

Nama : Deanissa Sherly Sabilla

Kelas : 1B SIB

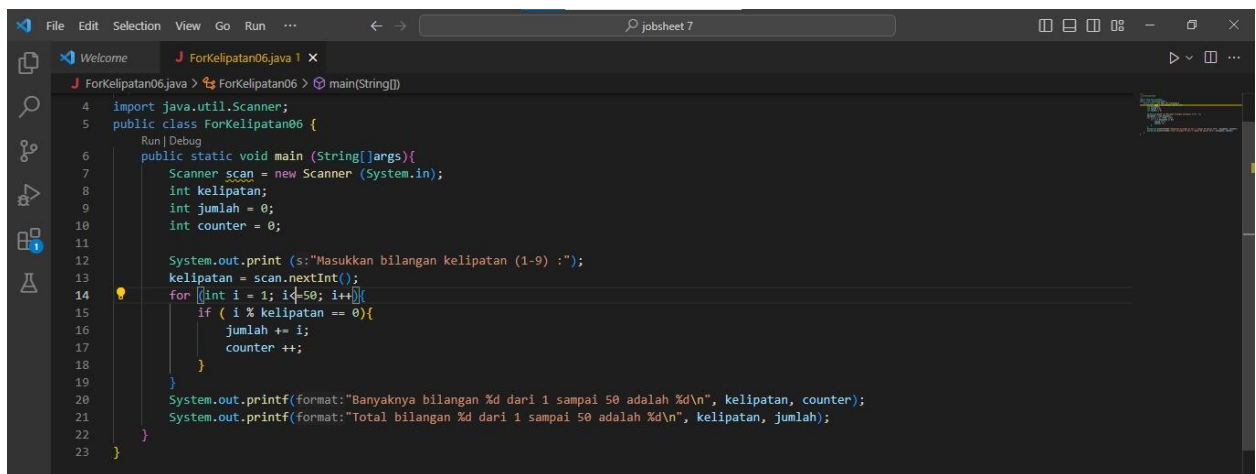
JOBSHEET 7

PERULANGAN 1

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

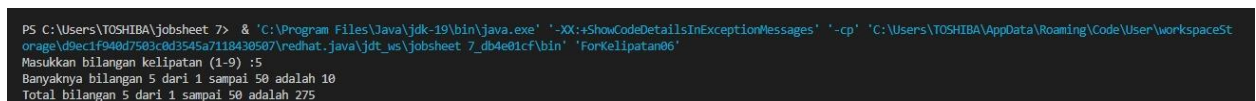
Pada percobaan ini dilakukan pembuatan kode program untuk menampilkan bilangan kelipatan angka tertentu dari rentang 1 sampai dengan 50 menggunakan perulangan FOR, serta menghitung total dari bilangan-bilangan tersebut.

➤ INPUT :



```
File Edit Selection View Go Run ... jobsheet 7
Welcome
J ForKelipatan06.java x
J ForKelipatan06.java > ForKelipatan06 > main(String[])
4 import java.util.Scanner;
5 public class ForKelipatan06 {
6     public static void main (String[] args){
7         Scanner scan = new Scanner (System.in);
8         int kelipatan;
9         int jumlah = 0;
10        int counter = 0;
11
12        System.out.print (s:"Masukkan bilangan kelipatan (1-9) :");
13        kelipatan = scan.nextInt();
14        for (int i = 1; i<=50; i++){
15            if ( i % kelipatan == 0){
16                jumlah += i;
17                counter ++;
18            }
19        }
20        System.out.printf(format:"Banyaknya bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);
21        System.out.printf(format:"Total bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, jumlah);
22    }
23 }
```

➤ OUTPUT :



```
PS C:\Users\TOSHIBA\jobsheet 7> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\TOSHIBA\AppData\Roaming\Code\User\workspaceStorage\99ec1f940d7593c8d3545a7118439507\redhat.java\jdk_ws\jobsheet 7_db4e01cf\bin' 'ForKelipatan06'
Masukkan bilangan kelipatan (1-9) :5
Banyaknya bilangan 5 dari 1 sampai 50 adalah 10
Total bilangan 5 dari 1 sampai 50 adalah 275
```

➤ 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!
 - inisialisasi: deklarasi dan inisialisasi variabel counter (variabel pengontrol perulangan)
int i = 1;
 - kondisi: batas atau syarat agar perulangan tetap dieksekusi
i<=50;
 - update: perubahan nilai variabel counter pada setiap putaran perulangan (increment atau decrement)
i++

