**LAPORAN PRAKTIKUM VII DASAR PEMROGRAMAN**



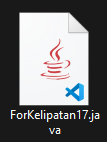
**1-B**

**Teknologi Informasi**

**Teknik Informatika**

**Percobaan 1 : ForKelipatan 17**

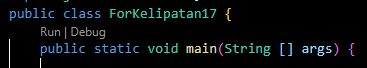
1. Pertama buat file dengan format ForKelipatan**NoAbsen**.java



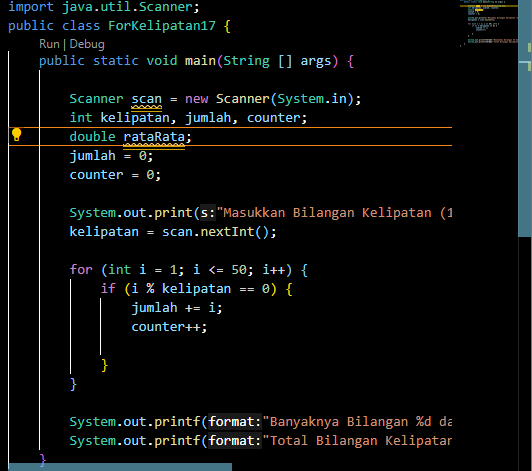
2. Kemudian masukkan perintah **import java.util.Scanner;** untuk memulai pemrograman.



3. Masukkan perintah **public static void main(String args[]) {** terlebih dahulu sebelum memulai pemrograman.



4. Kemudian mulailah pemrograman



5. Maka hasilnya akan seperti ini

****

**Pertanyaan 1 : ForKelipatan 17**

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. **‘int i = 1;’ adalah inisialisasi, yang menetapkan awal nilai variable iterasi menjadi 1.**
2. **i <= 50; adalah kondisi, yang menentukan apakah loop harus terus berlanjut atau berhenti. Selama nilai i kurang dari atau sama dengan 50, loop akan terus berlanjut.**
3. **i++; adalah iterasi, yang meningkatkan nilai variabel iterasi i sebesar 1 setelah setiap iterasi.**

2. Jelaskan alur kerja dari potongan kode program berikut!

**Potongan kode program tersebut merupakan bagian dari struktur kontrol perulangan for dalam bahasa pemrograman Java. Berikut adalah penjelasan alur kerjanya:**

1. **Loop dimulai dengan nilai awal 1.**
2. **Selama nilai ‘I’ kurang dari sama dengan 50,loop akan terus berlanjut.**
3. **Nilai I akan selalu bertambah setiap iterasi.**
4. **Dalam setiap iterasi, kode program akan memeriksa apakah i adalah kelipatan dari variabel kelipatan. Jika ya (i % kelipatan == 0), maka dua hal terjadi:**

* **Nilai i ditambahkan ke variabel total (total += i;).**
* **Variabel counter bertambah satu (counter++;).**

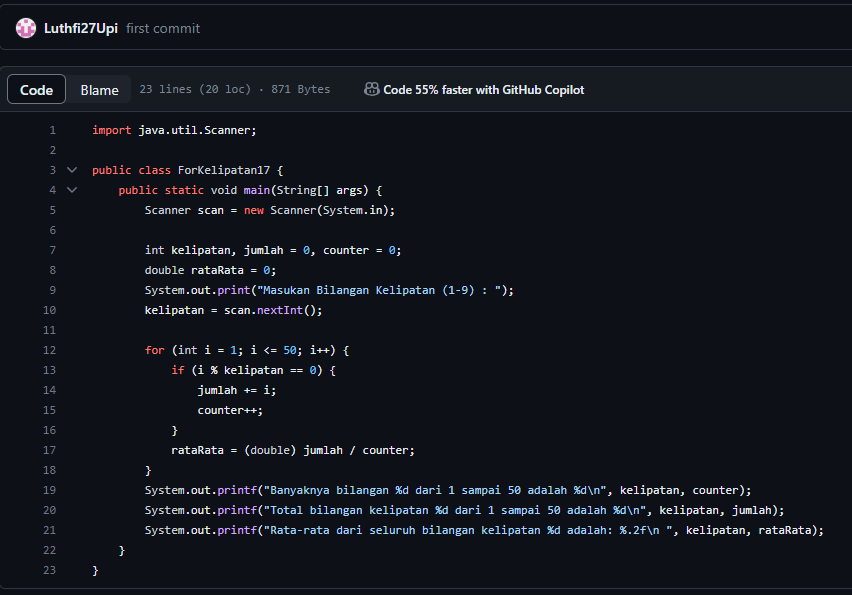
**Jadi, loop ini akan berjalan sebanyak 50 kali, dan dalam setiap iterasi, jika nilai i adalah kelipatan dari variabel kelipatan, maka nilai i akan ditambahkan ke total, dan counter akan bertambah satu, lalu variabel total akan berisi jumlah total dari semua bilangan yang merupakan kelipatan dari variabel kelipatan, dan variabel counter akan berisi banyaknya bilangan tersebut.**

