

LAPORAN PRAKTIKUM
MATA KULIAH PRAKTIKUM DASAR PEMROGRAMAN

Dosen Pengampu : Triana Fatmawati, S.T, M.T

PERTEMUAN 7 : PERULANGAN 1



Nama : Yonanda Mayla Rusdiaty

NIM : 2341760184

Prodi : D-IV Sistem Informasi Bisnis

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2023

2.1 Percobaan 1 : Menghitung Bilangan Kelipatan Menggunakan FOR

Terlebih dahulu membuat file di text editor dengan nama **ForKelipatan29.java**. Kemudian tuliskan struktur dasar Bahasa pemrograman java yang terdiri dari fungsi main. Tambahkan library Scanner dan buat deklarasi Scanner dengan nama **scan**. ++

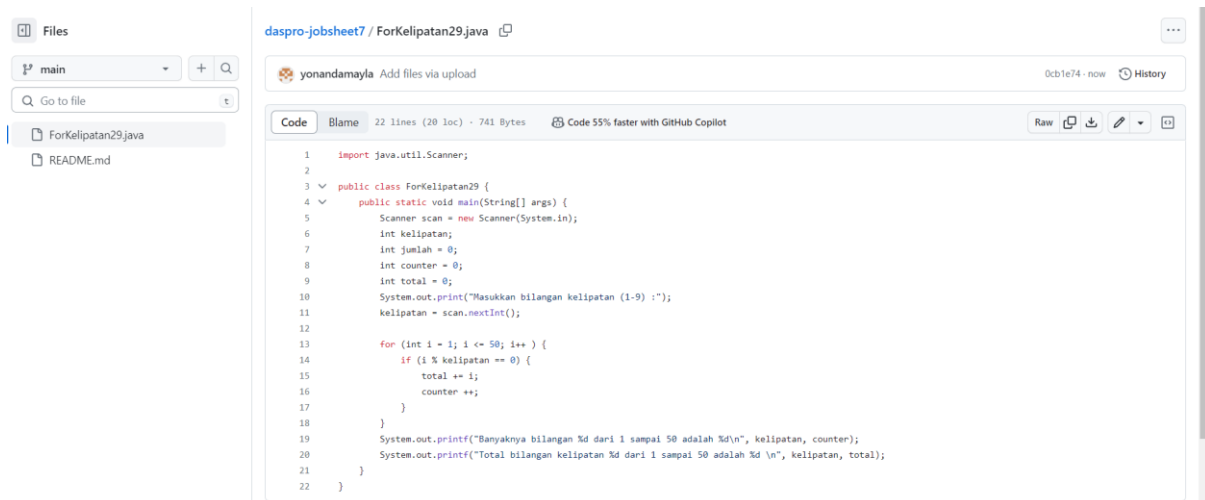
Kode program :

```
1  import java.util.Scanner;
2
3  public class ForKelipatan29 {
4      public static void main(String[] args) {
5          Scanner scan = new Scanner(System.in);
6          int kelipatan;
7          int jumlah = 0;
8          int counter = 0;
9          int total = 0;
10         System.out.print("Masukkan bilangan kelipatan (1-9) :");
11         kelipatan = scan.nextInt();
12
13         for (int i = 1; i <= 50; i++) {
14             if (i % kelipatan == 0) {
15                 total += i;
16                 counter ++;
17             }
18         }
19         System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);
20         System.out.printf("Total bilangan kelipatan %d dari 1 sampai 50 adalah %d \n", kelipatan, total);
21     }
22 }
```

Output :

```
Masukkan bilangan kelipatan (1-9) :5
Banyaknya bilangan 5 dari 1 sampai 50 adalah 10
Total bilangan kelipatan 5 dari 1 sampai 50 adalah 275
PS D:\DASPRO\daspro-jobsheet7> █
```

Commit dan push kode program ke Github :



```
1 import java.util.Scanner;
2
3 public class Forkelipatan29 {
4     public static void main(String[] args) {
5         Scanner scan = new Scanner(System.in);
6         int kelipatan;
7         int jumlah = 0;
8         int counter = 0;
9         int total = 0;
10        System.out.print("Masukkan bilangan kelipatan (1-9) :");
11        kelipatan = scan.nextInt();
12
13        for (int i = 1; i <= 50; i++) {
14            if (i % kelipatan == 0) {
15                total += i;
16                counter ++;
17            }
18        }
19        System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);
20        System.out.printf("Total bilangan kelipatan %d dari 1 sampai 50 adalah %d\n", kelipatan, total);
21    }
22 }
```

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!
2. Jelaskan alur kerja dari potongan kode program berikut!

