

**LAPORAN PRAKTIKUM  
DASAR PEMROGRAMAN**

**JOBSHEET 7  
Perulangan 1**

**HIKMAH ALDRIN ABDILLAH**

**NIM: 2341720049**



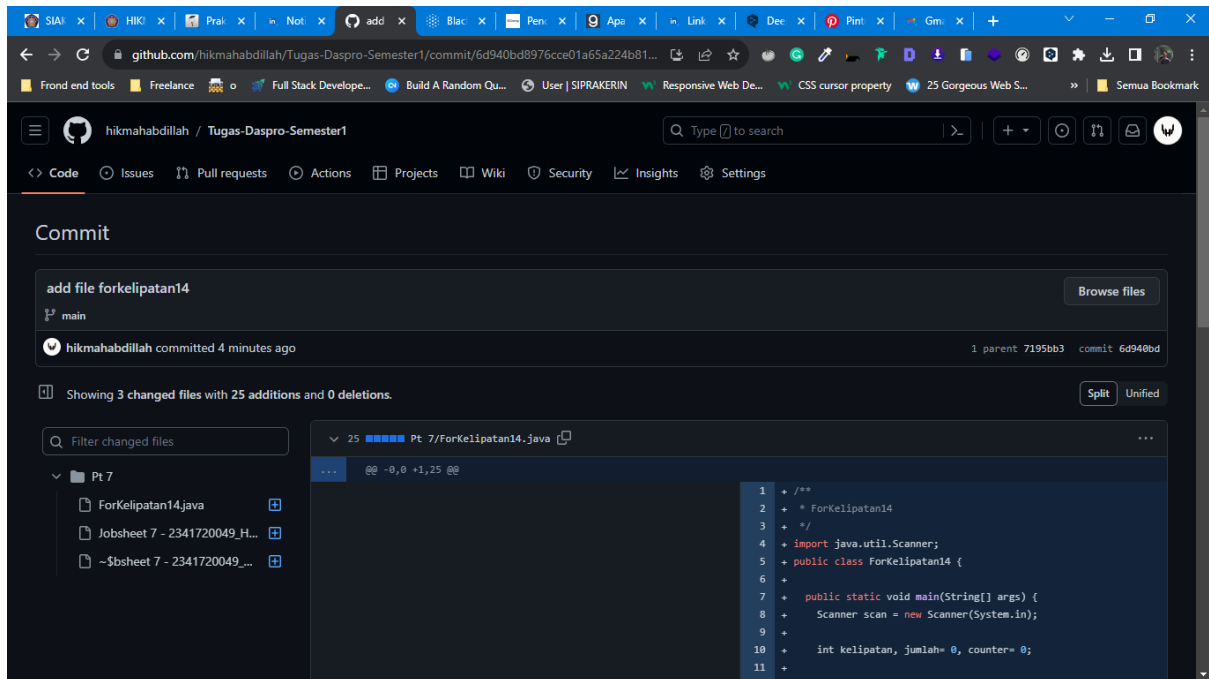
**PROGRAM STUDI D-IV TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG  
OKTOBER 2023**

## 2.1 Percobaan 1: Menghitung Bilangan Kelipatan Menggunakan FOR

```
Masukkan bilangan kelipatan 1-9: 5
Banyaknya bilangan 5 dari 1 sampai 50 adalah 10
Total bilangan 5 dari 1 sampai 50 adalah 275
PS D:\Kuliah\SEMESTER 1\Tugas-Daspro-Semester1\
```