3. 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.

Dibawah ini adalah kode program setelah diganti menjadi rata rata



Gambar dibawah ini adalah program setelah di push ke git

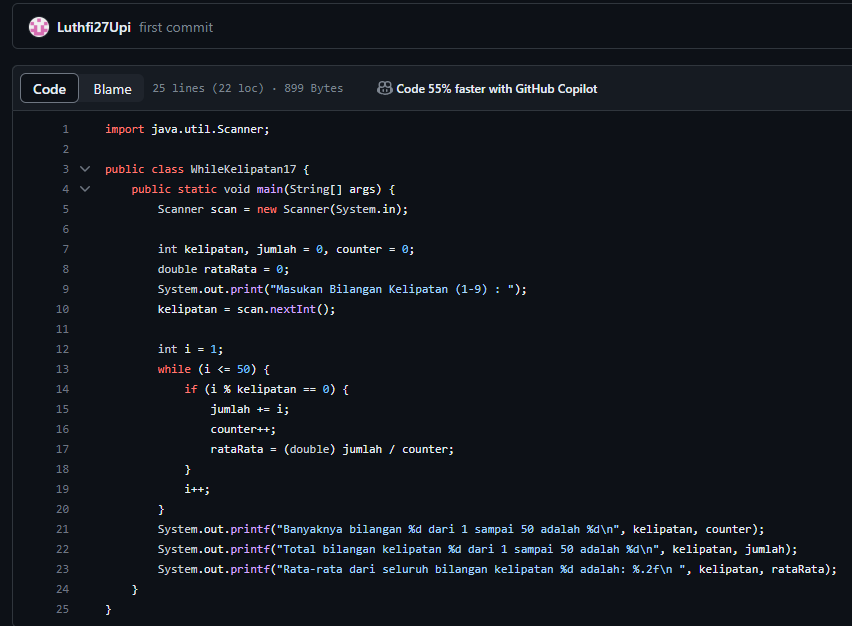


4. Buatlah file baru dengan nama WhileKelipatanNoAbsen.java. Buatlah kode program dengan tujuan serupa tetapi menggunakan WHILE. Push dan commit kode program ke github.

Dibawah ini adala kode program setelah diganti menjadi while

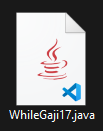


Gambar dibawah ini adalah kode program setelah di push ke git



**Percobaan 2 : WhileGaji 17**

1. Pertama buat file dengan format WhileGaji**NoAbsen**.java



2. Kemudian masukkan perintah **import java.util.Scanner;** untuk memulai pemrograman.



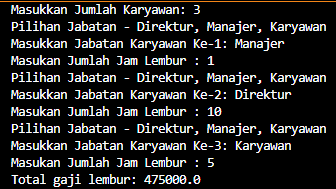
3. Masukkan perintah **public static void main(String args[]) {** terlebih dahulu sebelum memulai pemrograman.