2. Jelaskan alur kerja dari potongan kode program berikut!

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

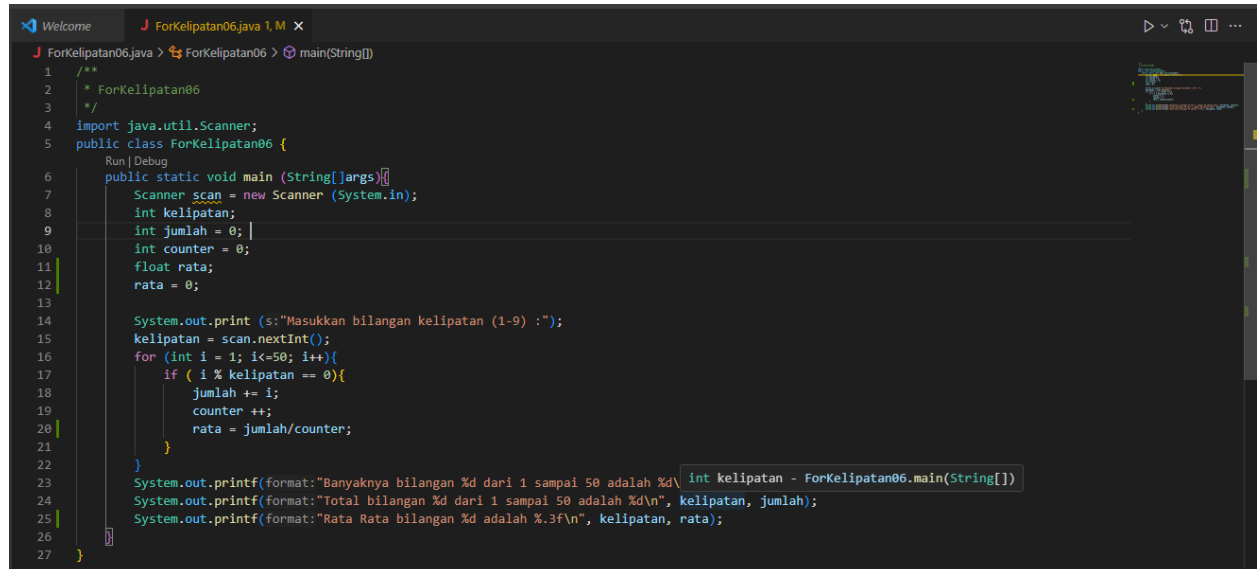
- Berikut merupakan alur kerja program :

1. `for (int i = 1; i <= 50; i++)`: akan berjalan dari `i = 1`, hingga `i` kurang dari atau sama dengan 50`. Ini berarti program akan melakukan perulangan 50 kali.
2. `if (i % kelipatan == 0)`: pernyataan `if` memeriksa apakah i` kelipatan dari suatu nilai yang disimpan dalam variabel kelipatan`. Jika benar (artinya i` adalah kelipatan), akan dieksekusi.`
3. `jumlah += i;` Ini akan menambahkan nilai `i` ke dalam variabel jumlah`.`
4. `counter++`: Ini akan meningkatkan nilai variabel setiap kali kita menemukan sebuah kelipatan.

Dengan kata lain, program ini menghitung jumlah dari semua angka dalam 1 hingga 50 yang merupakan kelipatan dari `kelipatan`, dan juga menghitung berapa banyak angka yang memenuhi kondisi tersebut. Hasil akhirnya adalah jumlah total dari angka-angka tersebut yang disimpan dalam variabel jumlah`.`

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.

- INPUT :



```
1  /**  
2   * ForKelipatan06  
3   */  
4  import java.util.Scanner;  
5  public class ForKelipatan06 {  
6      public static void main (String[] args) {  
7          Scanner scan = new Scanner (System.in);  
8          int kelipatan;  
9          int jumlah = 0;  
10         int counter = 0;  
11         float rata;  
12         rata = 0;  
13  
14         System.out.print (s:"Masukkan bilangan kelipatan (1-9) :");  
15         kelipatan = scan.nextInt();  
16         for (int i = 1; i<=50; i++){  
17             if ( i % kelipatan == 0){  
18                 jumlah += i;  
19                 counter ++;  
20                 rata = jumlah/counter;  
21             }  
22         }  
23         System.out.printf(format:"Banyaknya bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan - ForKelipatan06.main(String[])  
24         System.out.printf(format:"Total bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, jumlah);  
25         System.out.printf(format:"Rata Rata bilangan %d adalah %.3f\n", kelipatan, rata);  
26     }  
27 }
```