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

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

Jawaban

1. Terdapat 3 komponen perulangan pada sintaks FOR pada kode program tersebut. Berikut adalah masing-masing komponen perulangan FOR tersebut :

- a. **Inisialisasi** : yaitu terletak pada bagian pertama dari perulangan FOR.

Komponen inisialisasi ini ditandai dengan :

```
for (int i = 1;
```

- b. **Kondisi** : yaitu terletak pada bagian tengah dari perulangan FOR. Komponen kondisi ini ditandai dengan :

```
i <= 50;
```

- c. **Peningkatan/update** : yaitu terletak pada bagian terakhir dari perulangan FOR. Komponen peningkatan ini ditandai dengan :

```
i++ ) {
```

2. Alur kerja dari potongan kode program berikut adalah :

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

- 1) Pertama **int i = 1** adalah inisialisasi dari variable i dengan nilai awal 1. Perulangan akan dimulai dari 1. **i <= 50** menunjukkan kondisi yang harus dipenuhi agar perulangan dapat terus berjalan. **i++** adalah ekspresi peningkatan yang akan menambahkan 1 ke nilai I setiap kali perulangan dilakukan.
- 2) Selanjutnya dalam kondisi if, apakah I merupakan kelipatan dari **kelipatan** yang akan diinputkan user. Jika kondisi ini bernilai true, maka pernyataan if akan dieksekusi
- 3) Kemudian **total +i** digunakan untuk menambahkan nilai I ke dalam variable total untuk menghitung total dari bilangan kelipatan yang ditemukan. **counter++** digunakan untuk menghitung berapa banyak bilangan kelipatan yang telah ditemukan.

3. Memodifikasi kode program yang telah dibuat dengan menambahkan variabel baru untuk menghitung rata-rata dari seluruh bilangan kelipatan yang ditentukan.

Kode Program :

```
1  import java.util.Scanner;
2
3  public class ForKelipatan29_1 {
4      public static void main(String[] args) {
5          Scanner scan = new Scanner(System.in);
6          int kelipatan;
7          int jumlah = 0;
8          int counter = 0;
9          int total = 0;
10         double rataRata = 0.0;
11
12         System.out.print("Masukkan bilangan kelipatan (1-9) :");
13         kelipatan = scan.nextInt();
14
15         for (int i = 1; i <= 50; i++ ) {
16             if (i % kelipatan == 0) {
17                 total += i;
18                 counter ++;
19             }
20         }
21
22         rataRata = (double) total / counter;
23
24         System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);
25         System.out.printf("Total bilangan kelipatan %d dari 1 sampai 50 adalah %d\n", kelipatan, total);
26         System.out.printf("Rata-rata kelipatan %d dari 1 sampai 50 adalah %.2f\n", kelipatan, rataRata);
27     }
28 }
```

Output :

```
Masukkan bilangan kelipatan (1-9) :5
Banyaknya bilangan 5 dari 1 sampai 50 adalah 10
Total bilangan kelipatan 5 dari 1 sampai 50 adalah 275
Rata-rata kelipatan 5 dari 1 sampai 50 adalah 27.50
```

Push dan commit hasil modifikasi ke repository Github :

ForkKelipatan29.java
ForkKelipatan29_1.java
README.md

```
Code Blame 26 lines (22 loc) · 913 Bytes Code 55% faster with GitHub Copilot
1 import java.util.Scanner;
2
3 public class ForkKelipatan29_1 {
4     public static void main(String[] args) {
5         Scanner scan = new Scanner(System.in);
6         int kelipatan;
7         int jumlah = 0;
8         int counter = 0;
9         int total = 0;
10        System.out.print("Masukkan bilangan kelipatan (1-9) :");
11        kelipatan = scan.nextInt();
12
13        for (int i = 1; i <= 50; i++) {
14            if (i % kelipatan == 0) {
15                total += i;
16                counter ++;
17            }
18        }
19
20        double rataRata = (double) total / counter;
21
22        System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);
23        System.out.printf("Total bilangan kelipatan %d dari 1 sampai 50 adalah %d\n", kelipatan, total);
24        System.out.printf("Rata-rata kelipatan %d dari 1 sampai 50 adalah %.2f\n", kelipatan, rataRata);
25    }
26 }
```