4. Kemudian mulailah pemrograman



5. Maka hasilnya akan seperti ini

****

**Pertanyaan 2 : WhileGaji 17**

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

**Kode untuk menghentikan perulangan adalah i < jumlahKaryawan. Ini berarti bahwa selama nilai i kurang dari jumlahKaryawan, perulangan akan terus berlanjut. Jadi jika input jumlahKaryawan 5, maka perulangan akan dilakukan sampai 5 kali.**

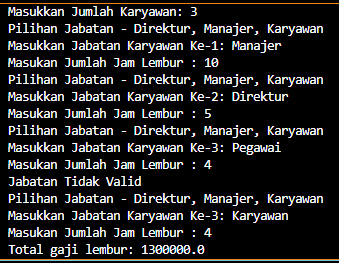
2. Pada potongan kode berikut, Apa yang sebenarnya terjadi jika variabel jabatan berisi nilai “DIREKTUR”? Apa peran CONTINUE yang dituliskan di dalam sintaks perulangan?

**Fungsi continue dalam program ini adalah untuk melewati penghitungan gaji lembur untuk karyawan dengan jabatan “Direktur”.**

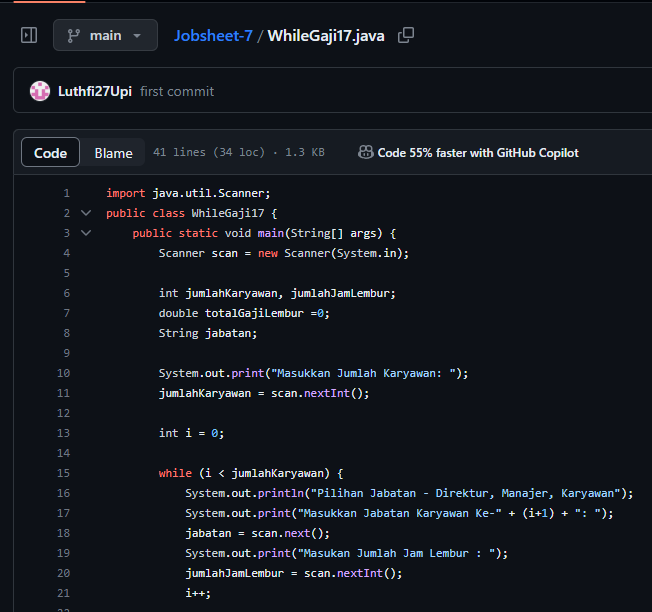
3. Mengapa komponen update i++ diletakkan di posisi tengah, tidak di bagian akhir statement? Pindahkan i++ di bagian akhir, lalu jalankan kembali program dengan memasukkan “direktur” sebagai jabatan karyawan pertama. Apa yang terjadi? Jelaskan!

**Memindahkan i++ ke bagian akhir pernyataan akan menyebabkan loop tak terbatas jika jabatan karyawan pertama adalah “Direktur”.**