➤ **OUTPUT :**

```
PS C:\Users\TOSHIBA\jobsheet 7> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\TOSHIBA\jobsheet\src\main\java\com\example\calculator\ForKelipatan06'
Masukkan bilangan kelipatan (1-9) : 5
Banyaknya bilangan 5 dari 1 sampai 50 adalah 10
Total bilangan 5 dari 1 sampai 50 adalah 275
Rata Rata bilangan 5 adalah 27.000
PS C:\Users\TOSHIBA\jobsheet 7> █
```

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

- **INPUT :**

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Users\TOSHIBA\jobsheet> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\TOSHIBA\AppData\Roaming\Code\User\workspaceStorage\d9ec1f948d7503c0d3545a7118430507\redhat.java\jdt_ws\jobsheet_db_7d4e01cf\bin' 'whileKelipatan06'
Masukkan bilangan kelipatan (1-9) :3
Banyaknya bilangan 3 dari 1 sampai 50 adalah 16
Total bilangan 3 dari 1 sampai 50 adalah 408
PS C:\Users\TOSHIBA\jobsheet> █
```

- **OUTPUT :**

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS C:\Users\TOSHIBA\jobsheet > & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\TOSHIBA\AppDataRoaming\Code\User\workspaceStorage\d9ec1f940d7583c0d3545a7118438507\redhat.java\jdt_ws\jobsheet_7_db4e01cf\bin' 'WhileKelipatan06'
```

Masukkan bilangan kelipatan (1-9) : 3

Banyaknya bilangan 3 dari 1 sampai 50 adalah 16

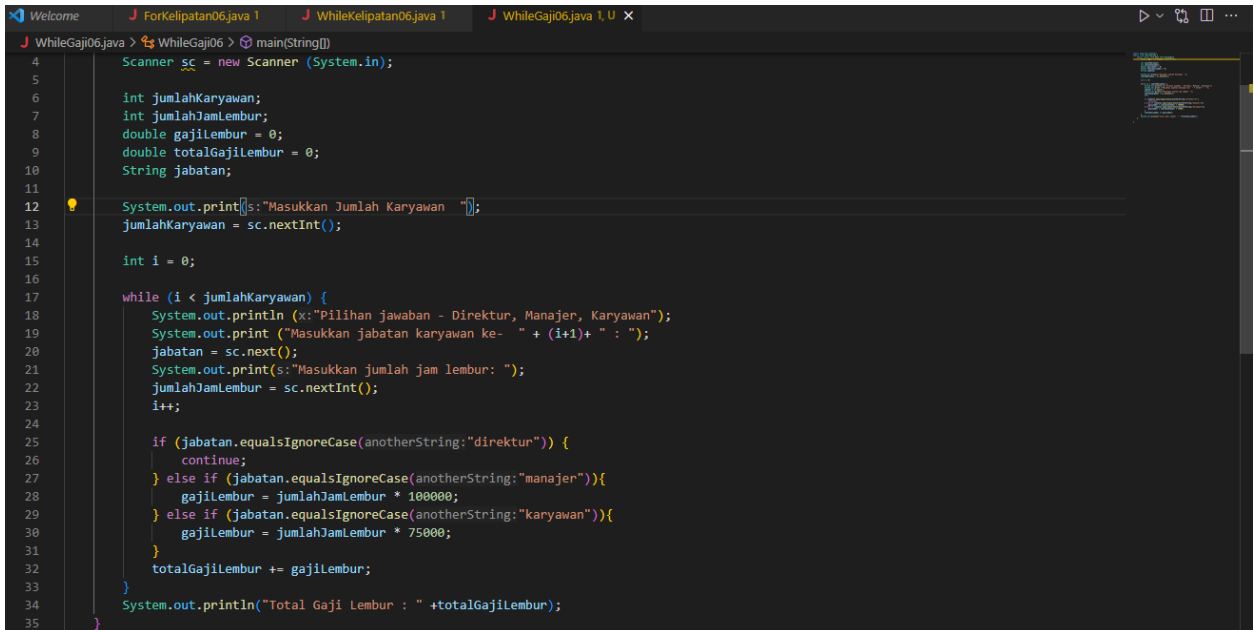
Total bilangan 3 dari 1 sampai 50 adalah 408

```
PS C:\Users\TOSHIBA\jobsheet > █
```

- B. Percobaan 2** Menghitung Gaji Lembur Karyawan Menggunakan WHILE dan CONTINUE.

