**LAPORAN PRAKTIKUM**

**DASAR PEMROGRAMAN**

**JOBSHEET 7**

**Perulangan 1**

|  |  |
| --- | --- |
| **HIKMAH ALDRIN ABDILLAH** | **NIM: 2341720049** |

****

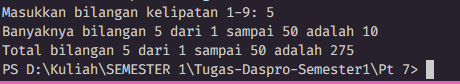
**PROGRAM STUDI D-IV TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

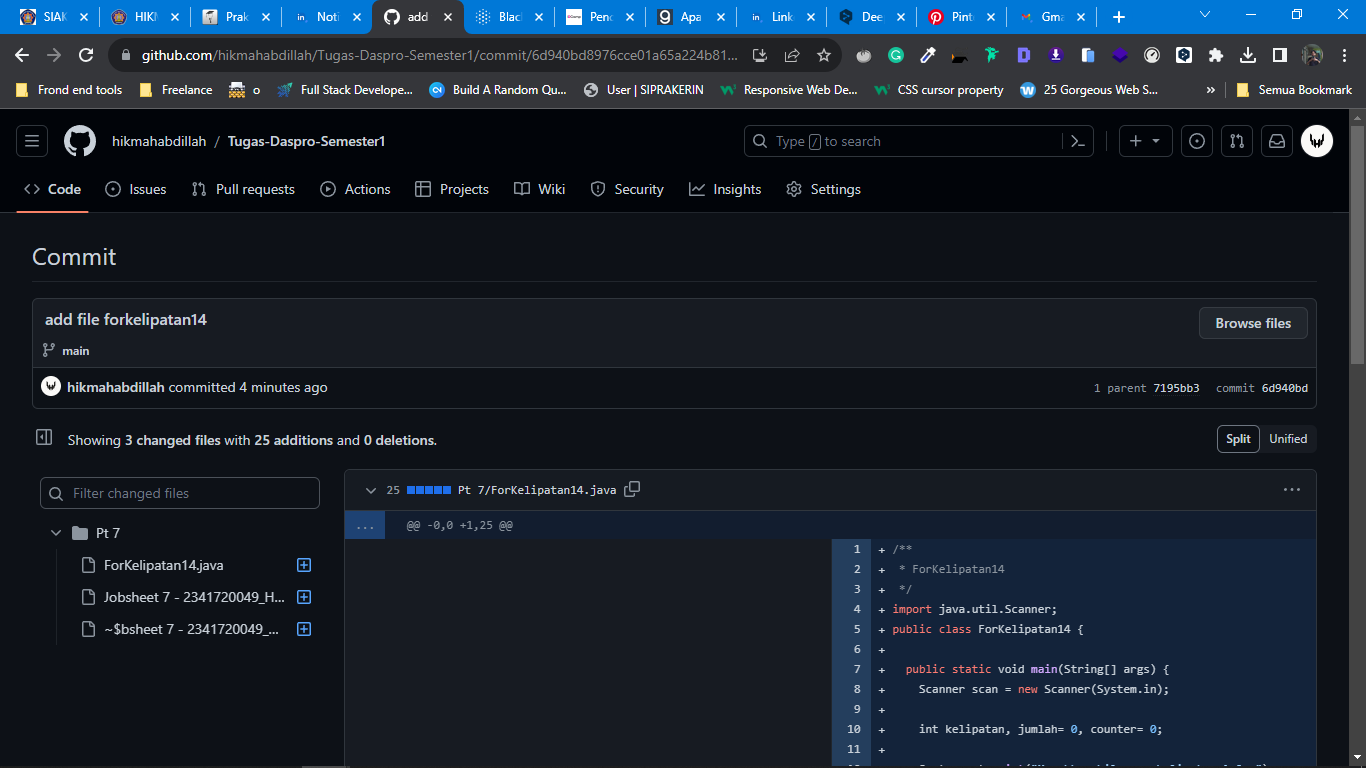
**POLITEKNIK NEGERI MALANG**

**OKTOBER 2023**

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

Hasil 

Bukti commit



**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!

**Jawab :**

* Int I = 1 merupakan deklarasi untuk variable i.
* I <= 50 merupakan conditional/pengkondisian yaitu ketika perulangan bernilai benar, maka akan melakukan statement yang sama sampai akhirnya bernilai false.(batas perulangan)
* I++ merupakan increment(update)

2. Jelaskan alur kerja dari potongan kode program berikut!

**Jawab:**

kode tersebut melakukan perulangan dari 1 hingga 50, memeriksa apakah setiap nilai **i** adalah kelipatan dari **kelipatan**, dan jika iya, maka nilai **i** ditambahkan ke **jumlah**, dan **counter** ditingkatkan. Setelah perulangan selesai, total jumlah dari semua kelipatan yang memenuhi kondisi, dan juga jumlah kelipatan yang telah ditemukan.

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.

**Jawab:**

/\*\*

 \* ForKelipatan14

 \*/

import java.util.Scanner;

public class ForKelipatan14 {

  public static void main(String[] args) {

    Scanner scan = new Scanner(System.in);

    int kelipatan, average=0, jumlah= 0, counter= 0;

    System.out.print("Masukkan bilangan kelipatan 1-9: ");

    kelipatan = scan.nextInt();

    for (int i = 1; i <= 50; i++) {

      if(i % kelipatan == 0){

        jumlah += i;

        counter++;

      }

    }

    // operasi untuk menghitung rata rata

    average = (kelipatan \* jumlah) / counter;

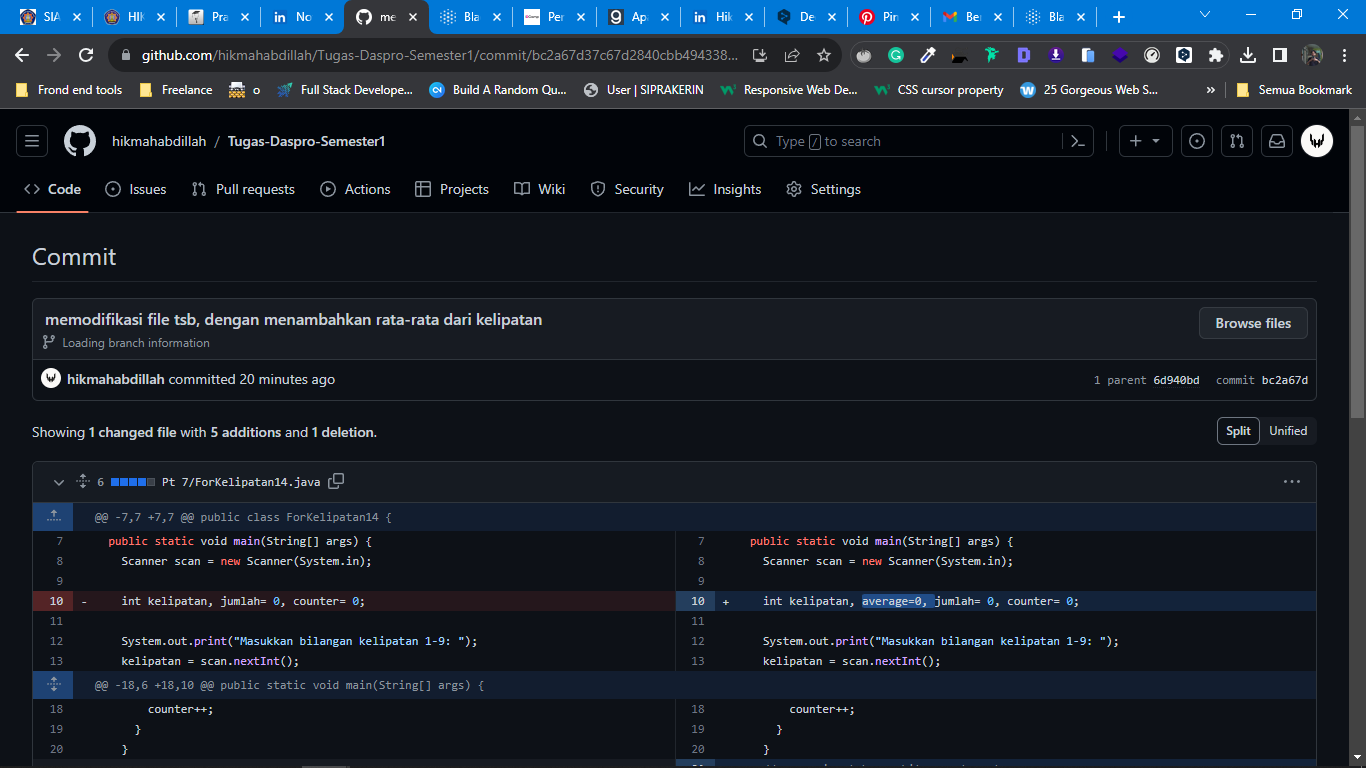
    System.out.println("Rata-rata dari seluruh bilangan kelipatan adalah " + average);

    System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);

    System.out.printf("Total bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, jumlah);

  }

}



4. Buatlah file baru dengan nama WhileKelipatanNoAbsen.java. Buatlah kode program

dengan tujuan serupa tetapi menggunakan WHILE. Push dan commit kode program ke

github.

**Jawab:**

import java.util.Scanner;

/\*\*

 \* WhileKelipatan14

 \*/

public class WhileKelipatan14 {

  public static void main(String[] args) {

     Scanner scan = new Scanner(System.in);

    int kelipatan, jumlah= 0, counter= 0;

    float average=0;

    System.out.print("Masukkan bilangan kelipatan 1-9: ");

    kelipatan = scan.nextInt();

    // for (int i = 1; i <= 50; i++) {

    //   if(i % kelipatan == 0){

    //     jumlah += i;

    //     counter++;

    //   }

    // }

    int i = 1;

    while (i <= 50) {

      if(i % kelipatan == 0){

        jumlah += i;

        counter++;

      }

      i++;

    }

    // operasi untuk menghitung rata rata

    average = (kelipatan \* jumlah) / counter;

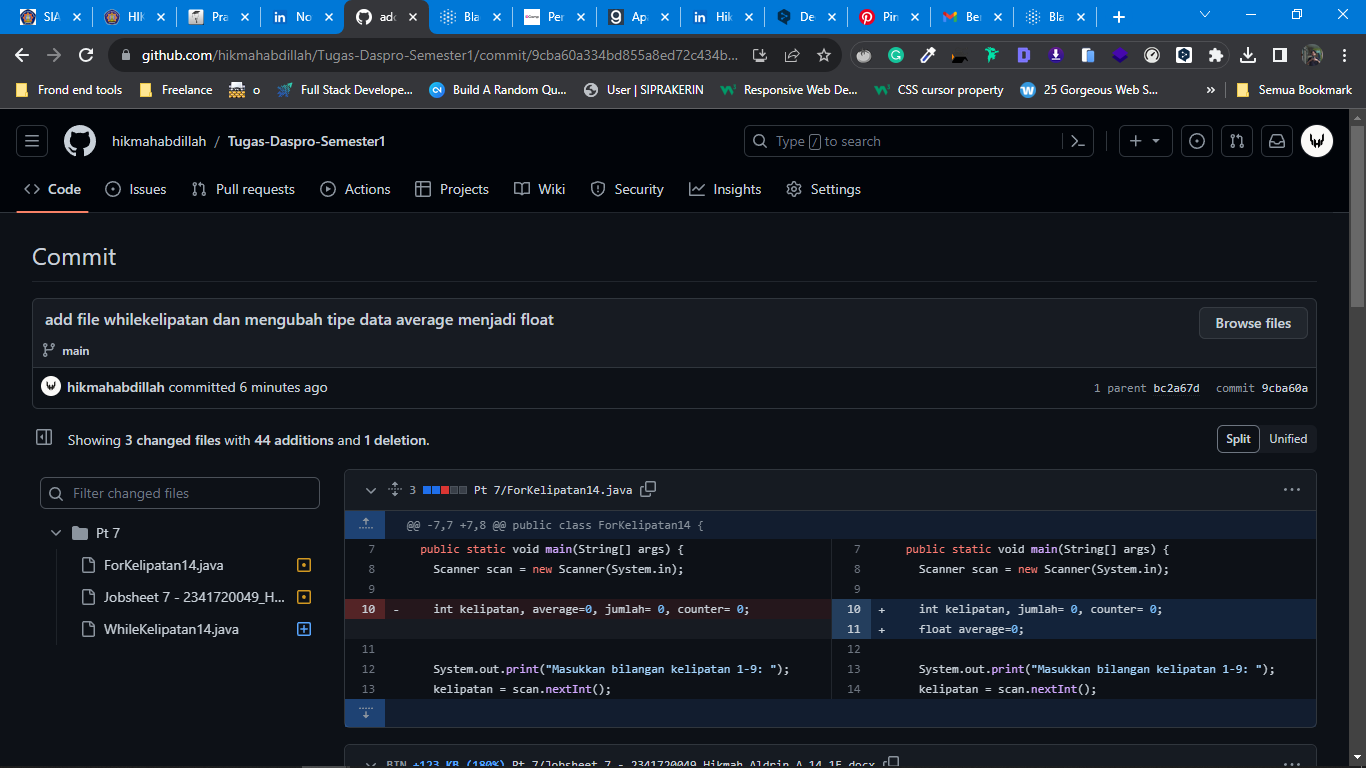
    System.out.println("Rata-rata dari seluruh bilangan kelipatan adalah " + average);

    System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);

    System.out.printf("Total bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, jumlah);

  }

}



**2.2 Percobaan 2: Menghitung Gaji Lembur Karyawan Menggunakan WHILE**

**dan CONTINUE**

**Pertanyaan**

1. Tunjukkan bagian kode program yang digunakan sebagai syarat untuk menghentikan

perulangan WHILE! Berapa kali perulangan dilakukan?

**Jawab:**

while (i < jumlahkaryawan) perulangan akan berhenti sesuai dengan nilai pada jumlah karyawan. Perulangan akan dilakukan sebanyak 3 kali (jika jumlah karyawan = 3)

2. Pada potongan kode berikut,

Apa yang sebenarnya terjadi jika variabel jabatan berisi nilai “DIREKTUR”? Apa peran

CONTINUE yang dituliskan di dalam sintaks perulangan?

**Jawab:**

Jika variable jabatan berisi nilai “DIREKTUR” , maka tidak terjadi apa apa karna menerapkan **equalsIgnore.**

Peran CONTINUE adalah untuk menghentikan kondisi/statement saat ini dan melanjutkan ke kondisi/statement berikutnya.

3. Mengapa komponen update i++ diletakkan di posisi tengah, tidak di bagian akhir

statement? Pindahkan i++ di bagian akhir, lalu jalankan kembali program dengan

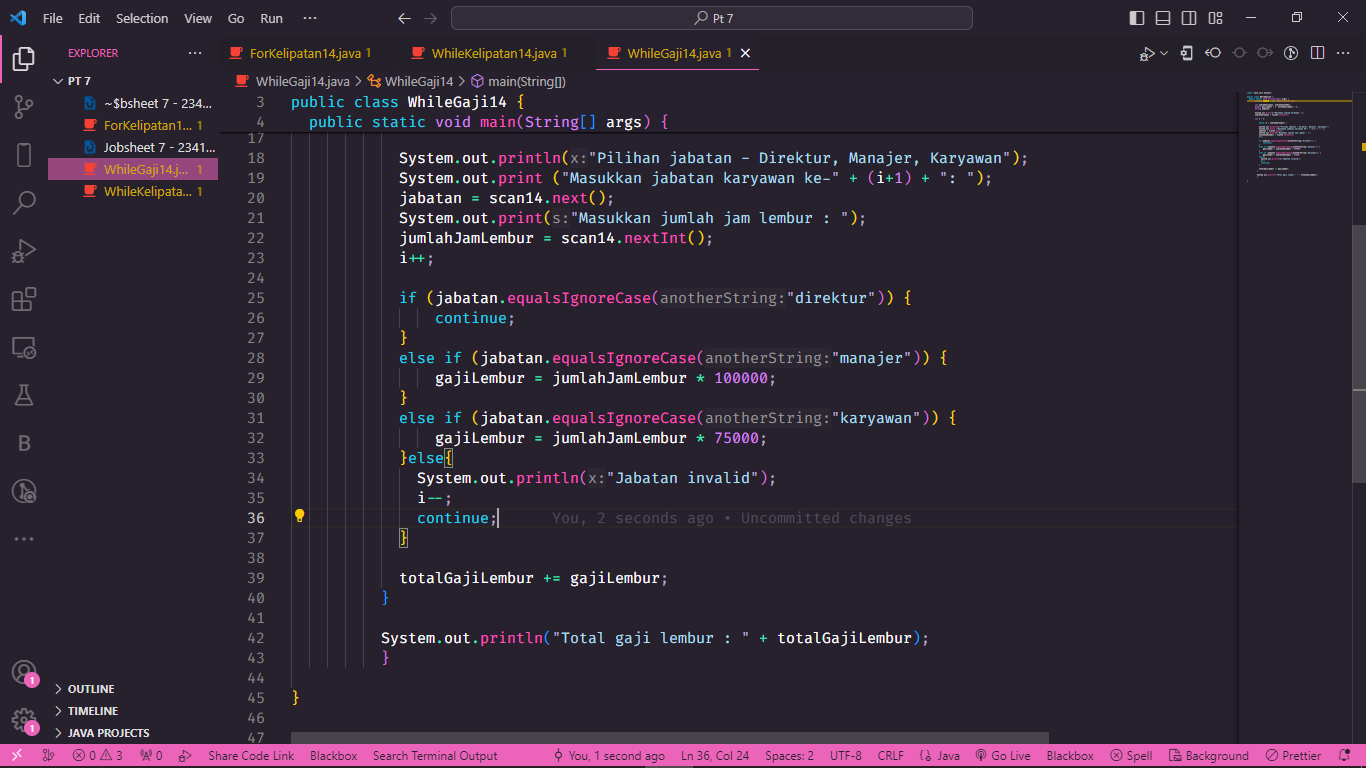
memasukkan “direktur” sebagai jabatan karyawan pertama. Apa yang terjadi? Jelaskan!

**Jawab:**

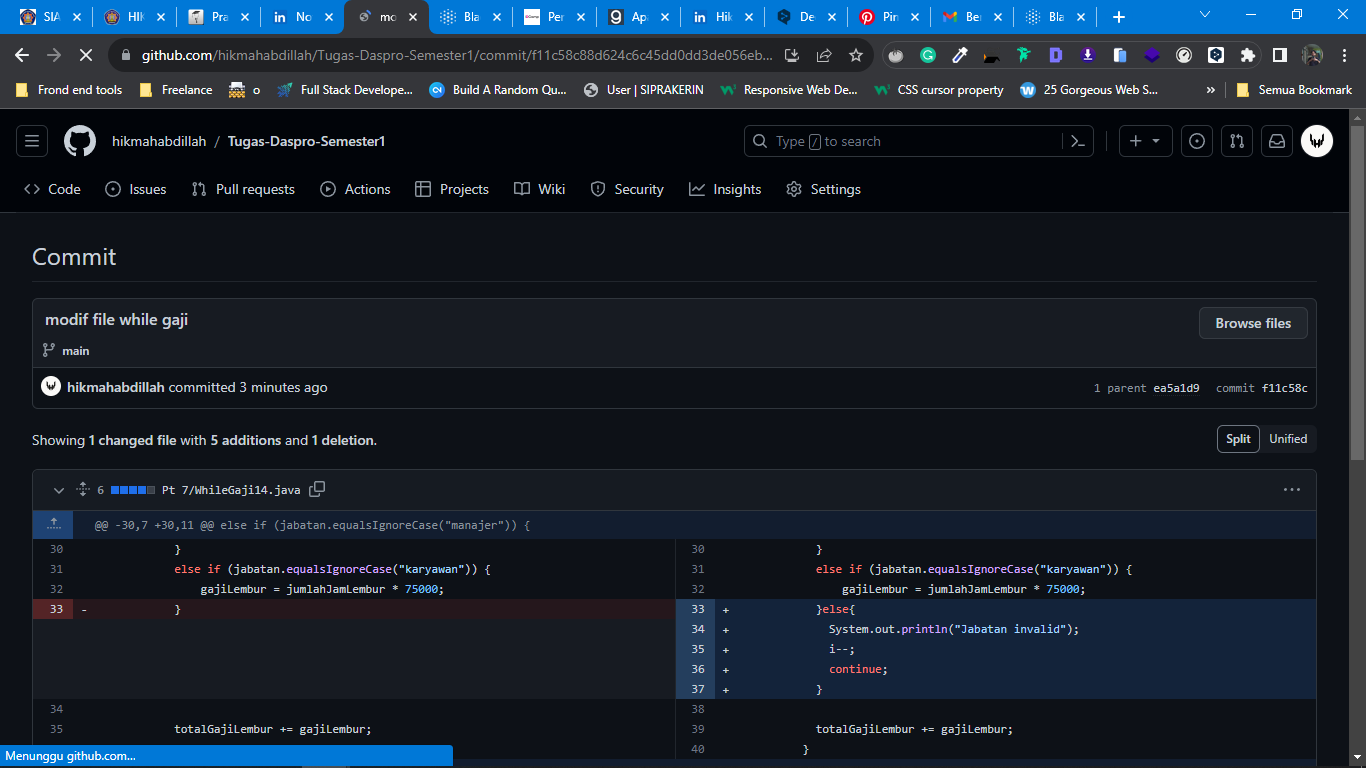
Update i++ diletakkan diposisi tengah karena penghitungan **i** akan terjadi setiap kali iterasi, sebelum kita memeriksa jenis jabatan yang dimasukkan pengguna.

Jika pindahkan i++ dibagian akhir, program akan terus berjalan tanpa henti, karena direktur dikecualikan dari perhitungan gaji lembur dengan pernyataan **if (jabatan.equalsIgnoreCase("direktur"))**. Oleh karena itu, program akan terus menanyakan jenis jabatan tanpa henti dan tidak akan pernah mengakhiri loop.

4. Modifikasi kode program untuk menghandle jabatan yang invalid seperti contoh berikut:



5. Push dan commit kode program ke github



**2.3 Percobaan 3: Menghitung Jatah Cuti Menggunakan DO-WHILE**

**Pertanyaan**

1. Apa kegunaan sintaks BREAK di dalam sintaks perulangan?

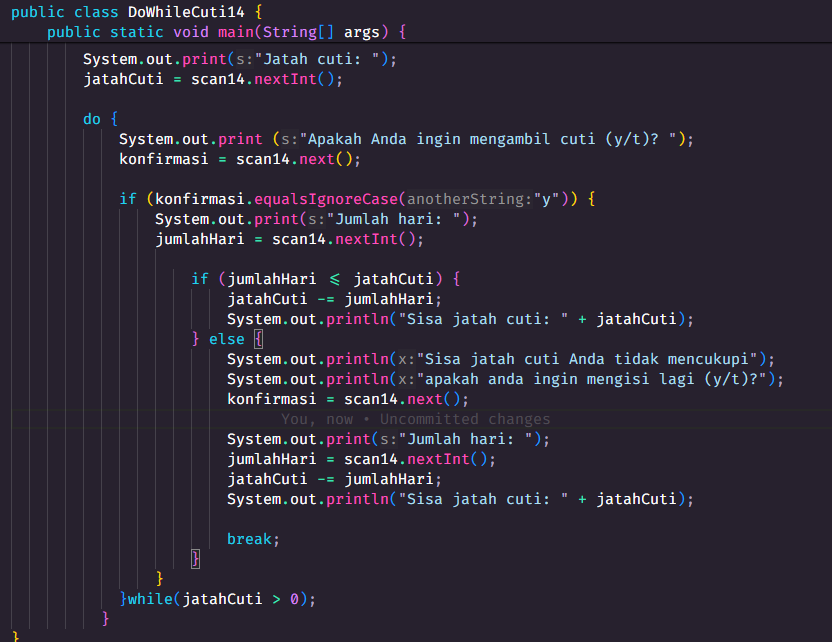
**Jawab:**

Untuk menghentikan suatu statement agar tidak melanjutkan ke statement selanjutnya.

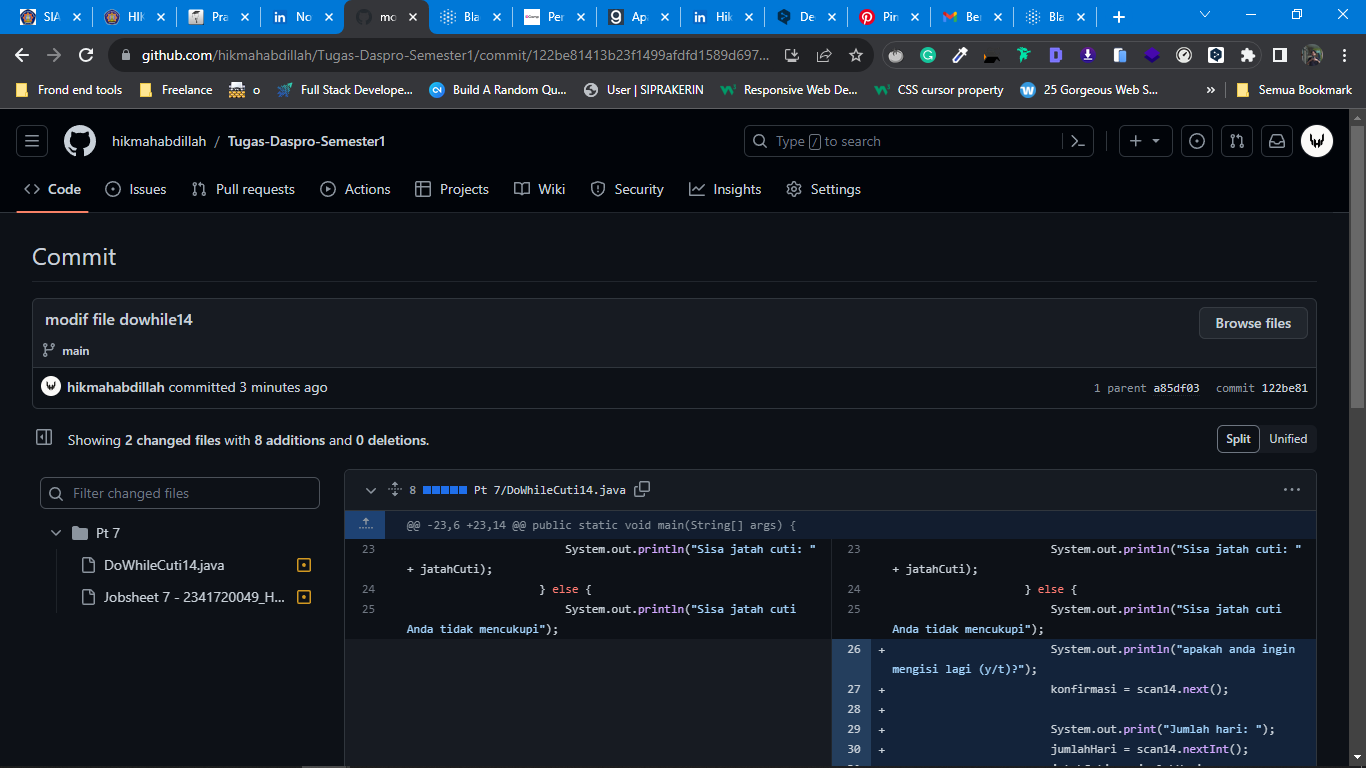
2. Modifikasi kode program sehingga jika jumlah hari cuti yang ingin diambil lebih besar

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?

**Jawab:**

Karena pada saat perulangan jika tidak ada break, maka kode akan terus mengulang.

5. Modifikasi kode program sehingga saat pengguna mengetikkan “t” sebagai input

konfirmasi, maka program akan berhenti

6. Push dan commit kode program ke github

**3. Tugas**

* Implementasikan flowchart yang telah dibuat pada tugas pertemuan 7 Matakuliah Dasar Pemrograman terkait project ke dalam kode program
* Push dan commit hasil kode program anda ke repository project Anda
* Catatan: tugas hanya boleh menerapkan materi dari pertemuan 1 hingga pertemuan 7.