## 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);

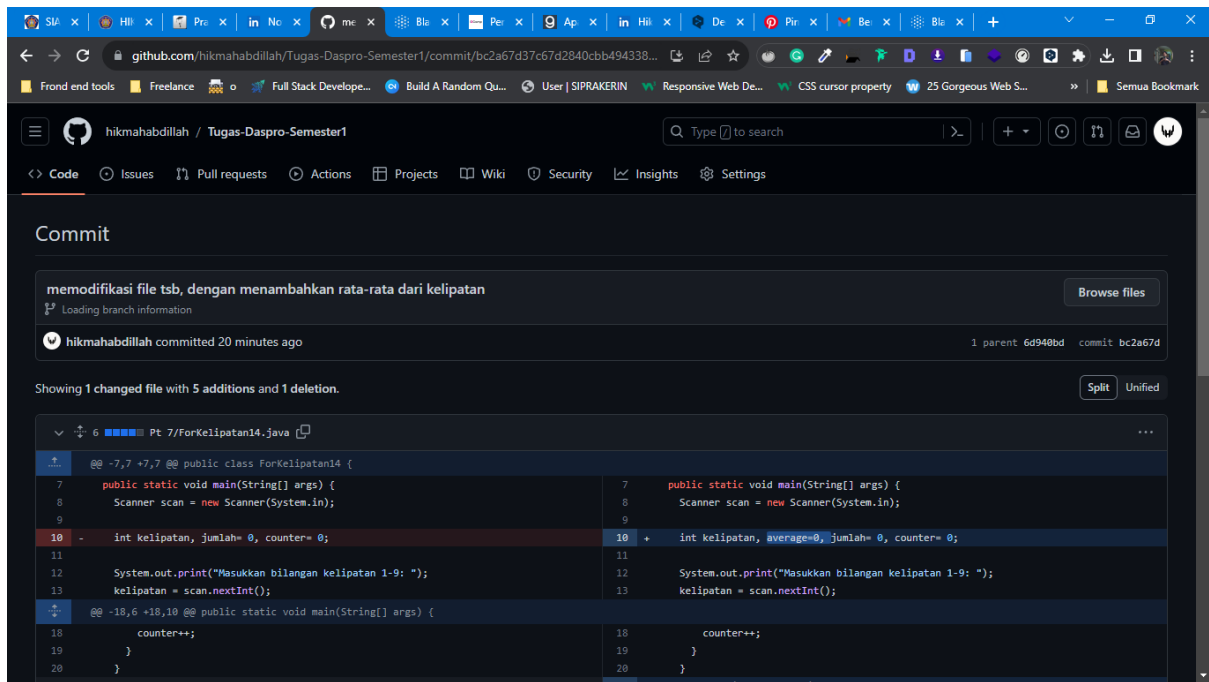
        int kelipatan, average=0, jumlah= 0, counter= 0;

        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;

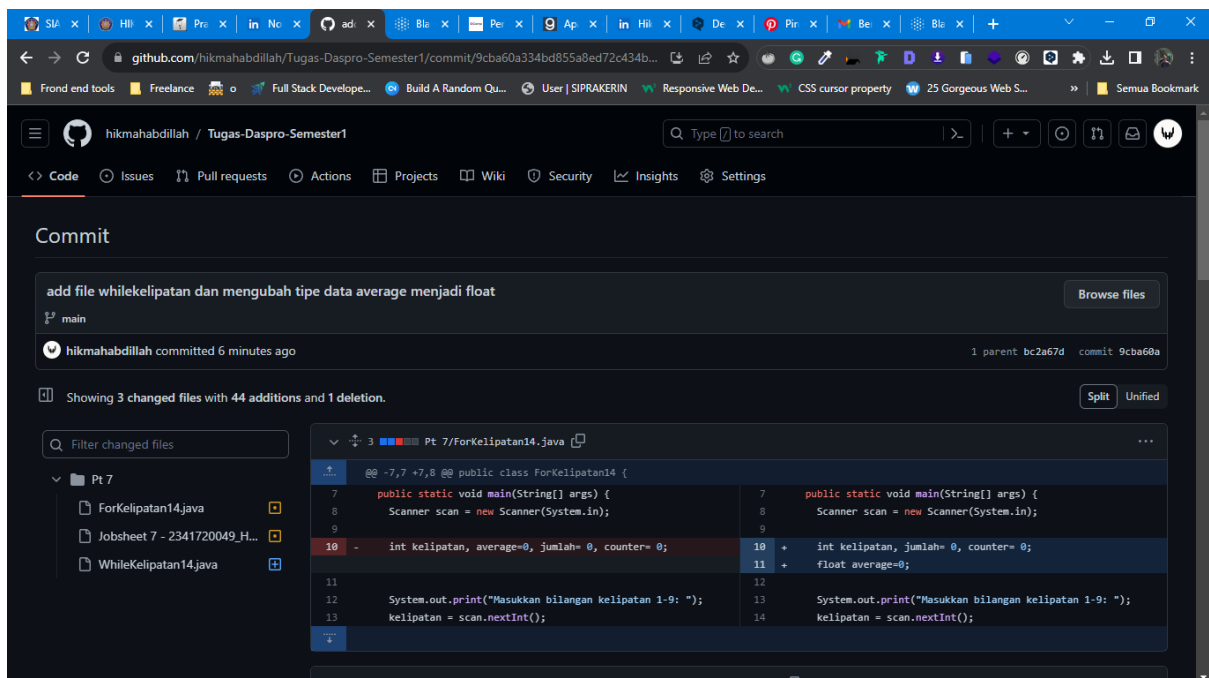
while (i <= 50) {
    if(i % kelipatan == 0){
        jumlah += i;
        counter++;
    }
    i++;
}

// 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.

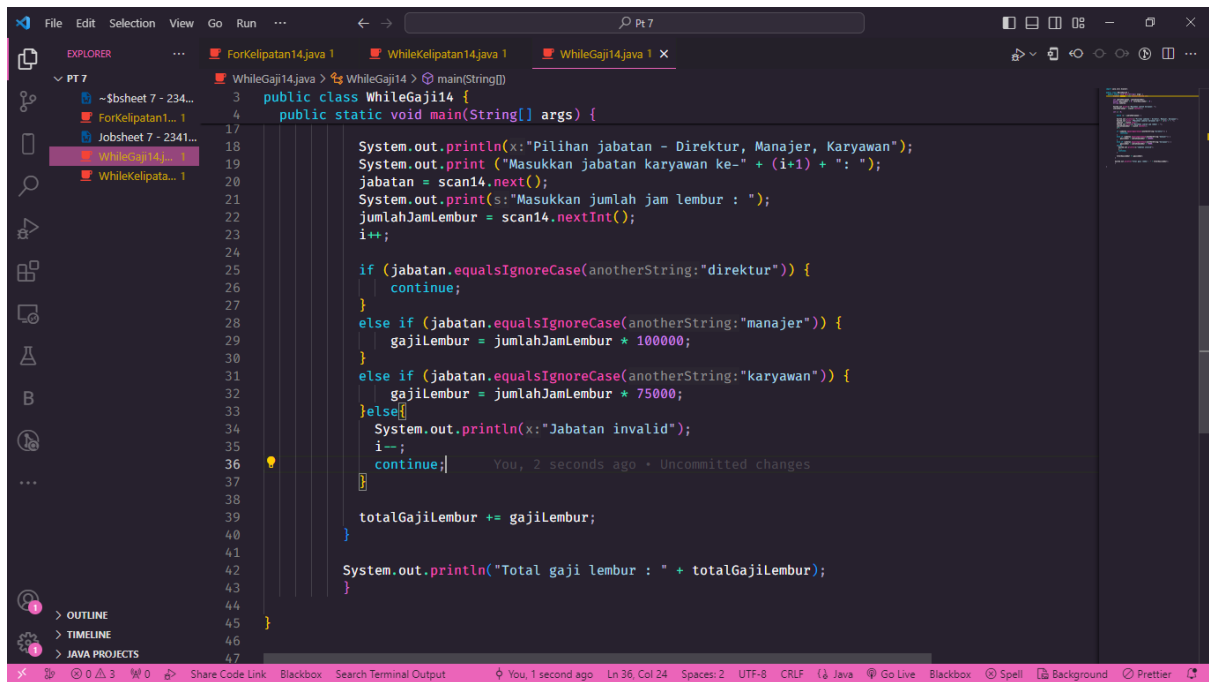
3. Mengapa komponen update i++ diletakkan di posisi tengah, tidak di bagian akhir statement? Pindahkan i++ di bagian akhir, lalu jalankan kembali program dengan memasukkan "direktur" sebagai jabatan karyawan pertama. Apa yang terjadi? Jelaskan!

#### 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 handle jabatan yang invalid seperti contoh berikut:



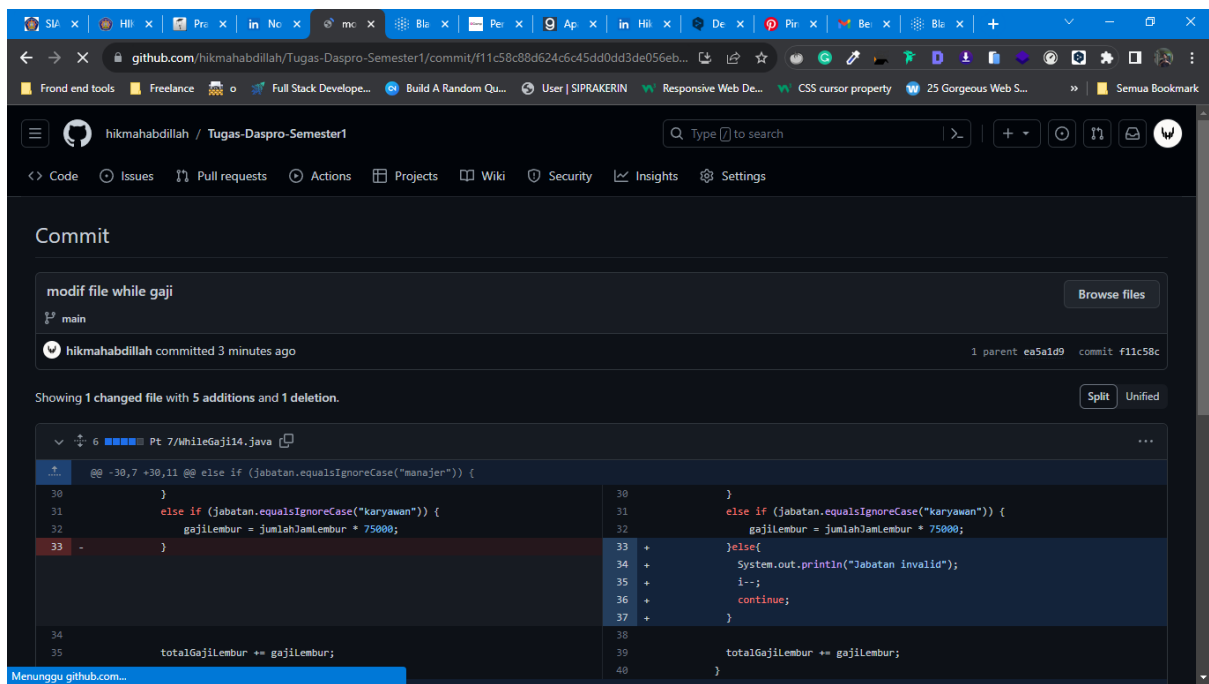
```
public class WhileGaji14 {
    public static void main(String[] args) {
        System.out.println("Pilihan jabatan - Direktur, Manajer, Karyawan");
        System.out.print("Masukkan jabatan karyawan ke- " + (i+1) + ": ");
        jabatan = scan14.next();
        System.out.print("Masukkan jumlah jam lembur : ");
        jumlahJamLembur = scan14.nextInt();
        i++;

        if (jabatan.equalsIgnoreCase("direktur")) {
            continue;
        } else if (jabatan.equalsIgnoreCase("manajer")) {
            gajilembur = jumlahJamLembur * 100000;
        } else if (jabatan.equalsIgnoreCase("karyawan")) {
            gajilembur = jumlahJamLembur * 75000;
        } else {
            System.out.println("Jabatan invalid");
            i--;
            continue;
        }

        totalGajilembur += gajilembur;
    }

    System.out.println("Total gaji lembur : " + totalGajilembur);
}
```

## 5. Push dan commit code program ke github



Commit

modif file while gaji

hikmahabdillah committed 3 minutes ago

Showing 1 changed file with 5 additions and 1 deletion.

File	Changes
Pt 7/WhileGaji14.java	<pre>@@ -30,7 +30,11 @@ else if (jabatan.equalsIgnoreCase("manajer")) { 30     } 31     else if (jabatan.equalsIgnoreCase("karyawan")) { 32         gajilembur = jumlahJamLembur * 75000; 33     } 34 35     totalGajilembur += gajilembur; 36 37     } 38 39     totalGajilembur += gajilembur; 40 }</pre>

## 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 mengisi jumlah hari sesuai jatah cuti.

```
public class DoWhileCuti14 {
    public static void main(String[] args) {
        System.out.print(s:"Jatah cuti: ");
        jatahCuti = scan14.nextInt();

        do {
            System.out.print (s:"Apakah Anda ingin mengambil cuti (y/t)? ");
            konfirmasi = scan14.next();

            if (konfirmasi.equalsIgnoreCase(anotherString:"y")) {
                System.out.print(s:"Jumlah hari: ");
                jumlahHari = scan14.nextInt();

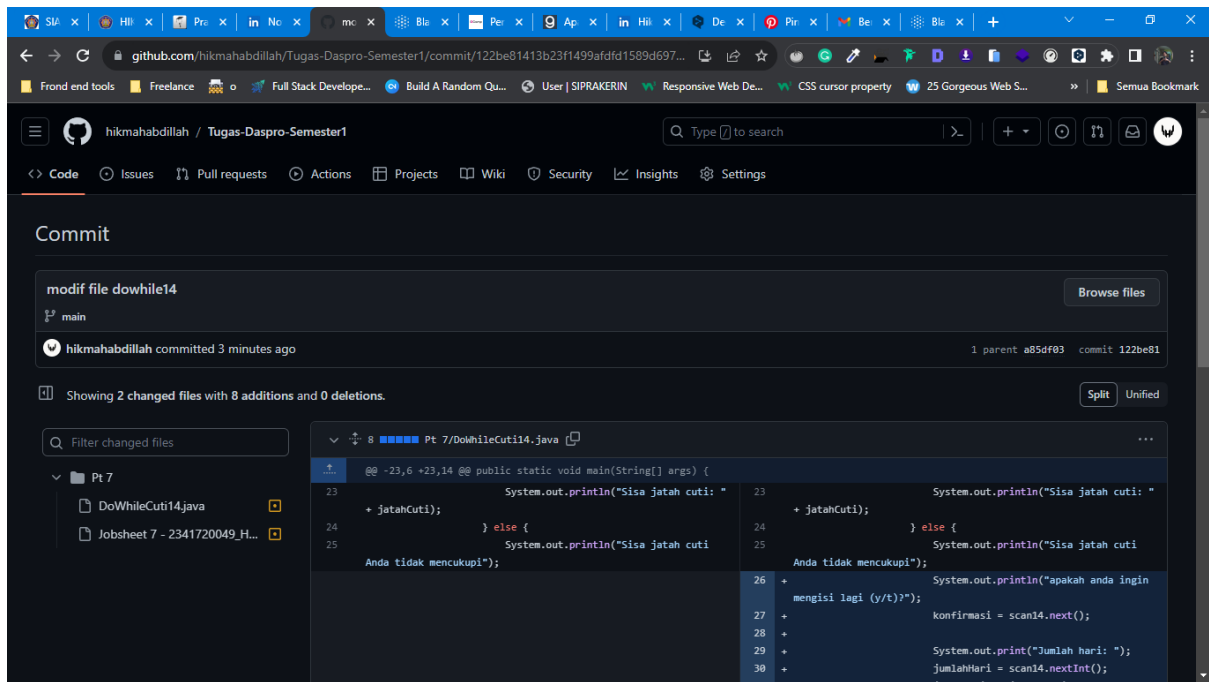
                if (jumlahHari <= jatahCuti) {
                    jatahCuti -= jumlahHari;
                    System.out.println("Sisa jatah cuti: " + jatahCuti);
                } else {
                    System.out.println(x:"Sisa jatah cuti Anda tidak mencukupi");
                    System.out.println(x:"apakah anda ingin mengisi lagi (y/t)?");
                    konfirmasi = scan14.next();

                    You, now * Uncommitted changes
                    System.out.print(s:"Jumlah hari: ");
                    jumlahHari = scan14.nextInt();
                    jatahCuti -= jumlahHari;
                    System.out.println("Sisa jatah cuti: " + jatahCuti);

                    break;
                }
            }
        }while(jatahCuti > 0);
    }
}
```

3. Push dan commit kode program ke github



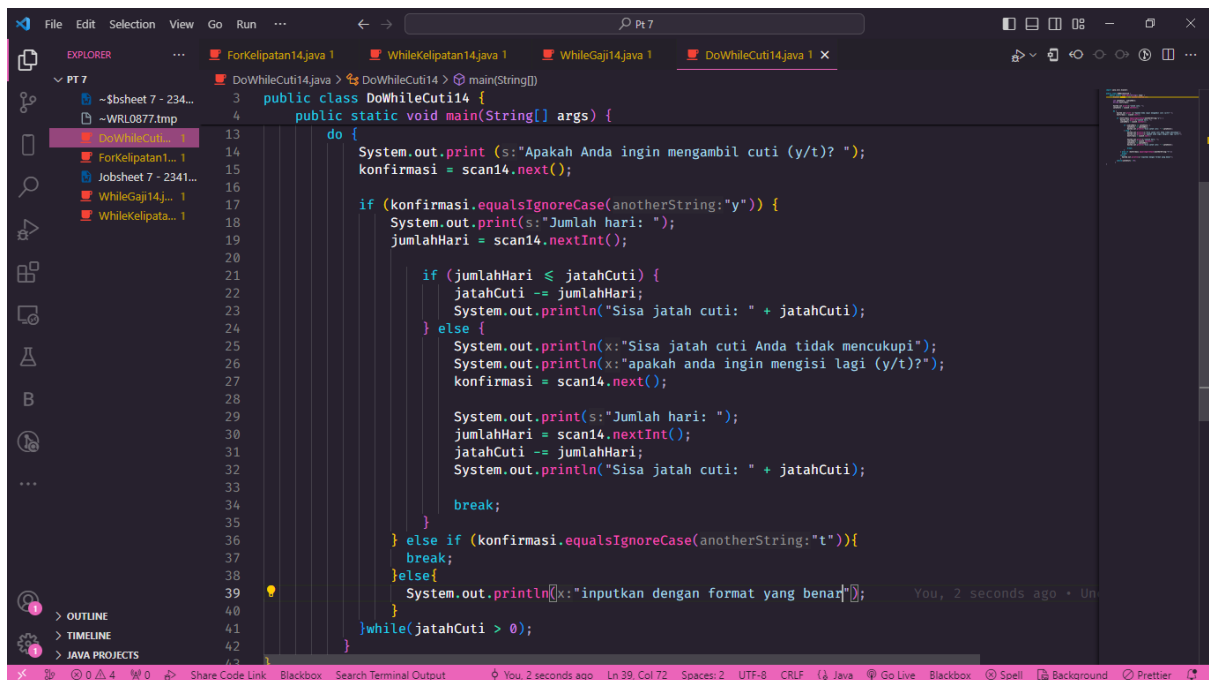


4. Pada saat input konfirmasi, ketikkan “t”, apa yang terjadi? Mengapa demikian?

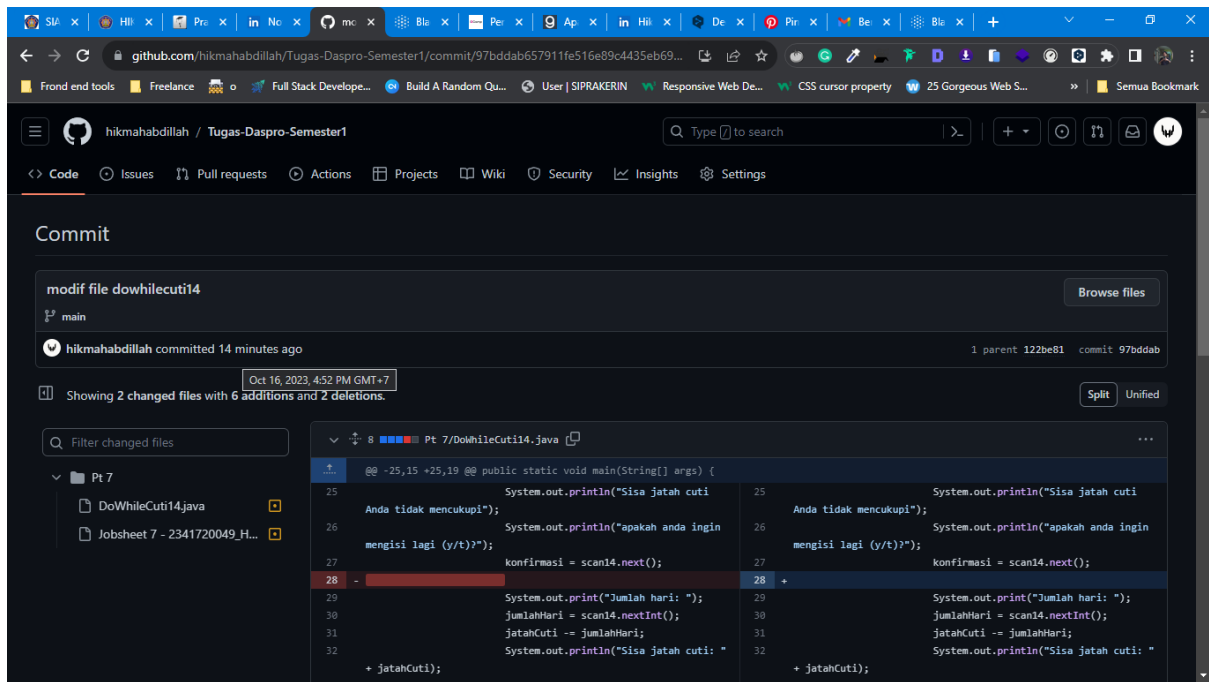
**Jawab:**

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.

### JAWAB:

```
import java.util.Scanner;

public class sistemakademikcobacoba {
    public static void main(String[] args) {
        String namamhs, jk, kelas, agama, alamat, tgl, email;
        int NIM, no_hp;

        Scanner inputScanner = new Scanner(System.in);

        System.out.println("");
        System.out.println("=====");
        System.out.println("=          Sistem Akademik          =");
        System.out.println("=          BERANDA ADMIN            =");
        System.out.println("-----");
        System.out.println("1. Input Data Mahasiswa");
        System.out.println("2. Update Data Mahasiswa");
        System.out.println("3. Cek Data Mahasiswa");
```

```

System.out.println("4. Input Nilai Mahasiswa");
System.out.println("5. Update Nilai Mahasiswa");
System.out.println("6. Cek Jadwal Perkuliahan");
System.out.println("7. Update KRS Mahasiswa");
System.out.println("8. Cek Data KRS Mahasiswa");
System.out.println("9. Contact");
System.out.println("0. LogOut\n");

System.out.print("Masukkan pilihan anda: ");
int choice = inputScanner.nextInt();
switch (choice) {
    case 1:
        do {
            System.out.println("\n");

            System.out.println("=====masukkan
data diri=====");

            inputScanner.nextLine();

            System.out.print("masukkan nama                :");
            namamhs = inputScanner.nextLine();

            System.out.print("masukkan NIM                :");
            NIM = inputScanner.nextInt();

            inputScanner.nextLine();

            System.out.print("masukkan Jenis Kelamin    :");
            jk = inputScanner.nextLine();

            System.out.print("masukkan kelas          :");
            kelas = inputScanner.nextLine();

            System.out.print("masukkan agama          :");
            agama = inputScanner.nextLine();

            System.out.print("masukkan tanggal lahir  :");
            tgl = inputScanner.nextLine();

            System.out.print("masukkan email          :");
            email = inputScanner.nextLine();

            System.out.print("masukkan nomor handpone  :");
            no_hp = inputScanner.nextInt();

```

```

        inputScanner.nextLine();

        System.out.print("masukkan alamat          :");
        alamat = inputScanner.nextLine();

        System.out.println("=====
=====");
        System.out.print("APAKAH ANDA INGIN MENAMBAHKAN
DATA MAHASISWA LAGI ?");

        System.out.println("\n");

        System.out.println("1. untuk ya");
        System.out.println("2. untuk tidak");

        choice = inputScanner.nextInt();

        if (choice == 2) {
            break;
        } else {
            System.out.println("pilihan tidak palid");
            System.out.println("\n");
            System.out.println("1. untuk ya");
            System.out.println("2. untuk tidak");

            choice = inputScanner.nextInt();
        }

        // switch (choice){
        //     case 2 :
        //         berandaAdmin();
        //         break;

        //     default:
        //         System.out.println("pilihan tidak palid");
        //         break;
        // }
    } while (choice == 1);
    break;
case 2:
    break;
case 3:
    break;
case 4:
    break;
case 5:

```

```

        break;
    case 6:
        break;
    case 7:
        break;
    case 8:
        break;
    case 9:
        break;
    case 0:
    default:
        System.out.println("Pilihan tidak valid.");
    }
}
}

```

The screenshot shows a web browser displaying the GitHub repository page for 'Sistem\_Akademik' by user 'tora180405'. The repository is public and has 0 stars. The 'Activity' tab is selected, showing a list of recent commits and pull requests. The activity includes:

- Merge pull request #7 from tora180405/hikmah (Pull request merge) - hikmahabdillah pushed 3 commits to main • 936d1bb...1163ca2 • 16 hours ago
- memodifikasi dan mensolved problem yang ada di fitur input data mahas... - hikmahabdillah pushed 1 commit to hikmah • 1cd106c...51b5a5b • 16 hours ago
- mengganti perulangan do while menjadi while - hikmahabdillah pushed 11 commits to hikmah • 574db0e...1cd106c • 17 hours ago
- merge branch tora ke branch main - tora180405 pushed 10 commits to main • 574db0e...936d1bb • 18 hours ago
- edit - tora180405 pushed 1 commit to tora • 7931010...49b608a • 18 hours ago
- menambah isi fitur tambah mahasiswa revisi - tora180405 pushed 1 commit to tora • 669c11c...7931010 • yesterday
- menambah isi fitur tambah mahasiswa - tora180405 pushed 3 commits to main • 936d1bb...1163ca2 • 16 hours ago

The bottom of the screen shows a Windows taskbar with various application icons and a system clock indicating 5:08 PM on 10/16/2023.