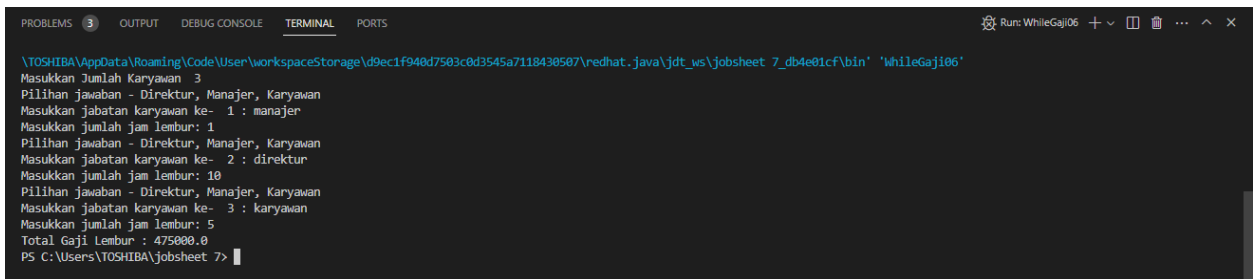
Sebuah perusahaan memberikan gaji lembur kepada karyawannya setiap minggu. Gaji tersebut dihitung berdasarkan jabatan karyawan dan jumlah jam lembur dalam seminggu. Karyawan dengan jabatan “direktur” tidak mendapatkan tambahan gaji meskipun melakukan lembur, karyawan dengan jabatan “manager” mendapatkan 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.

➤ INPUT :



```
4 Scanner sc = new Scanner (System.in);
5
6 int jumlahKaryawan;
7 int jumlahJamLembur;
8 double gajiLembur = 0;
9 double totalGajiLembur = 0;
10 String jabatan;
11
12 System.out.print(s:"Masukkan Jumlah Karyawan : ");
13 jumlahKaryawan = sc.nextInt();
14
15 int i = 0;
16
17 while (i < jumlahKaryawan) {
18     System.out.println (x:"Pilihan jawaban - Direktur, Manajer, Karyawan");
19     System.out.print ("Masukkan jabatan karyawan ke- " + (i+1)+ " : ");
20     jabatan = sc.next();
21     System.out.print(s:"Masukkan jumlah jam lembur: ");
22     jumlahJamLembur = sc.nextInt();
23     i++;
24
25     if (jabatan.equalsIgnoreCase(anotherString:"direktur")) {
26         continue;
27     } else if (jabatan.equalsIgnoreCase(anotherString:"manajer")){
28         gajiLembur = jumlahJamLembur * 100000;
29     } else if (jabatan.equalsIgnoreCase(anotherString:"karyawan")){
30         gajiLembur = jumlahJamLembur * 75000;
31     }
32     totalGajiLembur += gajiLembur;
33 }
34 System.out.println("Total Gaji Lembur : " +totalGajiLembur);
35 }
```

➤ OUTPUT :



```
Run: WhileGaji06
\\TOSHIBA\\AppData\\Roaming\\Code\\User\\workspaceStorage\\d9ec1f940d7503c0d3545a7118430507\\redhat.java\\jdt_ws\\jobsheet_7_db4e01cf\\bin' 'WhileGaji06'
Masukkan Jumlah Karyawan 3
Pilihan jawaban - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke- 1 : manajer
Masukkan jumlah jam lembur: 1
Pilihan jawaban - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke- 2 : direktur
Masukkan jumlah jam lembur: 10
Pilihan jawaban - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke- 3 : karyawan
Masukkan jumlah jam lembur: 5
Total Gaji Lembur : 475000.0
PS C:\\Users\\TOSHIBA\\jobsheet 7> |
```

➤ PERTANYAAN :

1. Tunjukkan bagian kode program yang digunakan sebagai syarat untuk menghentikan perulangan WHILE! Berapa kali perulangan dilakukan?
 - **while (i < jumlahKaryawan) {**
Pada kode program diatas, perulangan while akan terus berjalan selama jumlah karyawan sesuai dengan jumlah yang di input. Jadi, kondisi jumlahKaryawan **i <** adalah syarat untuk terjadinya perulangan.

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?

- Jika jabatan berisi DIREKTUR maka hasilnya akan sama, karena **equalsIgnoreCase** digunakan dalam pemrograman untuk membandingkan dua string tanpa memperhatikan perbedaan huruf besar dan kecil.

- Peran CONTINUE yaitu digunakan untuk menghentikan perulangan saat setelah menginput **jabatan**, kemudian melanjutkan perulangan **jabatan** 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!
- Jika i++ dipindahkan dibagian akhir, hasilnya tetap bisa eksekusi tetapi perulangan tersebut tidak bisa dilanjutkan ke perulangan selanjutnya.

```
Masukkan Jumlah Karyawan 3
Pilihan jawaban - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke- 1 : direktur
Masukkan jumlah jam lembur: 2
Pilihan jawaban - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke- 1 : manajer
Masukkan jumlah jam lembur: 1
Pilihan jawaban - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke- 1 : manajer
Masukkan jumlah jam lembur: 1
Pilihan jawaban - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke- 1 :
```

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

- **INPUT :**

```
System.out.print(s:"Masukkan Jumlah Karyawan ");
jumlahKaryawan = sc.nextInt();

int i = 0;

while (i < jumlahKaryawan) {
    System.out.println (x:"Pilihan jawaban - Direktur, Manajer, Karyawan");
    System.out.print ("Masukkan jabatan karyawan ke- " + (i+1)+ " : ");
    jabatan = sc.next();
    System.out.print(s:"Masukkan jumlah jam lembur: ");
    jumlahJamLembur = sc.nextInt();
    i++;
    if (jabatan.equalsIgnoreCase(anotherString:"direktur")) {
        continue;
    } else if (jabatan.equalsIgnoreCase(anotherString:"manajer")){
        gajiLembur = jumlahJamLembur * 100000;
    } else if (jabatan.equalsIgnoreCase(anotherString:"karyawan")){
        gajiLembur = jumlahJamLembur * 75000;
    } else {
        i-=1;
        System.out.println(x:"Jabatan Invalid");
        continue;
    } totalGajiLembur += gajiLembur;
    System.out.println("Total Gaji Lembur : " +totalGajiLembur);
}
```

Disini menambahkan **else** untuk mengandle jabatan invalid :

```
} else {
    i-=1;
    System.out.println("Jabatan Invalid");
    continue;
}
```

➤ **OUTPUT :**

```
PS C:\Users\TOSHIBA\jobsheet 7> c:; cd 'c:\Users\TOSHIBA\jobsheet 7'; & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\TOSHIBA\AppData\Roaming\Code\User\workspaceStorage\d9ec1f940d7503c0d3545a7118430507\redhat.java\jdt_ws\jobsheet_7_db4e01cf\bin' 'WhileGaji06'
```

```
Masukkan Jumlah Karyawan 3
Pilihan jawaban - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke- 1 : direktur
Masukkan jumlah jam lembur: 1
Pilihan jawaban - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke- 2 : karyawan
Masukkan jumlah jam lembur: 1
Total Gaji Lembur : 75000.0
Pilihan jawaban - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke- 3 : pegawai
Masukkan jumlah jam lembur: 2
Jabatan Invalid
Pilihan jawaban - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke- 3 : karyawan
Masukkan jumlah jam lembur: 2
Total Gaji Lembur : 225000.0
PS C:\Users\TOSHIBA\jobsheet 7> |
```

C. Percobaan 3 Menghitung Jatah Cuti Menggunakan DO-WHILE

Pada percobaan ini dilakukan pembuatan kode program menggunakan DO-WHILE untuk menghitung jatah cuti yang dimiliki oleh pegawai. Pegawai mempunyai jatah cuti sebanyak 5 hari. Jatah cuti akan dikurangi perhati setiap kali digunakan. Saat jatah cuti sisa 2 hari, pegawai mendapat peringatan untuk berhenti menggunakan jatah cutinya.