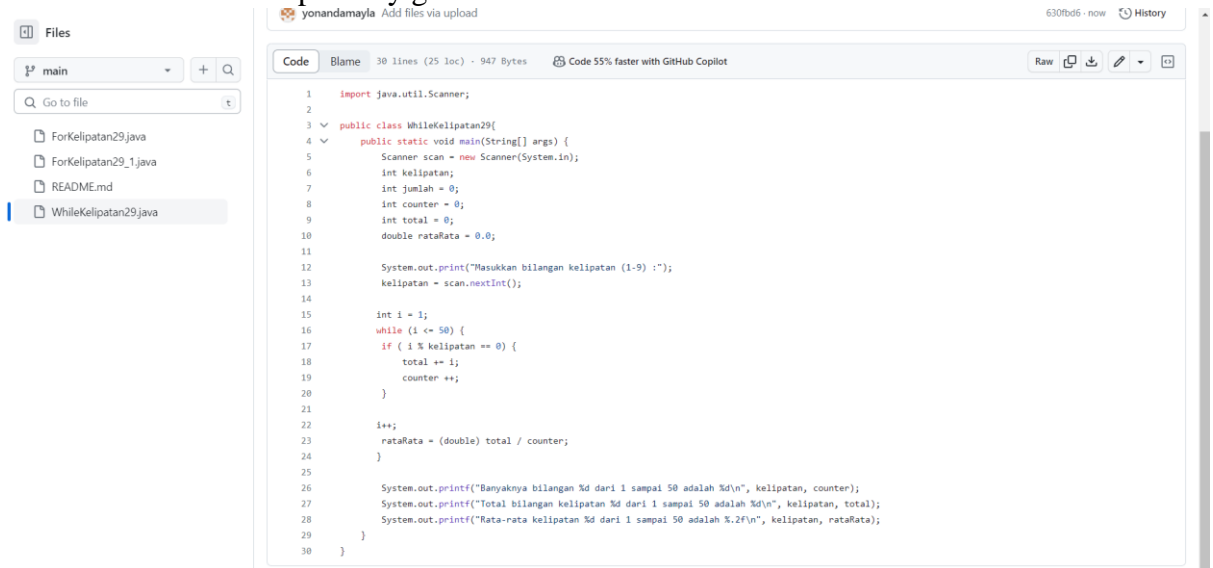
4. Buat file baru dengan nama WhileKelipatan29.java. Buatlah kode program dengan tujuan serupa tetapi menggunakan WHILE.
- Code program :

```
1 import java.util.Scanner;
2
3 public class WhileKelipatan29{
4     public static void main(String[] args) {
5         Scanner scan = new Scanner(System.in);
6         int kelipatan;
7         int jumlah = 0;
8         int counter = 0;
9         int total = 0;
10        double rataRata = 0.0;
11
12        System.out.print("Masukkan bilangan kelipatan (1-9) :");
13        kelipatan = scan.nextInt();
14
15        int i = 1;
16        while (i <= 50) {
17            if (i % kelipatan == 0) {
18                total += i;
19                counter ++;
20            }
21
22            i++;
23            rataRata = (double) total / counter;
24        }
25
26        System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);
27        System.out.printf("Total bilangan kelipatan %d dari 1 sampai 50 adalah %d\n", kelipatan, total);
28        System.out.printf("Rata-rata kelipatan %d dari 1 sampai 50 adalah %.2f\n", kelipatan, rataRata);
29    }
30 }
```

Output :

```
Masukkan bilangan kelipatan (1-9) :5
Banyaknya bilangan 5 dari 1 sampai 50 adalah 10
Total bilangan kelipatan 5 dari 1 sampai 50 adalah 275
Rata-rata kelipatan 5 dari 1 sampai 50 adalah 27.50
PS D:\DASPRO\daspro-jobsheet7> █
```

Push dan commit ke repository github :



