**LAPORAN PRAKTIKUM 7 DASAR PEMROGRAMAN**



**Rangga Dwi Saputra 2341720248**

**Kelas 1B**

**Prodi D-IV Teknik Informatika**

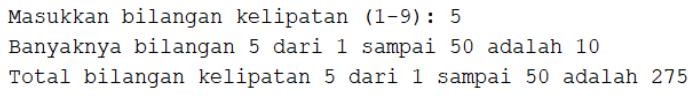
**Jobsheet 7**

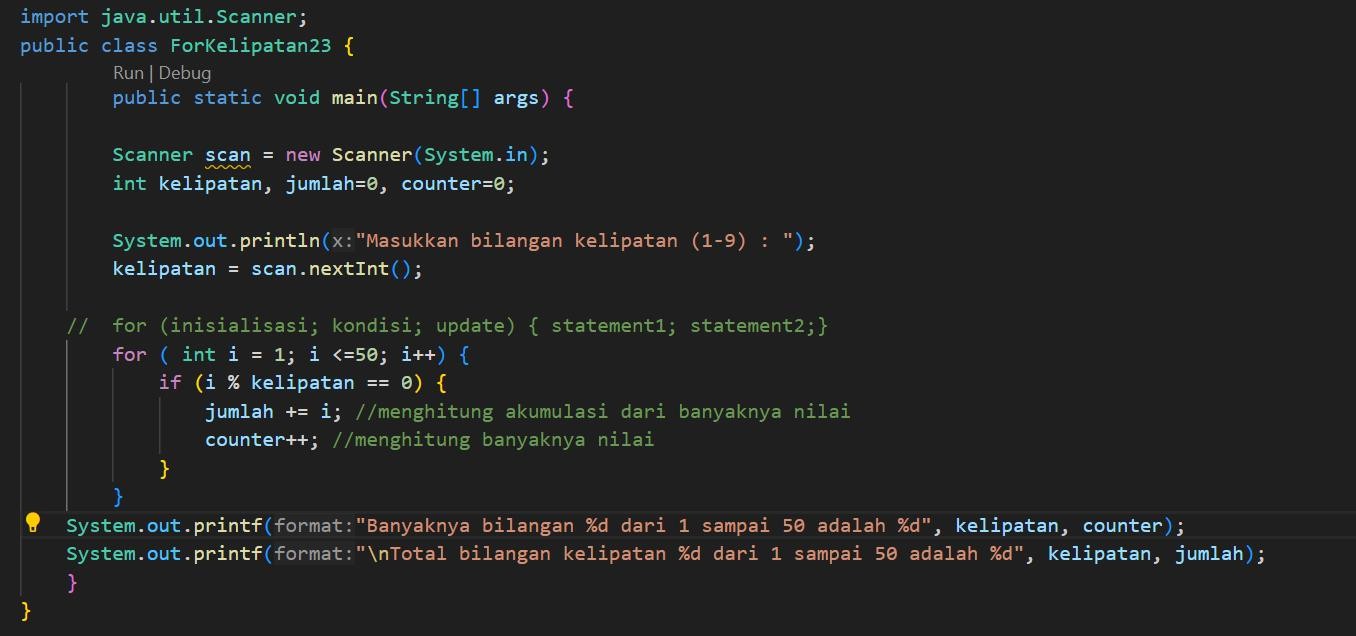
Perulangan 1

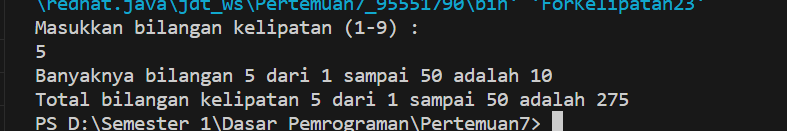
1. **Percobaan 1: Menghitung Bilangan Kelipatan Menggunakan FOR**

Pada percobaan ini dilakukan pembuatan kode program untuk menampilkan bilangan kelipatan angka tertentu dari rentang 1 sampai dengan 50 menggunakan perulangan FOR, serta menghitung total dari bilangan-bilangan tersebut.

* 1. Buat file baru di Visual Studio Code dengn nama file **ForKelipatanNoAbsen.java**
  2. Masukkan fungsi main() sebagai struktur dasar
  3. Tambahkan library Scanner
  4. Buat deklarasi Scanner dengan nama scan
  5. Buatlah variabel bertipe int dengan nama **kelipatan, jumlah, dan counter**. Inisialisasi variabel **jumlah=0** dan **counter=0**
  6. Tambahkan kode untuk menerima input dari keyboard
  7. Struktur perulangan FOR dengan kondisi pemilihan IF untuk menentukan bilangan kelipatan
  8. Tampilkan banyaknya bilangan kelipatan dan total bilangan kelipatan pada rentang 1 sampai dengan 50.
  9. Jalankan program tersebut. Cocokkan hasil compile kode program Anda dengan gambar berikut ini



* 1. Berdasarkan program didapatkan hasil sebagai berikut;

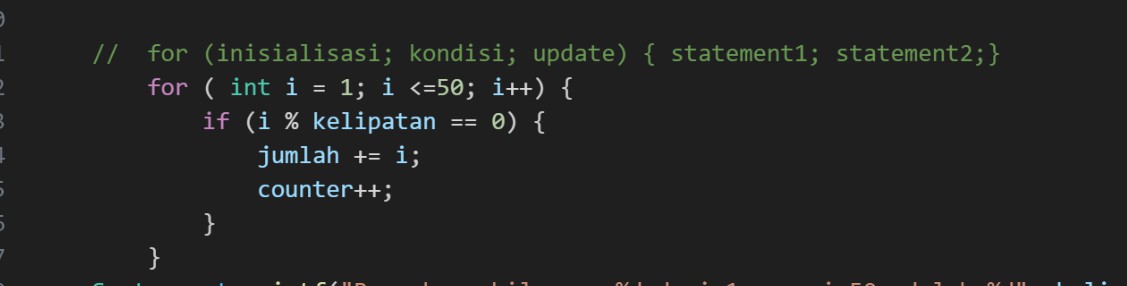
Output: sama dengan yang diminta

* 1. Commit dan push kode program ke github

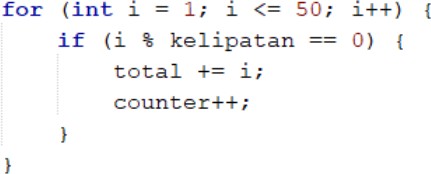


# Pertanyaan

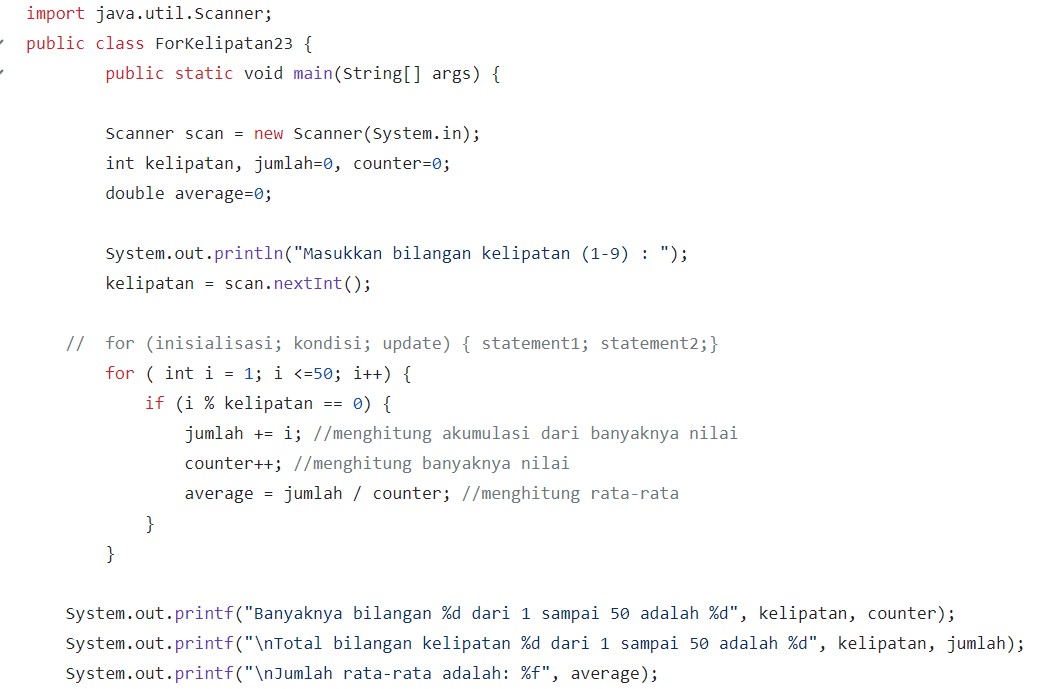
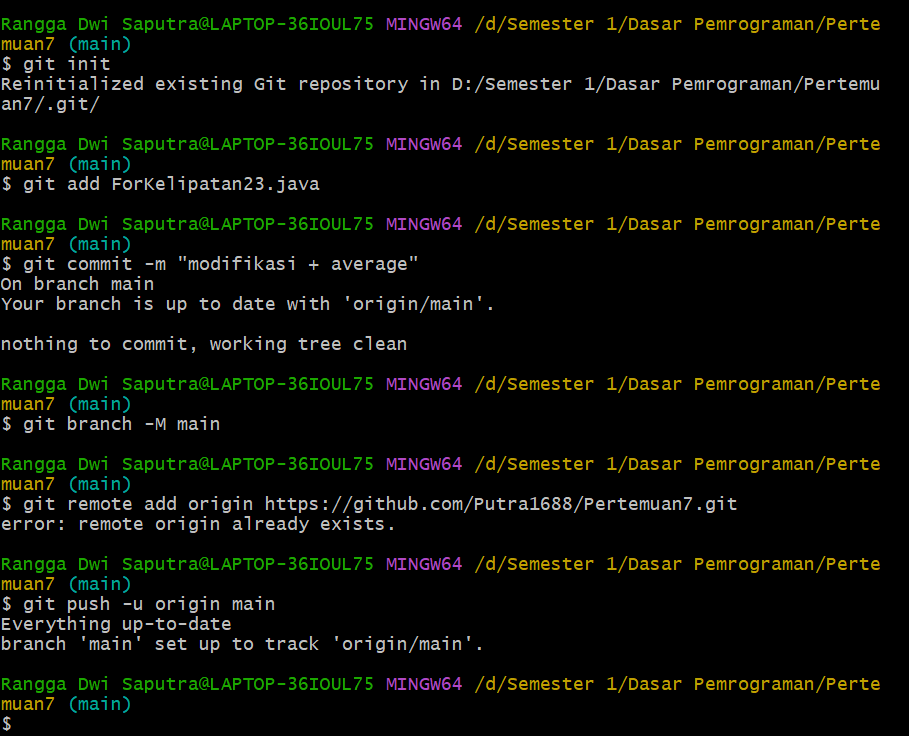
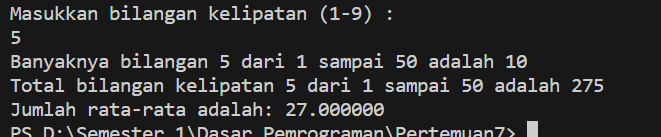
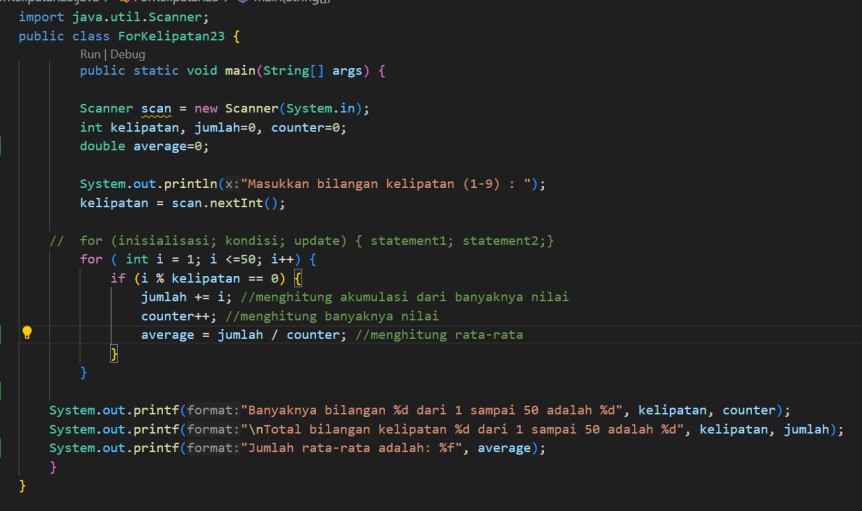
* + 1. Terdapat tiga komponen perulangan pada sintaks FOR. Berdasarkan Percobaan 1 tersebut, sebutkan dan tunjukkan masing-masing komponen perulangan FOR pada kode program yang telah dibuat!



* + 1. Jelaskan alur kerja dari potongan kode program berikut!



* + - 1. Perulangan diawali dengan melalukan inisialisasi **i=1**
      2. Tentukan kondisi **I <= 50**. Kemudian Evaluasi kondisi :
         * Kondisi **I** akan terus berlanjut selama kelipatannya tidak melebihi 50.
         * Counter++ akan menghitung banyaknya nilai dari kelipatan yang dihasilkan
         * Total += akan mengakumulasi nilai dari banyaknya kelipatan yang dihasilkan. akan melakukan update. Ulangi kembali langkah nomor 2
    1. Modifikasi kode program yang telah dibuat dengan menambahkan variabel baru untuk menghitung rata-rata dari seluruh bilangan kelipatan yang ditentukan! Push dan commit kode program ke github.



Output:

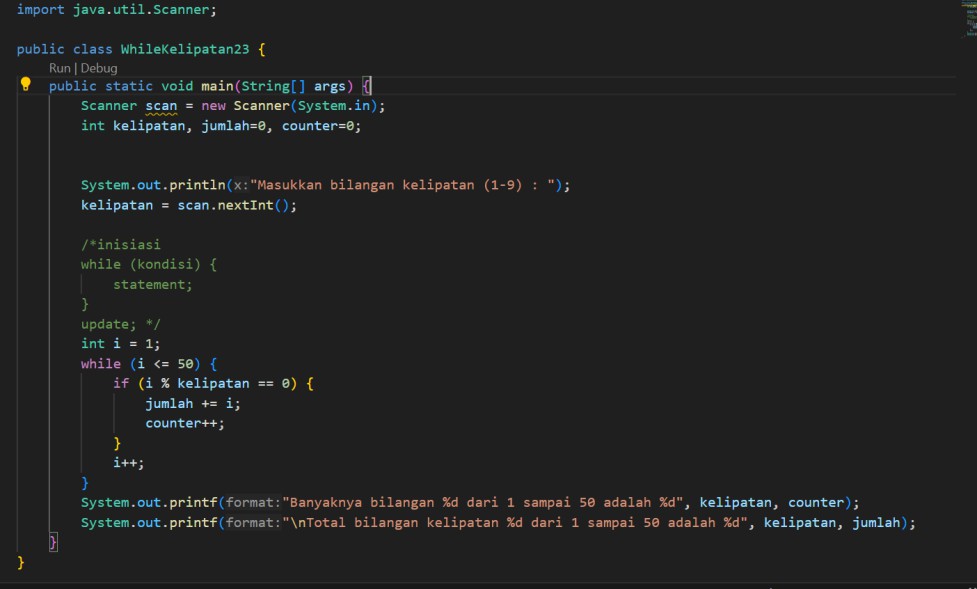
Push dan Commit ke Github:

average

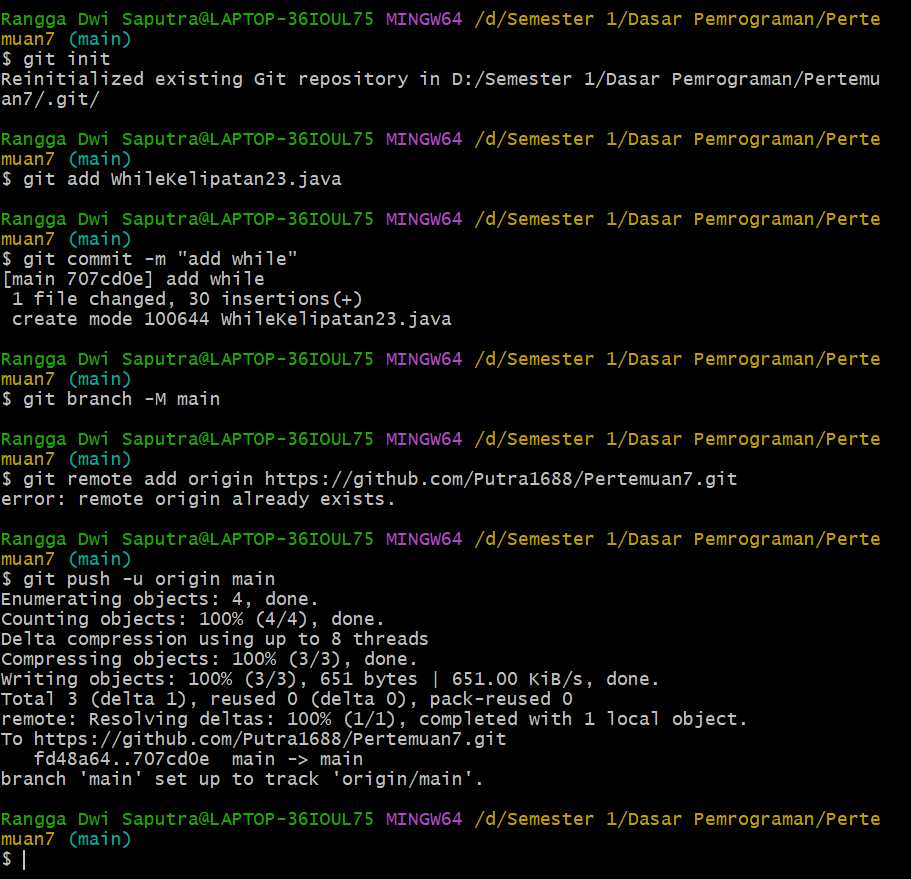
average

average

* + 1. Buatlah file baru dengan nama WhileKelipatanNoAbsen.java. Buatlah kode program dengan tujuan serupa tetapi menggunakan WHILE. Push dan commit kode program ke github.



Output



# Percobaan 2: Menghitung Gaji Lembur Karyawan Menggunakan WHILE dan CONTINUE

Sebuah perusahaan memberikan gaji lembur kepada karyawannya setiap minggu.

Gaji tersebut dihitung berdasarkan jabatan karyawan dan jumlah jam lembur dalam seminggu. Karyawan dengan jabatan “direktur” tidak mendapatkan tambahan gaji meskipun melakukan lembur, karyawan dengan jabatan “manager” mendapatkan gaji lembur sebesar 100000 per jam, sedangkan karyawan dengan jabatan “staf” mendapatkan gaji lembur sebesar 75000 per jam. Pada percobaan ini dilakukan pembuatan kode program menggunakan WHILE dan CONTINUE untuk menghitung pengeluaran perusahaan.

* 1. Buat file baru di Visual Studio Code dengn nama file

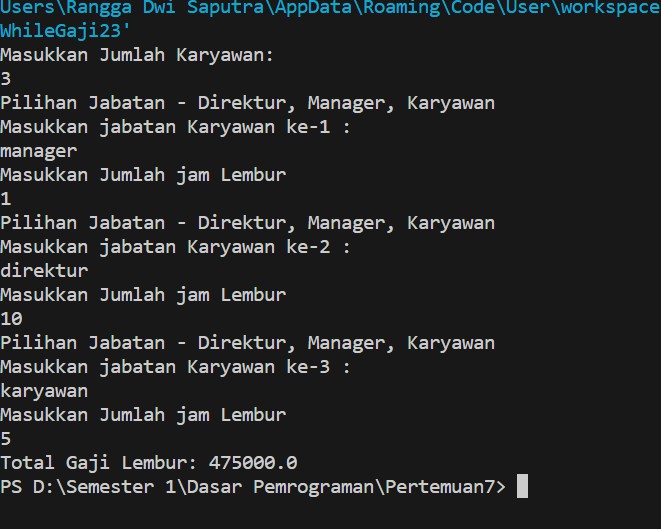
**ForKelipatanNoAbsen.java**

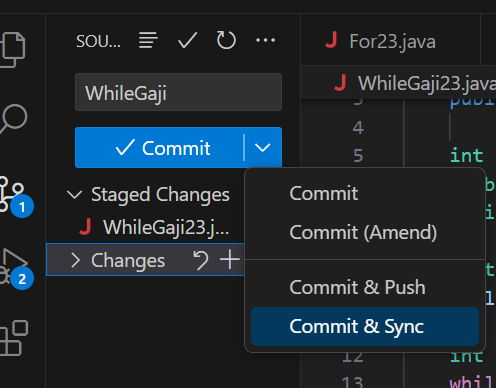
* 1. Masukkan fungsi main() sebagai struktur dasar
  2. Tambahkan library Scanner
  3. Buat deklarasi Scanner dengan nama scan
  4. variabel **jumlahKaryawan** dan **jumlahJamLembur** bertipe int serta

**gajiLembur** dan **totalGajiLembur** bertipe **double**.

* 1. Inisialisasi **variabel gajiLembur=0** dan **totalGajiLembur=0**
  2. Deklarasikan variabel **jabatan** bertipe **String**
  3. Tambahkan kode untuk menerima input dari keyboard guna menentukan jumlah karyawan yang akan dihitung gajinya
  4. Buat struktur perulangan **WHILE** dengan kondisi pemilihan **IF-ELSE** dan

**CONTINUE** untuk menentukan gaji lembur berdasarkan jabatan karyawan

* 1. Tampilkan hasil perhitungan jumlah gaji
  2. Berikut program dari output yang dihasilkan:
  3. Push dan Commit ke GitHub



# Pertanyaan

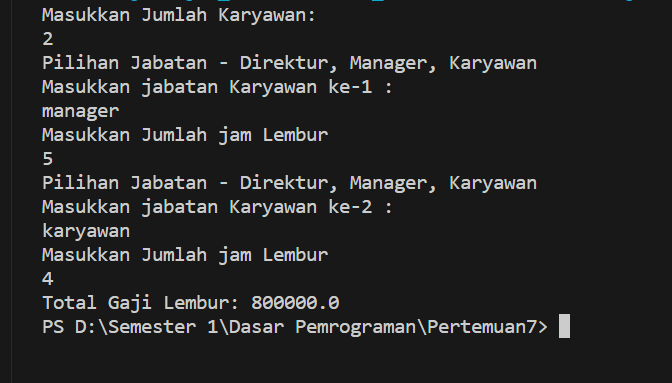
1. Tunjukkan bagian kode program yang digunakan sebagai syarat untuk menghentikan perulangan WHILE! Berapa kali perulangan dilakukan?



Perulangan

Syarat

**Syarat untuk menghentikan perulangan while ditentukan pada inputan berapa jumlah karyawan. Program akan tetap berajalan sampai pada jumlah yang sudah diinputkan**



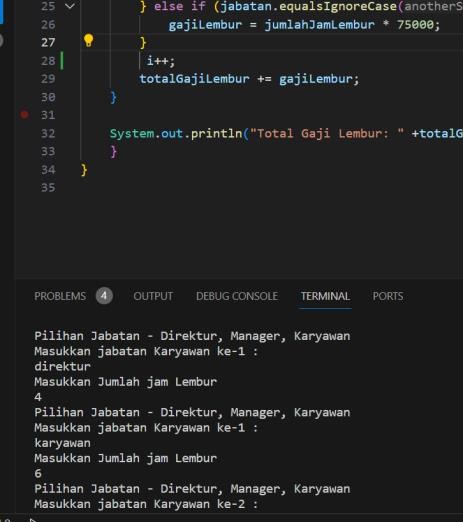
**Pada inputan berjumlah 2, maka program akan berulang sebanyak 2 kali saja. Setelah itu menjalankan program penutup.**

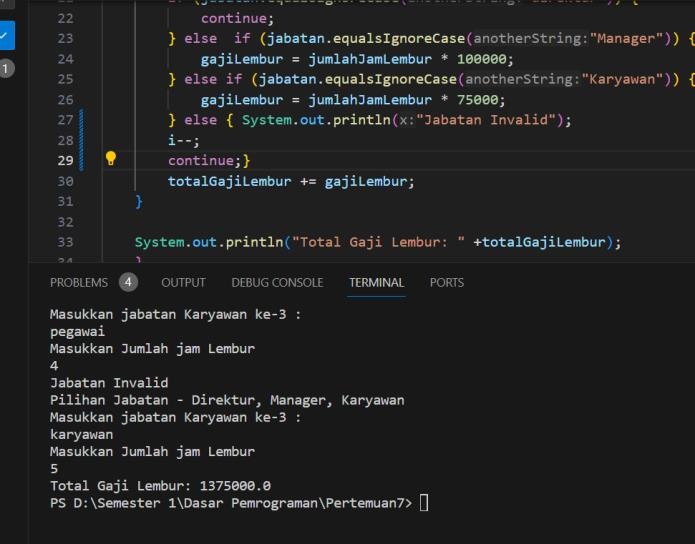
1. Pada potongan kode berikut, Apa yang sebenarnya terjadi jika variabel jabatan berisi nilai “DIREKTUR”? Apa peran CONTINUE yang dituliskan di dalam sintaks perulangan?

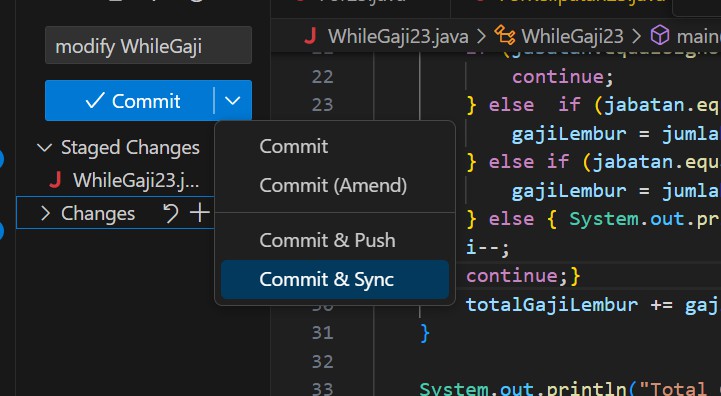
**Pada studi kasus disebutkan bahwa direktur tidak mendapatkan gaji tambahan, fungsi “continue” adalah melanjutkan program selnjutnya meskipun keluar dari ketentuan. Pada kondisi if memakai fungsi**

**equalsIgnoreCase, jadi pemeberian nama direktur/DIREKTUR akan sama saja karena penggunaan huruf kapital/tidak akan diabaikan**

1. 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!

**komponen i++ diletakkan ditengah untuk agar perhitungan dimulai ketika program mulai dijalankan, jika diletakkan di akhir maka perhitungan ke-1 akan dibaca 2 kali.**

1. Modifikasi kode program untuk menghandle jabatan yang invalid:

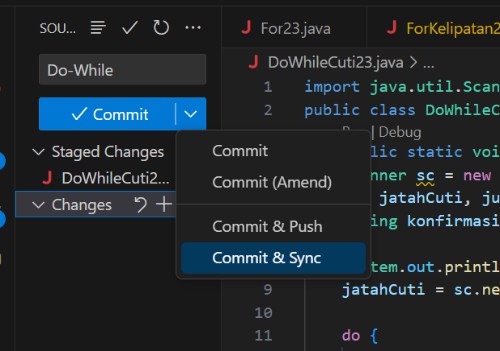
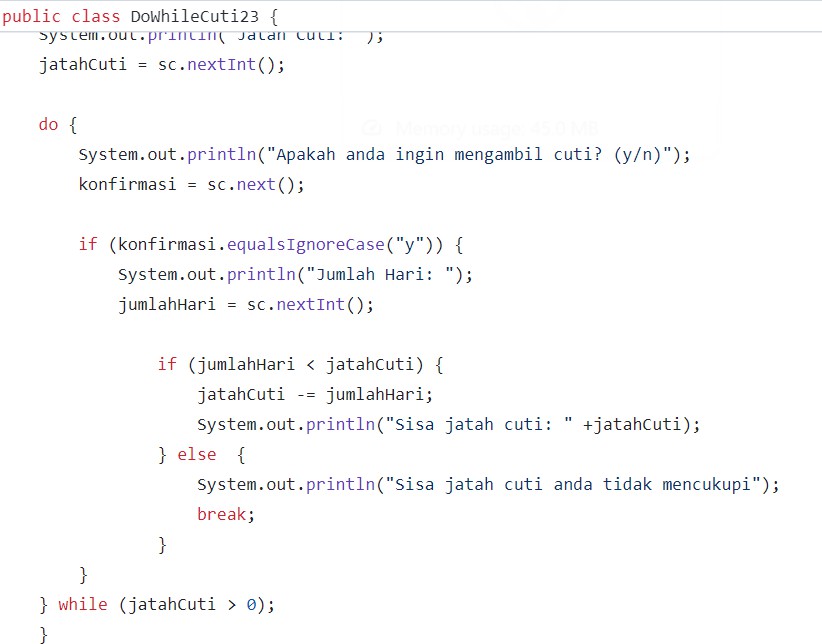
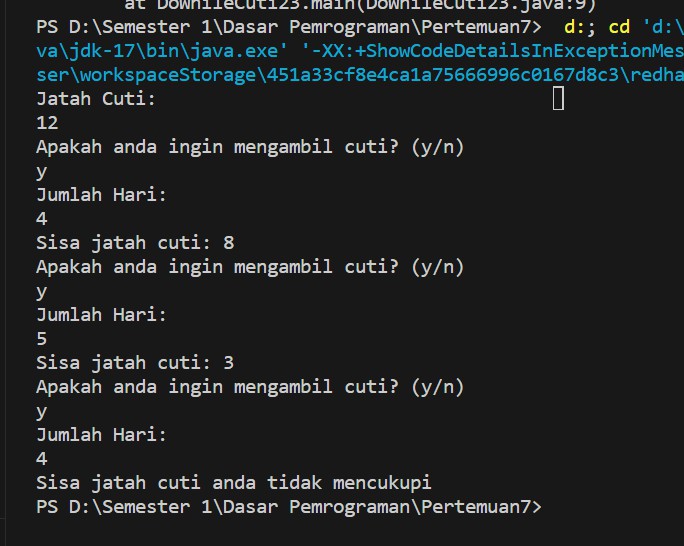


# Percobaan 3: Menghitung Jatah Cuti Menggunakan DO-WHILE

* 1. Buat file di Visual Studio Code dengan nama **DoWhile23.java**
  2. Buat struktur dasar fungsi main().
  3. Masukkan library Scanner
  4. Buat variable **jatahCuti** dan **jumlahHari** menggunakan tipe data int
  5. Buat variable **konfirmasi** menggunakan tipe data String
  6. Masukkan kode Scanner untuk menginput nilai jatahCuti
  7. Buat struktur data **Do-While** sebagai berikut



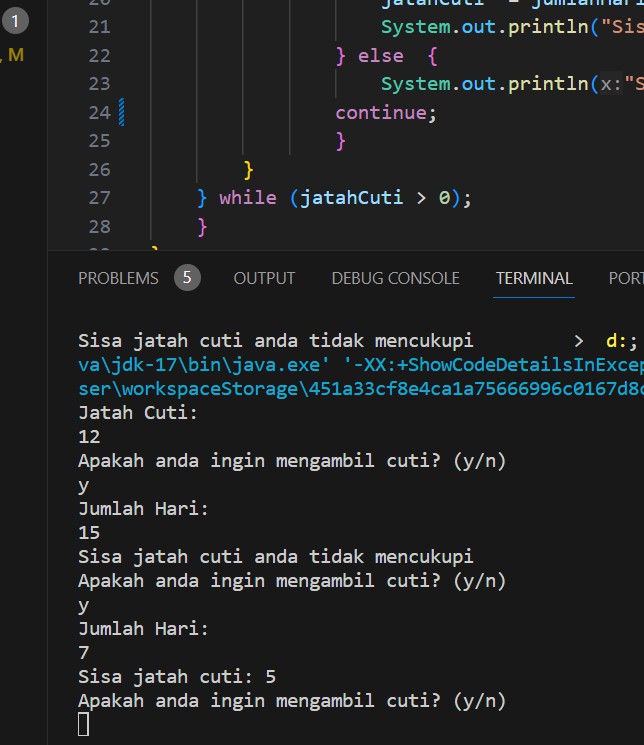
* 1. Akan menghasil output sebagai berikut:



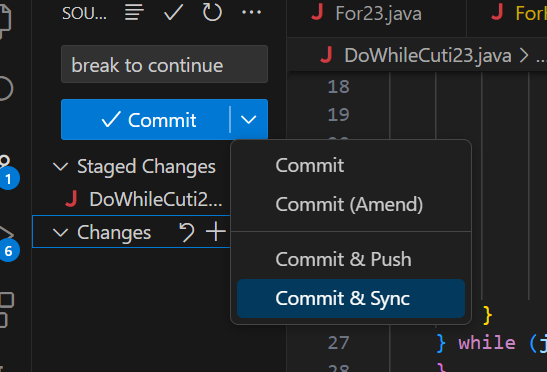
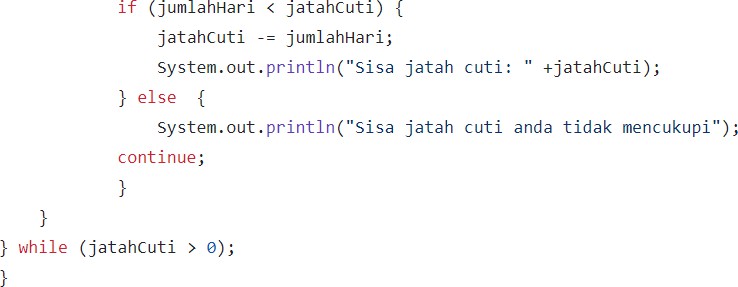
# Pertanyaan

* + 1. Apa kegunaan sintaks BREAK di dalam sintaks perulangan?

**Sintaks ‘break’ digunakan dalam perulangan dalam bahasa pemrograman untuk menghentikan perulangan secara paksa dan keluar dari perulangan saat kondisi tertentu terpenuhi**

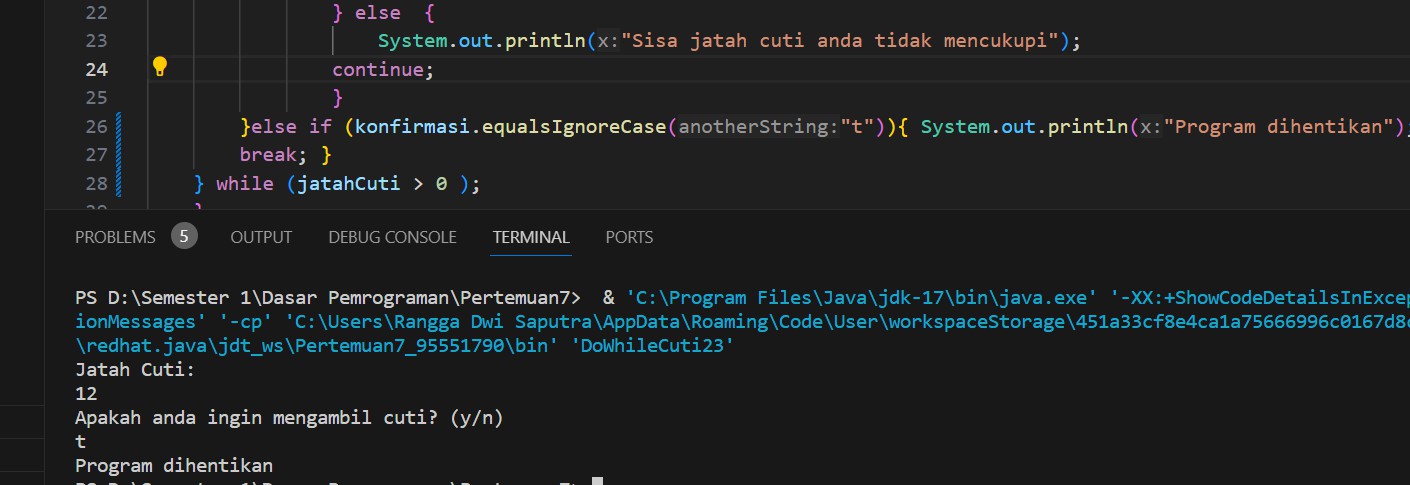
* + 1. Modifikasi kode program sehingga jika jumlah hari cuti yang ingin diambil lebih besar daripada jatah yang tersisa, program tidak berhenti sehingga pengguna masih memiliki kesempatan untuk mengisikan jumlah hari sesuai jatah cuti.
    2. Push dan commit kode program ke github

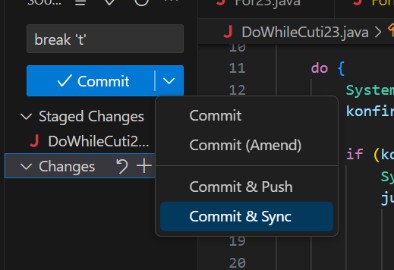
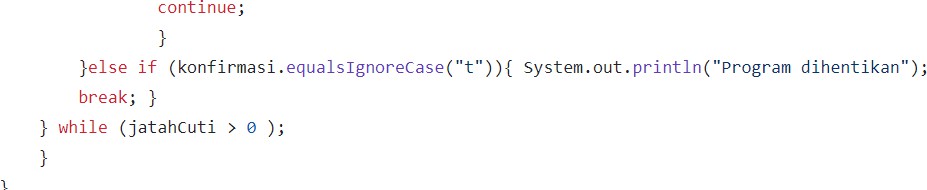
**kode ‘break’ diganti dengan ‘continue’.**



* + 1. Pada saat input konfirmasi, ketikkan “t”, apa yang terjadi? Mengapa demikian?

**Program akan terus mengulangi pertanyaan, karena program tidak ada kondisi yang menjalankan inputan “t”**

* + 1. Modifikasi kode program sehingga saat pengguna mengetikkan “t” sebagai input konfirmasi, maka program akan berhenti
    2. Push dan commit kode program ke github



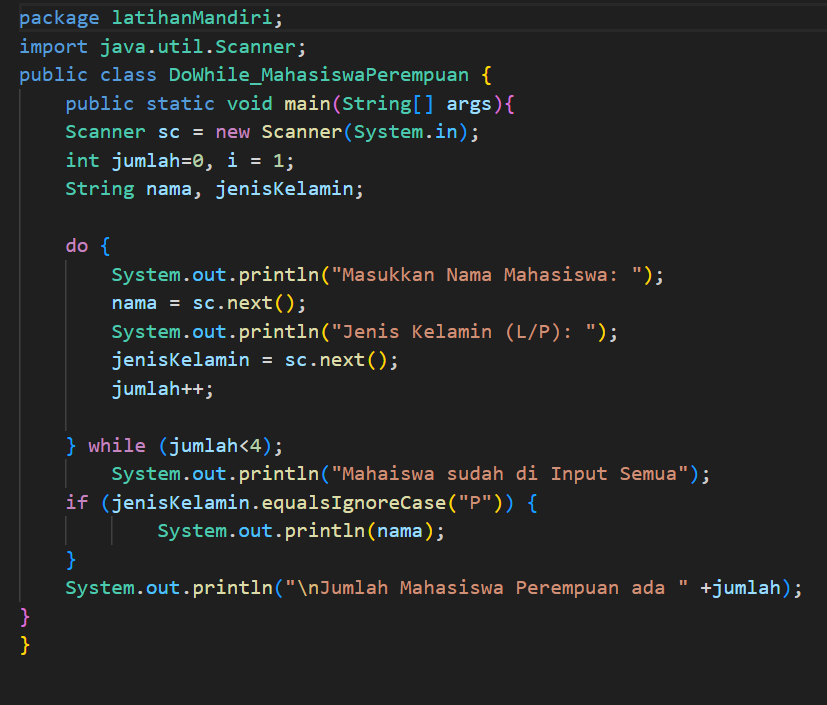
# TUGAS KELOMPOK

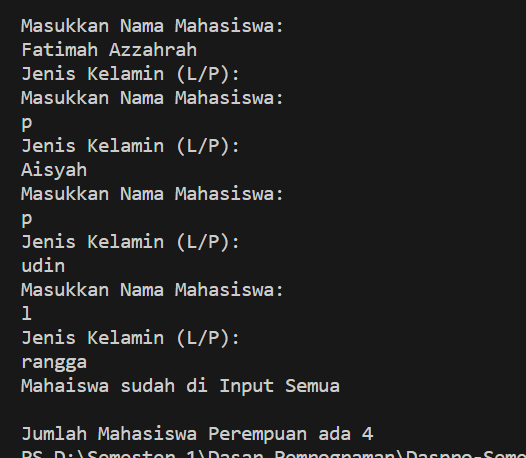
**PENERAPAN SISTEM PERULANGAN KEDALAM PROJECT**

1. **Fitur Login**
2. **Fitur Transaksi**

**LATIHAN MANDIRI**

1. **Menampilkan nama mahasiswa Perempuan menggunakan perulangan Do While**

****

****Harusnya sebagaimana di soal

kondisi yang ditentukan adalah

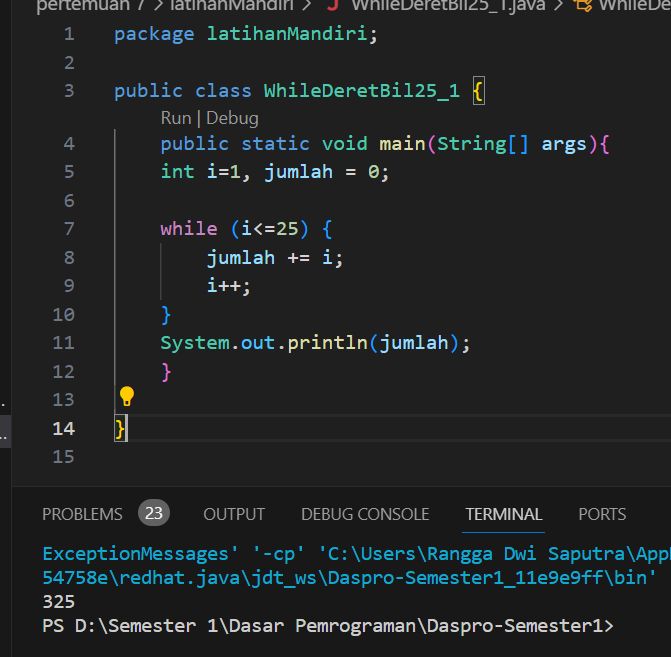
apabila jumlah <= 30. Namun,

demi mempermudah

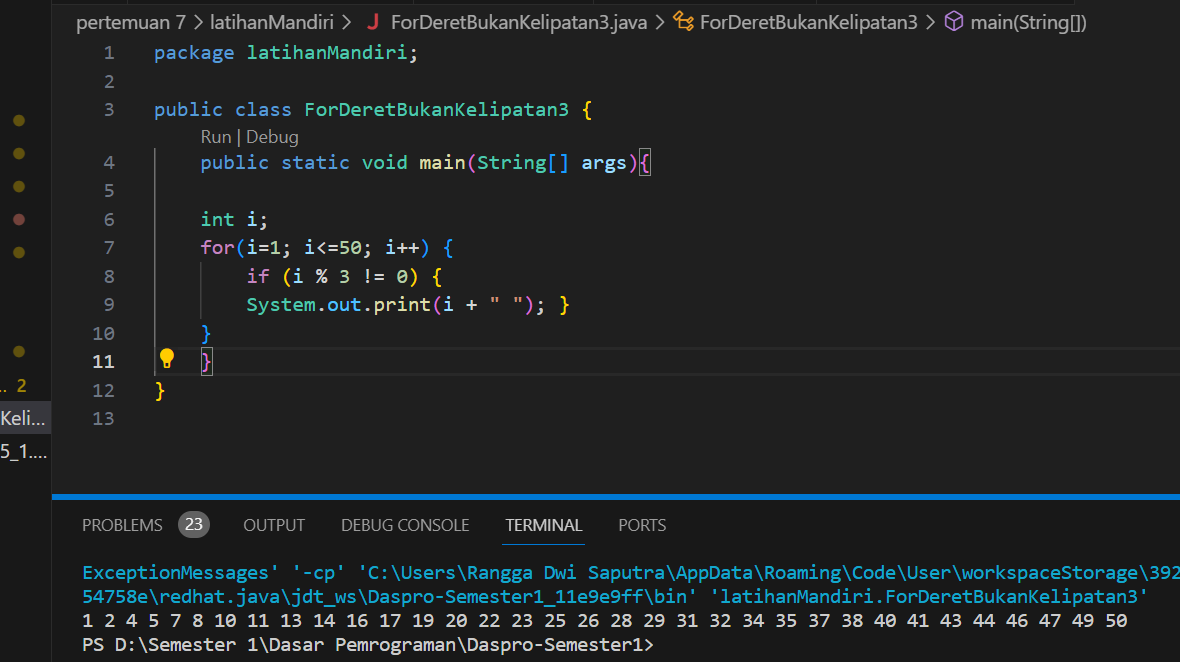
pengecekan maka diganti untuk

jumlah <4

1. **Menampilkan akumulasi jumlah deret dari bilangan 1-25 menggunakan perulangan while**

****

1. **Menampilkan deret 1 – 50 kecuali untuk bilangan berekelipatan 3 mengguankan perulangan for**

****