➤ **INPUT :**

```
J DoWhileCuti06.java > DoWhileCuti06 > main(String[])
```

```
1 import java.util.Scanner;
2 public class DoWhileCuti06 {
3     public static void main (String[] args){
4         Scanner sc = new Scanner (System.in);
5
6         int jatahCuti;
7         int jumlahHari;
8         String konfirmasi;
9
10        System.out.print(s:"Jatah Cuti : ");
11        jatahCuti = sc.nextInt();
12
13        do {
14            System.out.print (s:"Apakah Anda ingin mengambil cuti (y/n)");
15            konfirmasi = sc.next();
16
17            if (konfirmasi.equalsIgnoreCase(anotherString:"y")) {
18                System.out.print(s:"Jumlah hari :");
19                jumlahHari = sc.nextInt();
20
21                if (jumlahHari <= jatahCuti){
22                    jatahCuti -= jumlahHari;
23                    System.out.println ("Sisa jatah cuti : " + jatahCuti);
24                } else {
25                    System.out.println(x:"Sisa jatah cuti anda tidak mencukupi");
26                    break;
27                }
28            }
29        } while (jatahCuti >0);
30    }
31 }
```

➤ **OUTPUT :**

```
Data\Roaming\Code\User\workspaceStorage\d9ec1f940d7503c0d3545a7118430507\redhat.java\jdt_ws\jobsheet_7_db4e01cf\bin' 'DowhileCuti06'
Jatah Cuti : 12
Apakah Anda ingin mengambil cuti (y/n)y
Jumlah hari :4
Sisa jatah cuti : 8
Apakah Anda ingin mengambil cuti (y/n)y
Jumlah hari :5
Sisa jatah cuti : 3
Apakah Anda ingin mengambil cuti (y/n)y
Jumlah hari :4
Sisa jatah cuti anda tidak mencukupi
PS C:\Users\TOSHIBA\jobsheet 7> █
```

➤ **PERTANYAAN :**

1. Apa kegunaan sintaks BREAK di dalam sintaks perulangan?
 - Kegunaan BREAK yaitu untuk menghentikan paksa perulangan, kemudian kode di luar perulangan akan dieksekusi
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.

○ **OUTPUT :**

```
Sisa jatah cuti : 10
Apakah Anda ingin mengambil cuti (y/n)y
Jumlah hari :6
Sisa jatah cuti : 4
Apakah Anda ingin mengambil cuti (y/n)y
Jumlah hari :5
Sisa jatah cuti anda tidak mencukupi
Apakah Anda ingin mengambil cuti (y/n)█
```

Disini saya, menghapus dibagian **BREAK**, sehingga bisa menjalankan program tanpa berhenti meskipun perulangan tidak sesuai

3. Pada saat input konfirmasi, ketikkan "t", apa yang terjadi? Mengapa demikian?
 - Yang terjadi program tidak akan melanjutkan ke eksekusi selanjutnya, karena tidak ada konfirmasi atau perulangan untuk mengarahkan ke "t"
4. Modifikasi kode program sehingga saat pengguna mengetikkan "t" sebagai input konfirmasi, maka program akan berhenti

➤ **INPUT :**

```
26         }
27     } else
28     {
29         System.out.println(x:"Tidak dapat mencukupi");
30         break;
31     } while (jatahCuti >0);
32 }
```

Disini menambahkan **else** untuk inputan bagian "t"

➤ **OUTPUT :**

```
PS C:\Users\TOSHIBA\jobsheet 7> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessage' '-cp' 'C:\Users\TOSHIBA\AppData\Roaming\Code\User\workspaceStorage\d9ec1f940d7503c0d3545a7118430507\redhat.java\jdt_ws\jobsheet_7_db4e01cf\bin' 'DowhileCuti06'
Jatah Cuti : 2
Apakah Anda ingin mengambil cuti (y/t)t
Tidak dapat mencukupi
```

Hasil Push dan Commit di GitHub :

Commits

🔗 master ▾

Commits on Oct 12, 2023

Modifikasi dengan menambah t	0b833fd <>
deanissa committed 46 minutes ago	
Modifikasi dengan menghapus break	9f2ec41 <>
deanissa committed 1 hour ago	
Do While Cuti	d384df3 <>
deanissa committed 1 hour ago	
While Gaji	811f8ea <>
deanissa committed 2 hours ago	
Pindah For ke While kelipatan	f1b14fc <>
deanissa committed 3 hours ago	
For Kelipatan modifikasi ke rata-rata	0f8d27e <>
deanissa committed 3 hours ago	
For kelipatan sebelum dimodifikasi	c7ba9b9 <>
deanissa committed 4 hours ago	

Now available from the navigation bar for immediate access everywhere.

Okay, got it!

Link : [deanissa/jobsheet-7 \(github.com\)](https://github.com/deanissa/jobsheet-7)