4. Modifikasi kode program untuk menghandle jabatan yang invalid

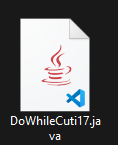


5. Commit dan Push ke git!



**Percobaan 3 : DoWhileCuti17**

1. Pertama buat file dengan format DoWhileCuti**NoAbsen**.java



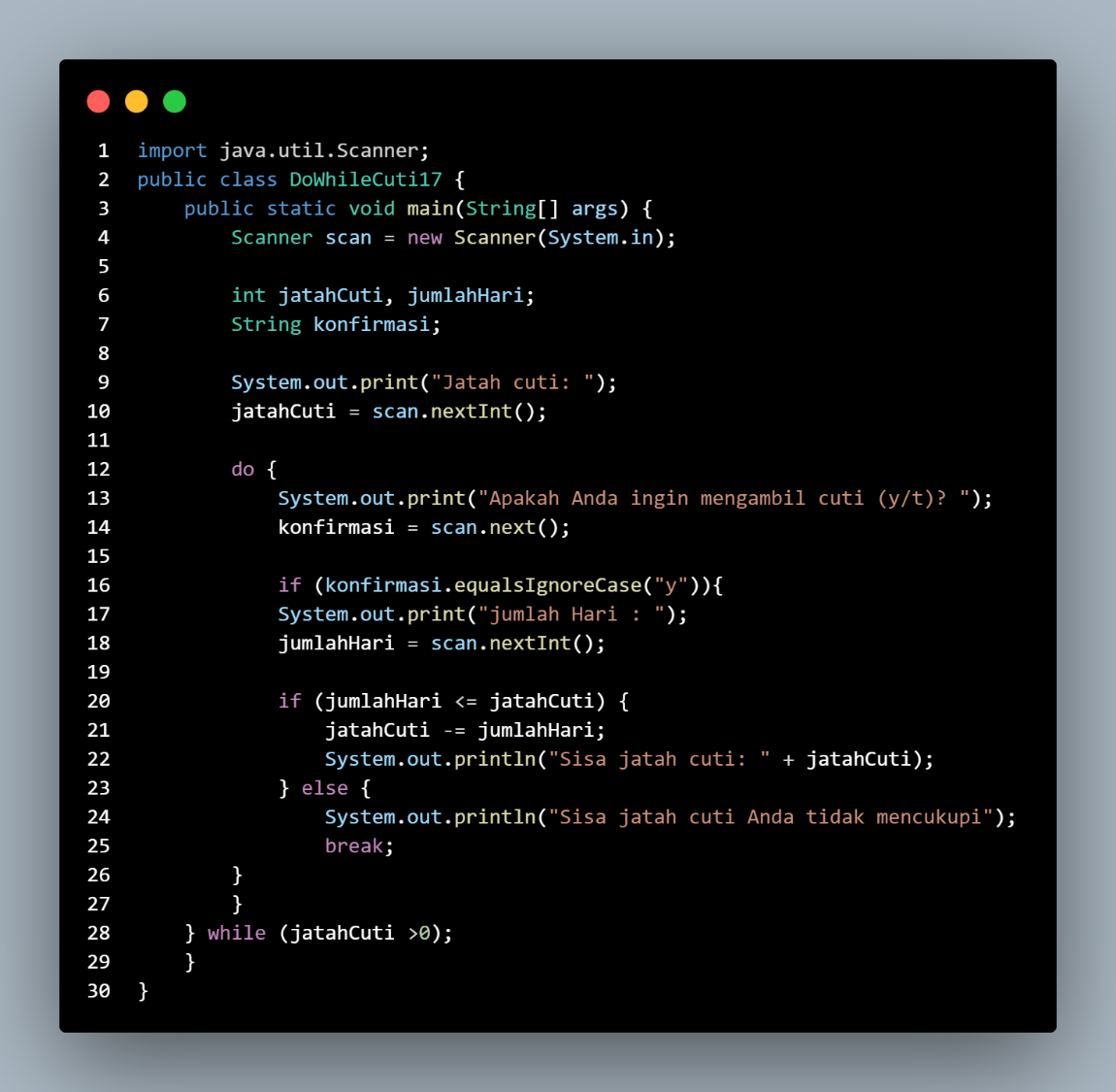
2. Kemudian masukkan perintah **import java.util.Scanner;** untuk memulai pemrograman.



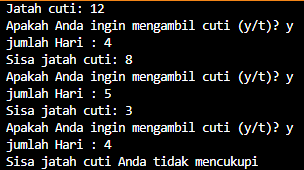
3. Masukkan perintah **public static void main(String args[]) {** terlebih dahulu sebelum memulai pemrograman.



4. Kemudian mulailah pemrograman



5. Maka hasilnya akan seperti ini

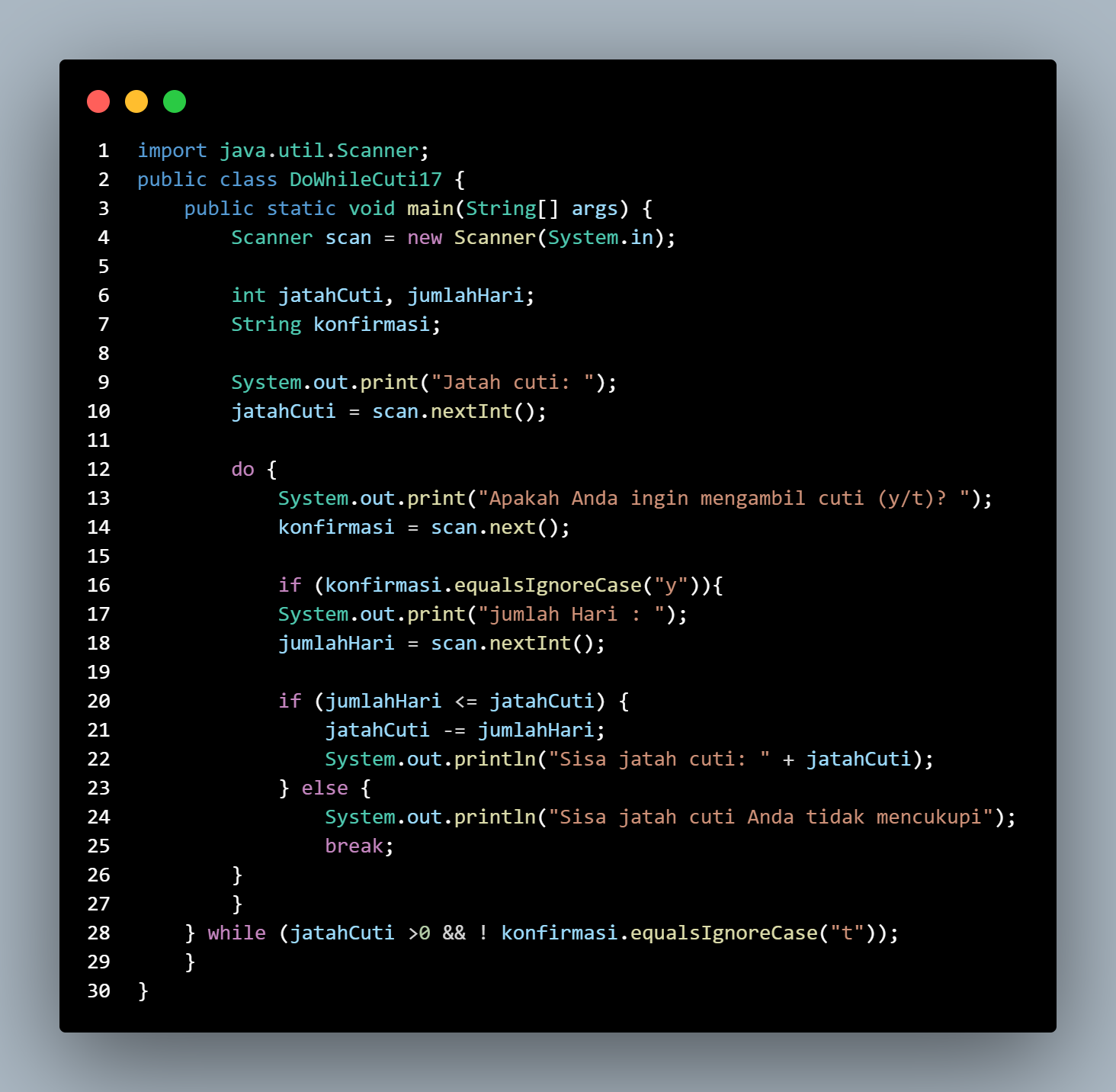
****

**Pertanyaan 3 : DoWhileCuti 17**

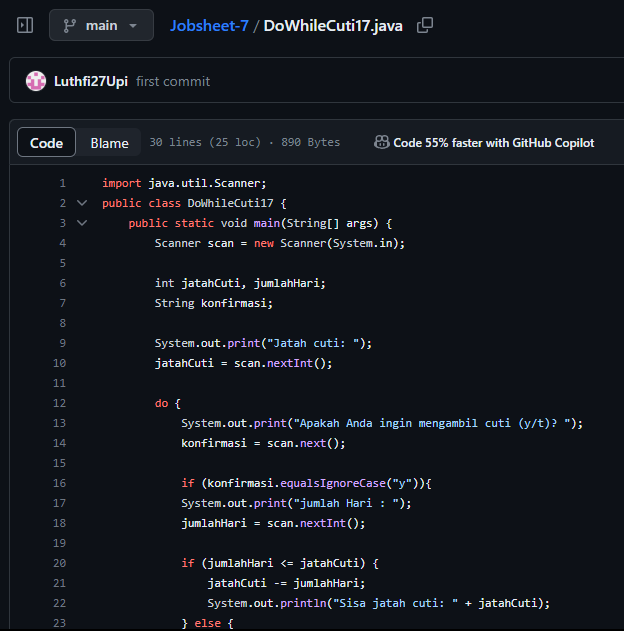
1. Apa kegunaan sintaks BREAK di dalam sintaks perulangan?

**Sintaks BREAK berguna untuk menghentikan eksekusi dari perulangan. Selanjutnya program akan mengeksekusi kode setelah blok perulangan.**

2. 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.



3. Push dan commit kode program ke github



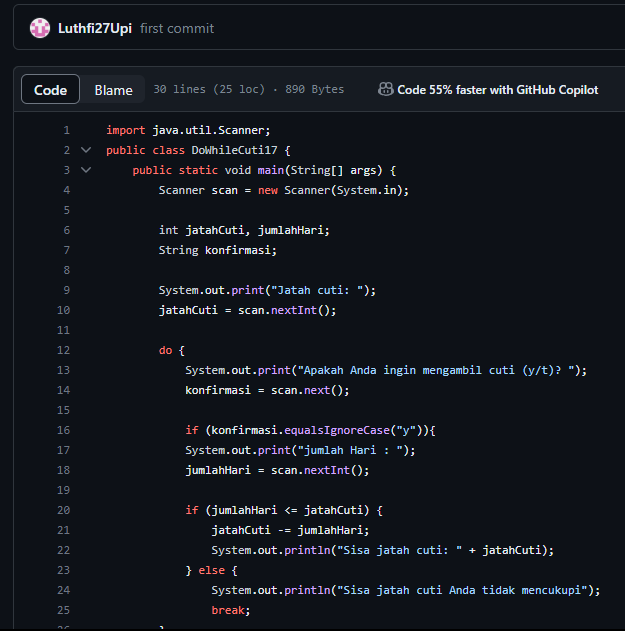
4. Pada saat input konfirmasi, ketikkan “t”, apa yang terjadi? Mengapa demikian?

**Program akan terus melakukan pengulangan pada input, karena kode program hanya melakukan pemeriksaan pada input equalsIgnoreCase= y. Dan tidak melakukan pengecekan pada input= t**

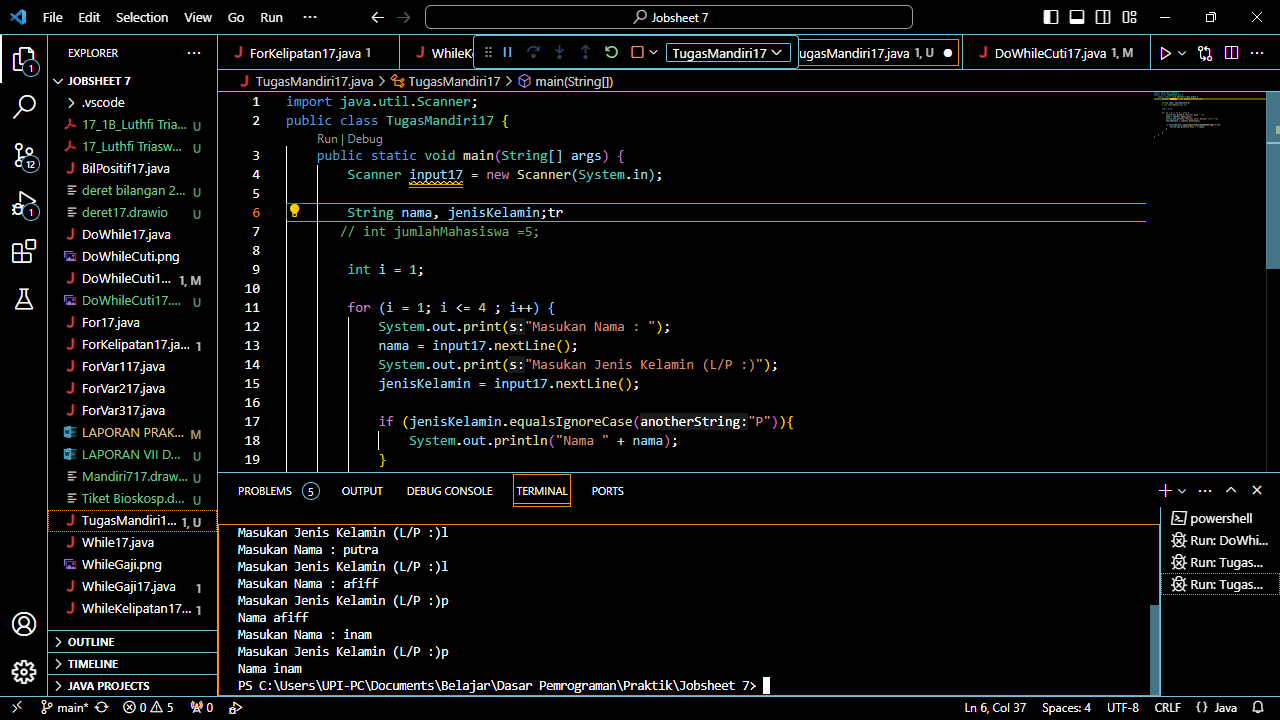
5. Modifikasi kode program sehingga saat pengguna mengetikkan “t” sebagai input konfirmasi, maka program akan berhenti



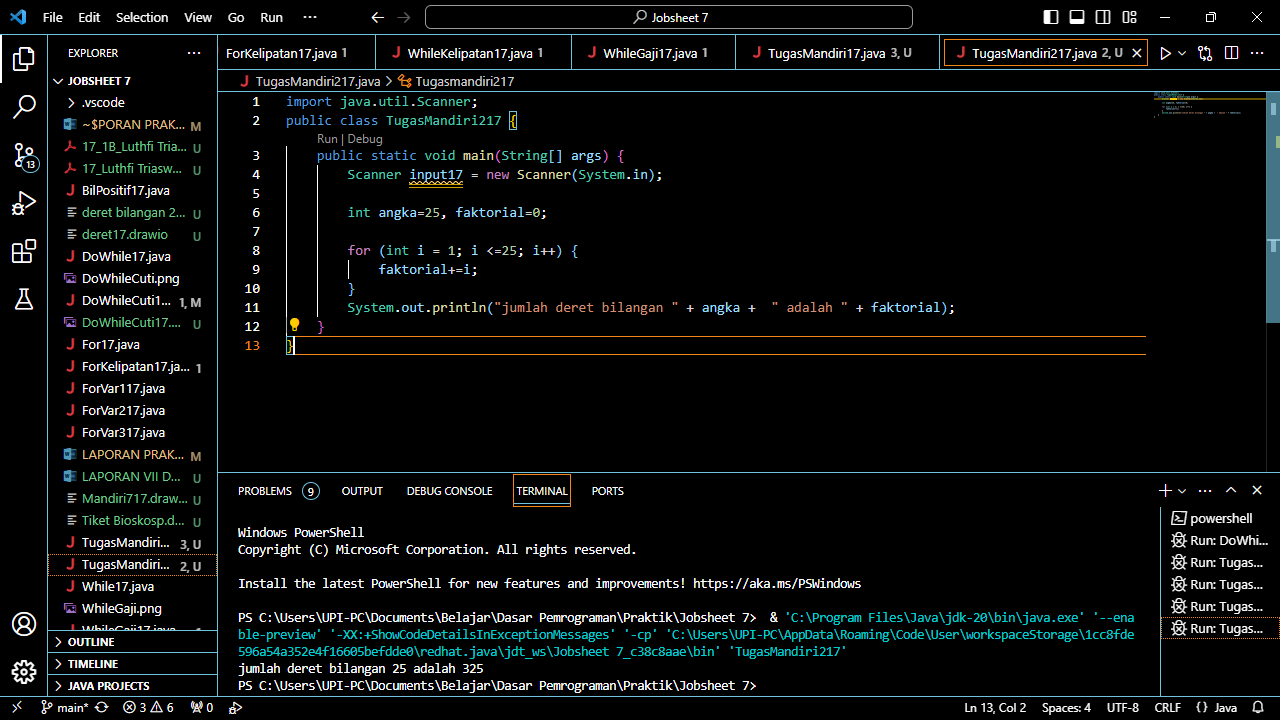
6. Push dan commit kode program ke github

****

**Tugas Mandiri**

**A. Jumlah Mahasiswa**

**B. Deret 25**



**C. Deret 1-50**



**D. Kelompok Login**



**E. Kelompok Pesan Kursi**

