

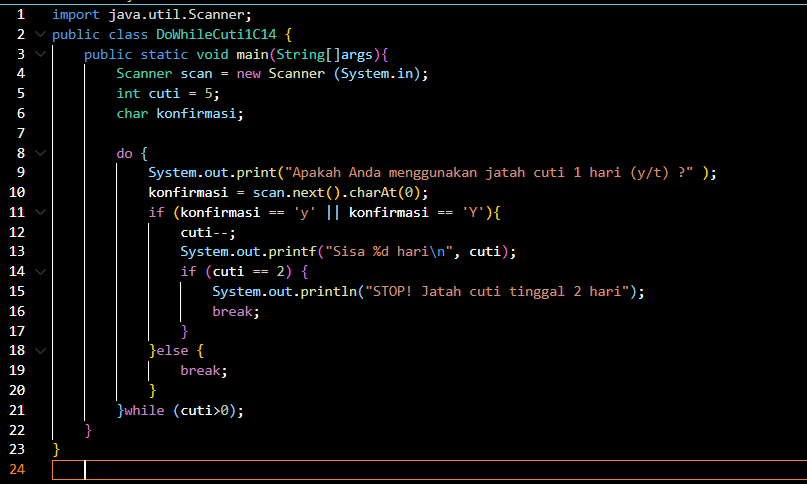
**Break** digunakan untuk menghentikan program dalam artian untuk keluar dari proses looping.

2. Jalankan kembali program tersebut. Pada saat input konfirmasi, ketikkan “Tidak”, apa yang terjadi? Mengapa demikian?



Perulangan terjadi tanpa batas dikarenakan kita tidak memasukkan fungsi **else** setelah **if** konfirmasi y. Sehingga apapun yang dimasukkan selain huruf (y/Y) akan terus melakukan perulangan.

3. Modifikasi kode program sehingga saat pengguna mengetikkan “Tidak” sebagai input konfirmasi, maka program akan berhenti!



Saya menambahkan fungsi **else** lalu di **break** agar program terhenti. Output sebagai berikut:

