

LAPORAN PRAKTIKUM 7

PRAKTIKUM DASAR PEMROGRAMAN



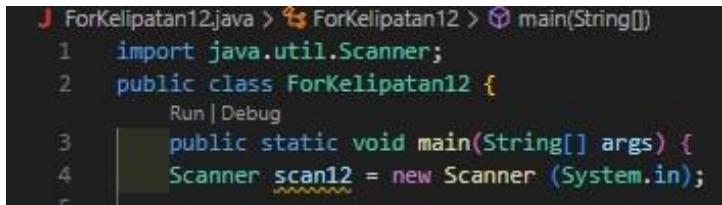
Nama: Farrel Augusta Dinata
NIM: 2341720081
Kelas: 1B
Prodi: D4 Teknik Informatika

Percobaan 1

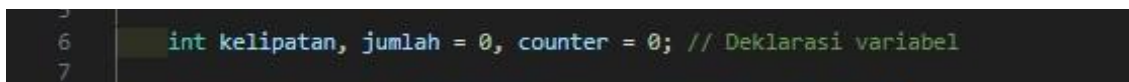
1. Membuat file baru dengan nama ForKelipatan12.java



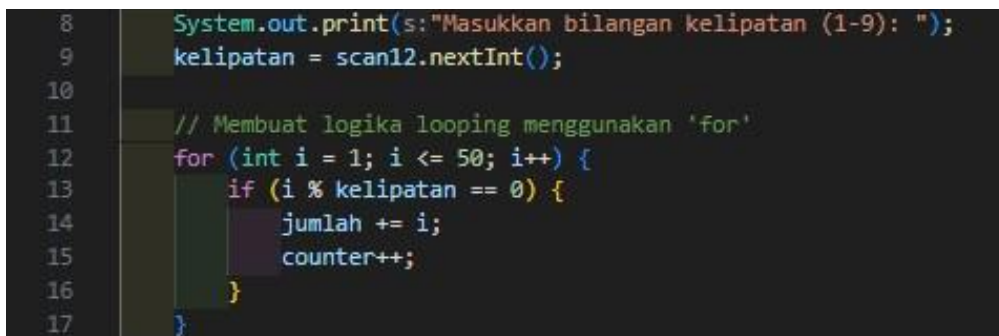
2. Menambahkan struktur dasar Java dan scanner



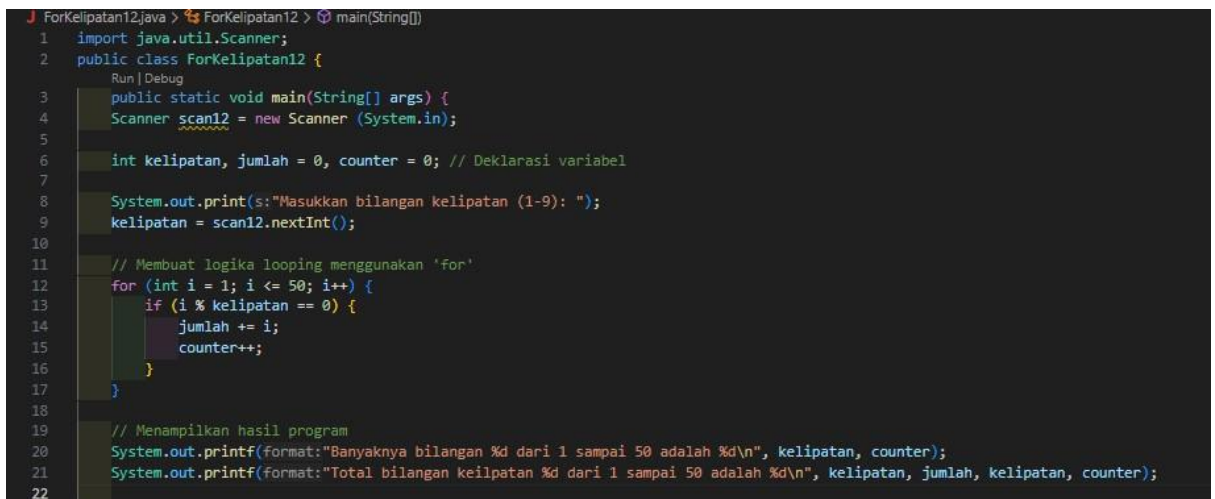
3. Deklarasi variabel



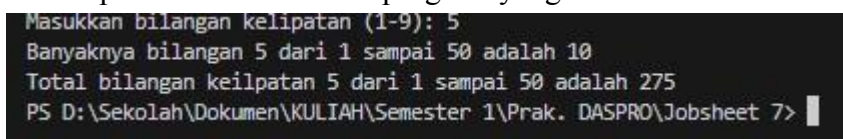
4. Membuat struktur perulangan dengan 'FOR'



5. Tampilan source code secara keseluruhan



6. Menampilkan hasil dari code program yang telah dibuat



7. Melakukan push dan commit ke GitHub

```
USER HP@LAPTOP-U2AGFHRQ MINGW64 /d/Sekolah/Dokumen/KULIAH/Semester 1/Prak. DASPR
0/Jobsheet 7
$ git init
Initialized empty Git repository in D:/Sekolah/Dokumen/KULIAH/Semester 1/Prak. D
ASPR0/Jobsheet 7/.git/

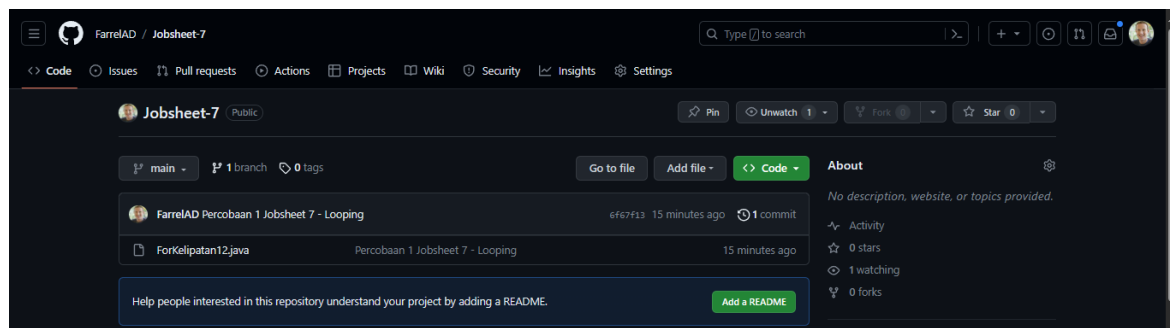
USER HP@LAPTOP-U2AGFHRQ MINGW64 /d/Sekolah/Dokumen/KULIAH/Semester 1/Prak. DASPR
0/Jobsheet 7 (master)
$ git add ForkKelipatan12.java

USER HP@LAPTOP-U2AGFHRQ MINGW64 /d/Sekolah/Dokumen/KULIAH/Semester 1/Prak. DASPR
0/Jobsheet 7 (master)
$ git commit -m "Percobaan 1 Jobsheet 7 - Looping"
[master (root-commit) 6f67f13] Percobaan 1 Jobsheet 7 - Looping
1 file changed, 28 insertions(+)
create mode 100644 ForkKelipatan12.java

USER HP@LAPTOP-U2AGFHRQ MINGW64 /d/Sekolah/Dokumen/KULIAH/Semester 1/Prak. DASPR
0/Jobsheet 7 (master)
$ git branch -M main

USER HP@LAPTOP-U2AGFHRQ MINGW64 /d/Sekolah/Dokumen/KULIAH/Semester 1/Prak. DASPR
0/Jobsheet 7 (main)
$ git remote add origin https://github.com/FarrelAD/Jobsheet-7.git

USER HP@LAPTOP-U2AGFHRQ MINGW64 /d/Sekolah/Dokumen/KULIAH/Semester 1/Prak. DASPR
0/Jobsheet 7 (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 620 bytes | 103.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/FarrelAD/Jobsheet-7.git
* [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```



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:

- a. Inisialisasi variabel

`int i = 1;` Inisialisasi integer i bernilai 1

- b. Kondisi

`i <= 50;` kondisi jika i kurang dari atau sama dengan 50

- c. Update

`i++` Apabila kondisinya true maka akan ada update berupa increment

2. Jelaskan alur kerja dari potongan kode program berikut!

```
for (int i = 1; i <= 50; i++) {  
    if (i % kelipatan == 0) {  
        total += i;  
        counter++;  
    }  
}
```

Jawab: Diawali dengan menginisialisasi berupa variabel integer *i* bernilai 1. Jika nilai dari *i* tersebut kurang dari atau sama dengan 50, maka proses tersebut terus berlanjut ke increment. Keadaan ini membuat nilai dari bertambah 1. Proses ini akan berulang seperti pada penghitungan kondisi $i \leq 50$.

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: Kodenya:

```
18 // Menghitung rata-rata  
19 if (counter != 0) {  
20     avg = jumlah / counter;  
21 } else {  
22     avg = 0;  
23 }
```

ForkKelipatan12.java

Update perhitungan rata-rata Percobaan 1 Jobsheet 7

2 minutes ago

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:

```
J WhileKelipatan12.java > WhileKelipatan12 > main(String[])  
1 import java.util.Scanner;  
2 public class WhileKelipatan12 {  
3     public static void main(String[] args) {  
4         Scanner scan12 = new Scanner (System.in);  
5  
6         int kelipatan, jumlah = 0, counter = 0, avg; // Deklarasi variabel  
7  
8         System.out.print(s:"Masukkan bilangan kelipatan (1-9): ");  
9         kelipatan = scan12.nextInt();  
10  
11         int i = 0;  
12         while (i <= 50) {  
13             if (i % kelipatan == 0) {  
14                 jumlah += i;  
15                 counter++;  
16             }  
17         }  
18         if (counter != 0) {  
19             avg = jumlah / counter;  
20         } else {  
21             avg = 0;  
22         }  
23         // Menampilkan hasil program  
24         System.out.printf(format:"Banyaknya bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);  
25         System.out.printf(format:"Total bilangan kelipatan %d dari 1 sampai 50 adalah %d\n", kelipatan, jumlah, kelipatan, counter);  
26         System.out.printf("Rata-rata dari seluruh bilangan kelipatan adalah: " + avg);  
27  
28     }  
29 }  
30
```


Percobaan 2

1. Membuat file baru bernama WhileGaji12.java



2. Menambahkan struktur dasar Java dan scanner

```
J WhileGaji12.java > WhileGaji12 > main(String[])
1  import java.util.Scanner;
2  public class WhileGaji12 {
    Run | Debug
3      public static void main(String[] args) {
4          Scanner scan12 = new Scanner (System.in);
5      }
```

3. Deklarasi variabel

```
6      // Deklarasi variabel int, double, String
7      int jumlahKaryawan, jumlahJamLembur;
8      double gajilembur = 0, totalGajilembur = 0;
9      String jabatan;
10 }
```

4. Membuat struktur perulangan 'WHILE'

```
11     // Proses input 'Jumlah Karyawan'
12     System.out.print(s:"Masukkan jumlah karyawan: ");
13     jumlahKaryawan = scan12.nextInt();
14
15     // Proses operasi looping menggunakan while
16     int i = 0;
17     while (i < jumlahKaryawan) {
18         System.out.println(x:"Pilihan jabatan - Direktur, Manajer, Karyawan");
19         System.out.print("Masukkan jabatan karyawan ke-" + (i+1) + ": ");
20         jabatan = scan12.next();
21         System.out.print(s:"Masukkan jumlah jam lembur: ");
22         jumlahJamLembur = scan12.nextInt();
23         i++;
24
25         // Logika pemilihan di dalam while terkait jabatan yang dipilih
26         if (jabatan.equals(anObject:"direktur")) {
27             continue;
28         } else if (jabatan.equals(anObject:"manajer")) {
29             gajilembur = jumlahJamLembur * 100000;
30         } else if (jabatan.equals(anObject:"karyawan")) {
31             gajilembur = jumlahJamLembur * 75000;
32         }
33
34         // Proses penghitungan 'Gaji Lembur'
35         totalGajilembur += gajilembur;
36     }
37
38     // Menampilkan hasil perhitungan jumlah gaji
39     System.out.println("Total gaji lembur: " + totalGajilembur);
40
41
42 }
43
44 }
```


5. Menampilkan hasil dari code yang telah dibuat

```
Masukkan jumlah karyawan: 3
Pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-1: manajer
Masukkan jumlah jam lembur: 1
Pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-2: direktur
Masukkan jumlah jam lembur: 10
Pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-3: karyawan
Masukkan jumlah jam lembur: 5
Total gaji lembur: 475000.0
PS D:\Sekolah\Dokumen\KULIAH\Semester 1\Prak. DASPRO\Jobsheet 7>
```

6. Melakukan push dan commit ke GitHub

```
USER HP@LAPTOP-U2AGFHRQ MINGW64 /d/Sekolah/Dokumen/KULIAH/Semester 1/Prak. DASPRO
0/Jobsheet 7 (main)
$ git init
Reinitialized existing Git repository in D:/Sekolah/Dokumen/KULIAH/Semester 1/Prak. DASPRO/Jobsheet 7/.git/

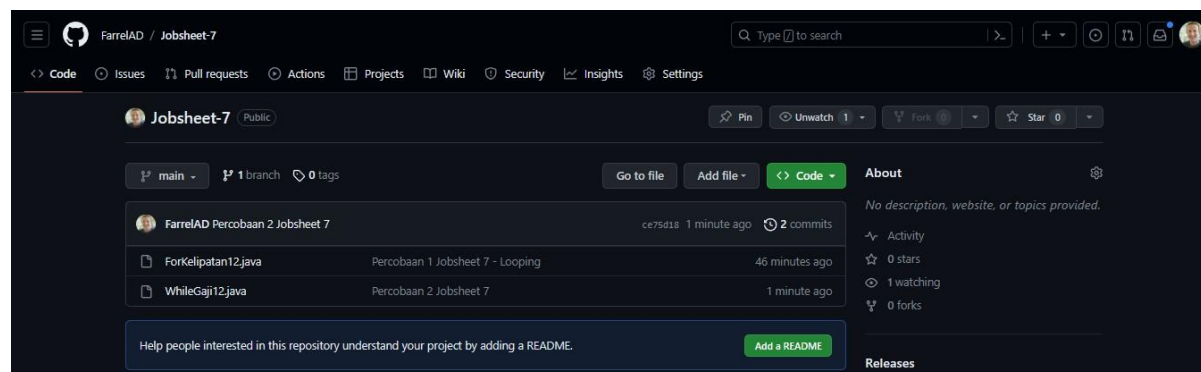
USER HP@LAPTOP-U2AGFHRQ MINGW64 /d/Sekolah/Dokumen/KULIAH/Semester 1/Prak. DASPRO/Jobsheet 7 (main)
$ git add WhileGaji12.java

USER HP@LAPTOP-U2AGFHRQ MINGW64 /d/Sekolah/Dokumen/KULIAH/Semester 1/Prak. DASPRO/Jobsheet 7 (main)
$ git commit -m "Percobaan 2 Jobsheet 7"
[main ce75d18] Percobaan 2 Jobsheet 7
1 file changed, 43 insertions(+)
create mode 100644 WhileGaji12.java

USER HP@LAPTOP-U2AGFHRQ MINGW64 /d/Sekolah/Dokumen/KULIAH/Semester 1/Prak. DASPRO/Jobsheet 7 (main)
$ git branch -M main

USER HP@LAPTOP-U2AGFHRQ MINGW64 /d/Sekolah/Dokumen/KULIAH/Semester 1/Prak. DASPRO/Jobsheet 7 (main)
$ git remote add origin https://github.com/FarrelAD/Jobsheet-7.git
error: remote origin already exists.

USER HP@LAPTOP-U2AGFHRQ MINGW64 /d/Sekolah/Dokumen/KULIAH/Semester 1/Prak. DASPRO/Jobsheet 7 (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 856 bytes | 214.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/FarrelAD/Jobsheet-7.git
6f67f13..ce75d18 main -> main
branch 'main' set up to track 'origin/main'.
```



Pertanyaan:

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

Jawab: Pada bagian ini:

```
// Proses input 'Jumlah Karyawan'
System.out.print(s:"Masukkan jumlah karyawan: ");
jumlahKaryawan = scan12.nextInt();

// Proses operasi looping menggunakan while
int i = 0;
while (i < jumlahKaryawan) {
```

Jadi proses 'WHILE' tergantung pada jumlah bilangan yang diinput pada jumlahKaryawan = scan12.nextInt();. Itu karena proses pada while bergantung pada input tersebut.

2. Pada potongan kode berikut,

```
if (jabatan.equalsIgnoreCase("direktur")) {
    continue;
```

Apa yang sebenarnya terjadi jika variabel jabatan berisi nilai "DIREKTUR"?

Apa peran CONTINUE yang dituliskan di dalam sintaks perulangan?

Jawab:

- Yang terjadi adalah sama saja seperti sebelumnya. Itu karena menggunakan kondisi equalsIgnoreCase. Jadi apabila karakter yang ditulis menggunakan kapital atau tidak akan dianggap sama saja.
- Peran CONTINUE adalah untuk menghentikan proses if sebelumnya dan melanjutkan kondisi if setelahnya

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: Apabila komponen update itu berada di akhir, saat proses ingin berakhir, sistem akan terus menjalankan keadaan if. Itu akan membuat sistemnya tidak terbatas

4. Modifikasi kode program untuk handle jabatan yang invalid seperti contoh berikut!

Jawab: Merubah kode yang berada di dalam while menjadi:

```
int i = 0;
while (i < jumlahKaryawan) {
    System.out.println(x:"Pilihan jabatan - Direktur, Manajer, Karyawan");
    System.out.print("Masukkan jabatan karyawan ke-" + (i + 1) + ": ");
    jabatan = scan12.next();
    System.out.print(s:"Masukkan jumlah jam lembur: ");
    jumlahJamLembur = scan12.nextInt();

    // Logika pemilihan di dalam while terkait jabatan yang dipilih
    if (jabatan.equalsIgnoreCase(anotherString:"direktur")) {
        // Tidak ada gaji lembur untuk direktur
    } else if (jabatan.equalsIgnoreCase(anotherString:"manajer")) {
        gajilembur = jumlahJamLembur * 100000;
    } else if (jabatan.equalsIgnoreCase(anotherString:"karyawan")) {
        gajilembur = jumlahJamLembur * 75000;
    } else {
        System.out.println(x:"Jabatan Invalid");
        continue; // Mengulangi lagi ke tahap input jenis karyawan
    }

    // Proses penghitungan 'Gaji Lembur'
    totalGajilembur += gajilembur;

    i++; // Perlu menginkrementasi i untuk menghindari loop tak terbatas
}
```

```
Masukkan jabatan karyawan ke-1: manajer
Masukkan jumlah jam lembur: 10
Pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-2: direktur
Masukkan jumlah jam lembur: 5
Pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-3: pegawai
Masukkan jumlah jam lembur: 4
Jabatan Invalid
Pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-3: karyawan
Masukkan jumlah jam lembur: 4
Total gaji lembur: 2300000.0
PS D:\Sekolah\Dokumen\KULIAH\Semester 1\Prak. DASPRO\Jobsheet 7>
```

5. Push dan commit ke GitHub

WhileGaji12.java	Update program invalid	now
------------------	------------------------	-----

Percobaan 3

1. Membuat file baru bernama DoWhileCuti12.java



2. Membuat struktur dasar Java dan scanner

```
1 import java.util.Scanner;
2 public class DoWhileCuti12 {
3     public static void main(String[] args) {
4         Scanner scan12 = new Scanner(System.in);
```

3. Deklarasi variabel

```
6 // Deklarasi variabel
7 int jatahCuti, jumlahHari;
8 String konfirmasi;
```

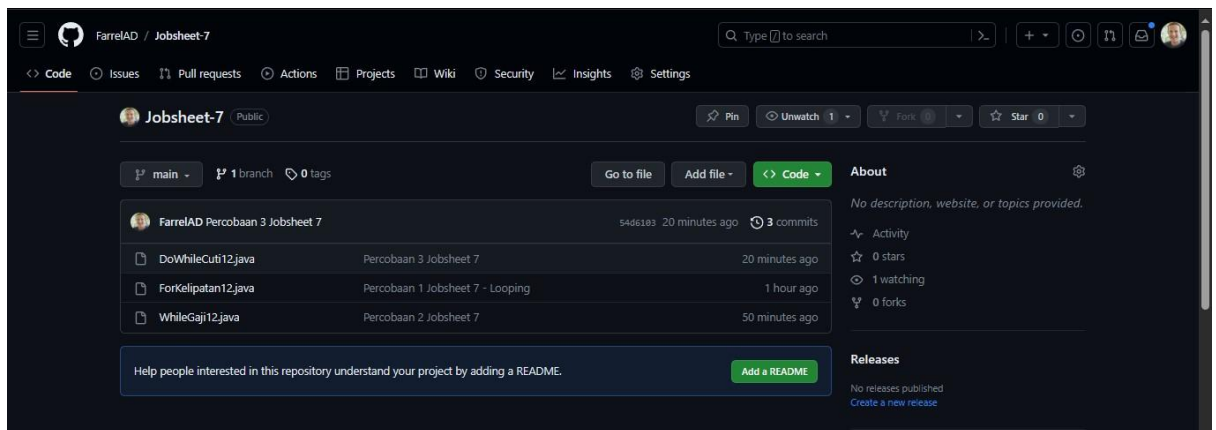
4. Membuat struktur perulangan 'DO-WHILE'

```
13 // Proses looping menggunakan 'DO-WHILE'
14 do {
15     System.out.print(s:"Apakah Anda ingin mengambil (y/t)?");
16     konfirmasi = scan12.next();
17
18     if (konfirmasi.equalsIgnoreCase (anotherString:"y")) {
19         System.out.print(s:"Jumlah hari: ");
20         jumlahHari = scan12.nextInt();
21
22         if (jumlahHari <= jatahCuti) {
23             jatahCuti -= jumlahHari;
24             System.out.println("Sisa jatah cuti: " + jatahCuti);
25         } else {
26             System.out.println(x:"Sisa jatah cuti Anda tidak mencukupi");
27             break;
28         }
29     }
30 } while (jatahCuti > 0);
```

5. Menampilkan hasil dari program yang telah dibuat

```
Jatah cuti: 12
Apakah Anda ingin mengambil (y/t)?y
Jumlah hari: 4
Sisa jatah cuti: 8
Apakah Anda ingin mengambil (y/t)?y
Jumlah hari: 5
Sisa jatah cuti: 3
Apakah Anda ingin mengambil (y/t)?y
Jumlah hari: 4
Jumlah hari: 4
Sisa jatah cuti Anda tidak mencukupi
PS D:\Sekolah\Dokumen\KULIAH\Semester 1\Prak. DASPRO\Jobsheet 7>
```

6. Melakukan push dan commit ke GitHub



Pertanyaan:

1. Apa kegunaan sintaks BREAK di dalam sintaks perulangan?

Jawab: Untuk menghentikan perulangan yang terjadi pada bagian if.

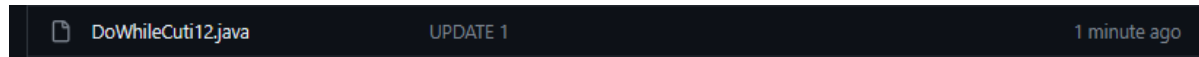
2. Modifikasi kode program sehingga jika jumlah hari cuti yang ingin diambil lebih besar daripada jatah yang tersisa, program tidak berhenti sehingga pengguna masih memiliki kesempatan untuk mengisi jumlah hari sesuai jatah cuti!

Jawab:

```
14 // Proses looping menggunakan 'DO-WHILE'
15 do {
16     System.out.print(s:"Apakah Anda ingin mengambil cuti (y/t)? ");
17     konfirmasi = scan12.next();
18
19     if (konfirmasi.equalsIgnoreCase(anotherString:"y")) {
20         System.out.print(s:"Jumlah hari: ");
21         jumlahHari = scan12.nextInt();
22
23         if (jumlahHari <= jatahCuti) {
24             jatahCuti -= jumlahHari;
25             System.out.println("Sisa jatah cuti: " + jatahCuti);
26         } else {
27             System.out.println(x:"Sisa jatah cuti Anda tidak mencukupi.");
28             System.out.print(s:"Apakah Anda ingin mencoba lagi (y/t)? ");
29             String cobaLagi = scan12.next();
30             if (cobaLagi.equalsIgnoreCase(anotherString:"t")) {
31                 break; // Menghentikan loop jika pengguna memilih 't'
32             }
33         }
34     } else if (konfirmasi.equalsIgnoreCase(anotherString:"t")) {
35         break; // Menghentikan loop jika pengguna memilih 't'
36     } else {
37         System.out.println(x:"Pilihan tidak valid. Masukkan 'y' atau 't'.");
38     }
39 } while (jatahCuti > 0);
40
41 System.out.println(x:"Anda telah menggunakan semua jatah cuti Anda.");
```

3. Push dan commit kode program ke github!

Jawab:

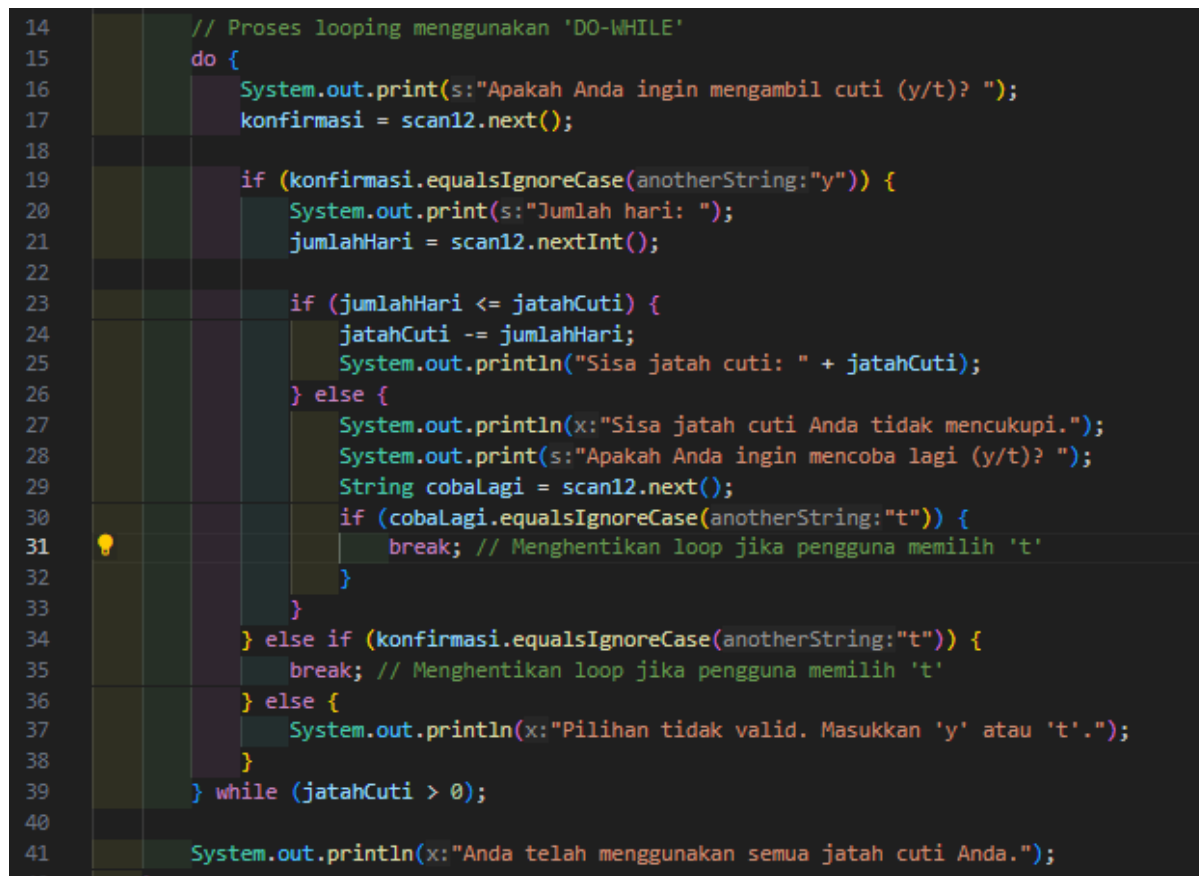


4. Pada saat input konfirmasi, ketikkan “t”, apa yang terjadi? Mengapa demikian?

Jawab: Akan muncul tulisan: Anda telah menggunakan jatah cuti anda

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

Jawab:



6. Push dan commit kode program ke github!

Jawab:



BAGIAN 2

Tugas Mandiri

- a. Pengguna memasukkan nama dan jenis kelamin dari 30 mahasiswa di suatu kelas
Nama-nama mahasiswa yang ditampilkan hanya yang berjenis kelamin perempuan!

Jawab:

Source code yang dibuat

```
Pertemuan 7 > J Individu1_12.java > Individu1_12 > main(String[])
1  import java.util.Scanner;
2  public class Individu1_12 {
    Run | Debug
3      public static void main(String[] args) {
4          Scanner sc12 = new Scanner (System.in);
5
6          int jumlahMahasiswa = 30;
7          String namaMahasiswa;
8          String jenisKelamin;
9          int i = 0;
10
11         do {
12             System.out.print("Masukkan nama mahasiswa ke " + (i+1) + " ");
13             namaMahasiswa = sc12.nextLine();
14             System.out.print("Masukkan jenis kelamin (L/P) mahasiswa ke " + (i+1) + " ");
15             jenisKelamin = sc12.nextLine();
16             if (jenisKelamin.equalsIgnoreCase("P")) {
17                 System.out.println("Mahasiswa perempuan dengan nama " + namaMahasiswa);
18             }
19             i++;
20         } while (i < jumlahMahasiswa);
21
22     }
23
24 }
```

Hasil yang dapat dihasilkan

```
Masukkan nama mahasiswa ke 1 Ari
Masukkan jenis kelamin (L/P) mahasiswa ke 1 L
Masukkan nama mahasiswa ke 2 Ani
Masukkan jenis kelamin (L/P) mahasiswa ke 2 P
Mahasiswa perempuan dengan nama Ani
Masukkan nama mahasiswa ke 3 Bagus
Masukkan jenis kelamin (L/P) mahasiswa ke 3 L
Masukkan nama mahasiswa ke 4 Rani
Masukkan jenis kelamin (L/P) mahasiswa ke 4 P
Mahasiswa perempuan dengan nama Rani
Masukkan nama mahasiswa ke 5 Agus
Masukkan jenis kelamin (L/P) mahasiswa ke 5 L
PS D:\Sekolah\Dokumen\KULIAH\Semester 1\DASPRO> |
```

Untuk percobaan dilakukan dengan men-setting jumlah Mahasiswa sebanyak 5. Namun kurang lebih hasil yang didapatkan sama. Pengguna dapat menginput nama mahasiswa beserta jenis kelaminnya hingga sebanyak 30 kali. Apabila pengguna menginputkan nama mahasiswa dengan jenis kelamin perempuan, maka nama yang diinput akan diprint.

- b. Menampilkan hasil penjumlahan deret bilangan 25 sampai dengan 1!

Jawab:

Source codenya

```
Pertemuan 7 > J Individu2_12.java > Individu2_12 > main(String[])
1 public class Individu2_12 {
    Run | Debug
2     public static void main(String[] args) {
3         int i = 25;
4         int total = 0;
5         do {
6             total += i;
7             i--;
8         } while (i >= 0);
9         System.out.println("Hasil penjumlahan deret bilangan " + total);
10    }
11
12 }
```

Hasilnya:

```
Hasil penjumlahan deret bilangan 325
PS D:\Sekolah\Dokumen\KULIAH\Semester 1\DASPRO>
```

Pada awalnya akan di inisialisasi variabel i dengan nilai 25. Kemudian ada variabel total yang didapat dengan variaumlahkan variabel total itu sendiri dengan variabel i. Setelah itu variabel i dilakukan proses decrement. Proses ini akan terjadi perulangan selama nilai lebih dari atau sama dengan dari 0.

- c. Menampilkan deret bilangan 1 sampai 50, kecuali bilangan kelipatan 3 (1 2 4 5 7 8 10 ... 47 49 50)

Jawab:

Source codenya:

```
J Individu3_12.java > Individu3_12 > main(String[])
1 public class Individu3_12 {
    Run | Debug
2     public static void main(String[] args) {
3         int i = 0;
4         do {
5             if (i % 3 != 0) {
6                 System.out.println("Hasil kelipatan 3 dimulai dari 1: " + i + " ");
7             }
8             i++;
9         } while (i <= 50);
10
11    }
12 }
```


Hasilnya:

```
Hasil kelipatan 3 dimulai dari 1: 13
Hasil kelipatan 3 dimulai dari 1: 14
Hasil kelipatan 3 dimulai dari 1: 16
Hasil kelipatan 3 dimulai dari 1: 17
Hasil kelipatan 3 dimulai dari 1: 19
Hasil kelipatan 3 dimulai dari 1: 20
Hasil kelipatan 3 dimulai dari 1: 22
Hasil kelipatan 3 dimulai dari 1: 23
Hasil kelipatan 3 dimulai dari 1: 25
Hasil kelipatan 3 dimulai dari 1: 26
Hasil kelipatan 3 dimulai dari 1: 28
Hasil kelipatan 3 dimulai dari 1: 29
Hasil kelipatan 3 dimulai dari 1: 31
Hasil kelipatan 3 dimulai dari 1: 32
Hasil kelipatan 3 dimulai dari 1: 34
Hasil kelipatan 3 dimulai dari 1: 35
Hasil kelipatan 3 dimulai dari 1: 37
Hasil kelipatan 3 dimulai dari 1: 38
Hasil kelipatan 3 dimulai dari 1: 40
Hasil kelipatan 3 dimulai dari 1: 41
Hasil kelipatan 3 dimulai dari 1: 43
Hasil kelipatan 3 dimulai dari 1: 44
Hasil kelipatan 3 dimulai dari 1: 46
Hasil kelipatan 3 dimulai dari 1: 47
Hasil kelipatan 3 dimulai dari 1: 49
Hasil kelipatan 3 dimulai dari 1: 50
```

Jadi output yang ditampilkan adalah semua bilangan dari 1 hingga 50 kecuali kelipatan 3 seperti 3, 6, 9, 12, 15, dll. Jadi tahapannya begini:

- Inisialisasi variabel *i* bernilai 0.
- Menggunakan program do-while. Di dalamnya terdapat logika pemilihan if. Ini berfungsi untuk menentukan bilangan yang tidak diprint di antara 1 hingga 50, yaitu kelipatan 3.
- Jadi jika nilai *i* apabila dibagi 3 sisa baginya tidak sama dengan 0, maka program akan mengeprint nilai dari variabel *i*. Jika iya, maka tidak akan diprint.
- Kemudian dilakukan increment pada variabel *i*.
- Setelah itu program akan diulang dari pengecekan kondisi nilai variabel *i* sisa baginya 3 tidak nol.
- Program ini akan berhenti ketika nilai dari *i* sudah mencapai 50

Tugas Kelompok

1. Fitur login dengan menggunakan while

```
19 while (true) {
20     System.out.println(x:"Masukkan nomor rekening :");
21     String input_no_rek = scanner.nextLine();
22
23     System.out.println(x:"Masukkan pin anda :");
24     String input_pin = scanner.nextLine();
25
26     System.out.println(x:"*****");
27
28     if (status.equals(anObject:"diblokir")) {
29         System.out.println("Akun anda berstatus " + status);
30         System.out.println(x:"*****");
31     }
32
33     if (input_no_rek.equals(no_rek) && input_pin.equals(pin) && status.equals(anObject:"aman")) {
34         String no_rek_tujuan;
35         int nom_transfer, nom_tarik, nom_setor, saldo_awal = 5000000, sisa_saldo;
36         System.out.println(x:"Anda berhasil login, silahkan memilih menu dibawah ini :");
37         System.out.println(x:"1. Transfer");
38         System.out.println(x:"2. Tarik tunai");
39         System.out.println(x:"3. Setor tunai");
40         System.out.println(x:"4. Pembayaran lain-lain");
41         System.out.println(x:"Menu yang dipilih :");
42         int menu = scanner.nextInt();
43
44         System.out.println(x:"*****");
45
46         switch (menu) {
47             case 1:
48                 Scanner scanner2 = new Scanner(System.in);
49                 System.out.println(x:"Anda memilih menu transfer");
50                 System.out.println(x:"*****");
51                 System.out.println(x:"Masukkan nomor rekening tujuan :");
52                 no_rek_tujuan = scanner2.nextLine();
53                 System.out.println(x:"Masukkan nominal transfer : ");
54                 nom_transfer = scanner.nextInt();
55                 if (nom_transfer > saldo_awal) {
56                     System.out.println(x:"Transaksi gagal, periksa kembali saldo anda");
57                 } else {
58                     sisa_saldo = saldo_awal - nom_transfer;
59                     System.out.println("Transfer ke nomor " + no_rek_tujuan + " berhasil dilakukan");
60                     System.out.println("Sisa saldo anda : " + sisa_saldo);
61                 }
62                 break;
63             case 2:
64                 System.out.println(x:"Anda memilih menu tarik tunai");
65                 System.out.println(x:"*****");
66                 System.out.println(x:"Masukkan nominal tarik tunai : ");
67                 nom_tarik = scanner.nextInt();
68                 if (nom_tarik > saldo_awal) {
69                     System.out.println(x:"Transaksi gagal, periksa kembali saldo anda");
70                 } else {
71                     sisa_saldo = saldo_awal - nom_tarik;
72                     System.out.println(x:"Tarik tunai berhasil dilakukan");
73                     System.out.println("Sisa saldo anda : " + sisa_saldo);
74                 }
75                 break;
76             case 3:
77                 System.out.println(x:"Anda memilih menu setor tunai");
```

```

76         case 3:
77             System.out.println(x:"Anda memilih menu setor tunai");
78             System.out.println(x:"Masukkan nominal setor tunai : ");
79             nom_setor = scanner.nextInt();
80             sisa_saldo = saldo_awal + nom_setor;
81             System.out.println(x:"Setor tunai berhasil dilakukan");
82             System.out.println("Sisa saldo anda : " + sisa_saldo);
83             break;
84         case 4:
85             System.out.println(x:"Anda memilih menu pembayaran lain lain");
86             break;
87         default:
88             System.out.println(x:"Periksa kembali inputan anda");
89             break;
90     }
91     break; // Successful login, exit the loop
92 } else {
93     System.out.println(x:"Gagal login, periksa kembali detail anda");
94     loginAttempts++;
95
96     if (loginAttempts >= maxLoginAttempts) {
97         System.out.println(x:"Anda telah gagal lebih dari 3 kali. Akun anda diblokir.");
98         status = "diblokir";
99         // break; // Account is blocked, exit the loop
100     }
101 }
102 System.out.println(x:"*****");
103 }
104
105 scanner.close();
106 }
107 }

```

Penggunaan looping while dilakukan untuk mengulangi perintah input PIN dan nomor rekening. Sesuai pada gambar di atas, terdapat logika if di akhir while. Di sana terdapat ketentuan yang menyatakan apabila user salah input lebih dari 3 kali. Jadi, apabila user menginput data tapi tidak sesuai dengan database sebanyak 3 kali, maka user akan diberitahu bahwa nomor rekeningnya sedang di blokir.

2. Fitur tarik tunai

```

switch (menu) {
    case 1:
        do {
            Scanner scanner2 = new Scanner(System.in);
            System.out.println("Anda memilih menu transfer");
            System.out.println("*****");
            System.out.println("Masukkan nomor rekening tujuan :");
            no_rek_tujuan = scanner2.nextLine();
            System.out.println("Masukkan nominal transfer : ");
            nom_transfer = scanner.nextInt();
            if (nom_transfer > saldo_awal) {
                System.out.println("Transaksi gagal, periksa kembali saldo anda");
            } else {
                sisa_saldo = saldo_awal - nom_transfer;
                System.out.println("Transfer ke nomor " + no_rek_tujuan + " berhasil dilakukan");
                System.out.println("Sisa saldo anda : " + sisa_saldo);
            }
            System.out.println("Apakah anda ingin mengulangi transaksi?");
            konfirmasi = scanner2.nextLine();
            if (konfirmasi.equalsIgnoreCase("t")) {
                System.out.println("Terimakasih telah bertransaksi");
                break;
            }
        } while (konfirmasi.equalsIgnoreCase("y"));
        break;
}

```

Perulangan ini terletak di dalam switch case. Apabila pengguna memilih case yang ke 1, maka sistem looping ini akan dimulai.

- Tahap awal: Pengguna perlu melakukan input nomor rekening tujuan dan nominal transfer
- Apabila nominal transfer lebih besar dari saldo awal, maka sistem akan menginformasikan bahwa transaksi yang dilakukan gagal dan program akan selesai
- Apabila kondisi sebaliknya, maka akan dilakukan operasi aritmatika sederhana yaitu pengurangan saldo awal dengan nominal transfer dan menyatakan bahwa transaksi berhasil dilakukan.
- Kemudian sistem akan memberi tahu apakah user ingin mengulang transaksi. Jika “Y” maka program akan diulang dari input nominal transfer dan nomor rekening tujuan. Jika “T”, maka program akan berhenti.