The screenshot displays a GitHub repository interface for a user named 'yonandamayla'. The left sidebar shows the file structure with 'main' selected, listing files: 'ForKelipatan29.java', 'ForKelipatan29_1.java', 'README.md', and 'WhileKelipatan29.java'. The main area shows the code for 'WhileKelipatan29.java', which is 30 lines long (25 loc) and 947 bytes. The code is as follows:

```
1  import java.util.Scanner;
2
3  public class WhileKelipatan29{
4      public static void main(String[] args) {
5          Scanner scan = new Scanner(System.in);
6          int kelipatan;
7          int jumlah = 0;
8          int counter = 0;
9          int total = 0;
10         double rataRata = 0.0;
11
12         System.out.print("Masukkan bilangan kelipatan (1-9) :");
13         kelipatan = scan.nextInt();
14
15         int i = 1;
16         while (i <= 50) {
17             if (i % kelipatan == 0) {
18                 total += i;
19                 counter ++;
20             }
21
22             i++;
23             rataRata = (double) total / counter;
24         }
25
26         System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);
27         System.out.printf("Total bilangan kelipatan %d dari 1 sampai 50 adalah %d\n", kelipatan, total);
28         System.out.printf("Rata-rata kelipatan %d dari 1 sampai 50 adalah %.2f\n", kelipatan, rataRata);
29     }
30 }
```

2.2 Menghitung Gaji Lembur Karyawan Menggunakan WHILE dan CONTINUE

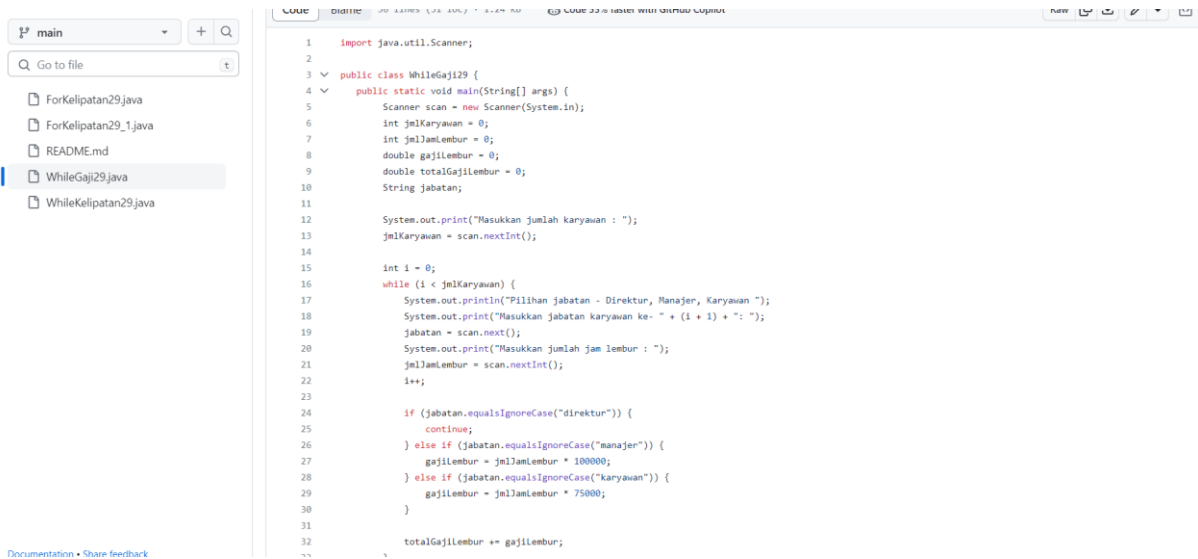
Kode program :

```
1  import java.util.Scanner;
2
3  public class WhileGaji29 {
4      public static void main(String[] args) {
5          Scanner scan = new Scanner(System.in);
6          int jmlKaryawan = 0;
7          int jmlJamLembur = 0;
8          double gajiLembur = 0;
9          double totalGajiLembur = 0;
10         String jabatan;
11
12         System.out.print("Masukkan jumlah karyawan : ");
13         jmlKaryawan = scan.nextInt();
14
15         int i = 0;
16         while (i < jmlKaryawan) {
17             System.out.println("Pilihan jabatan - Direktur, Manajer, Karyawan ");
18             System.out.print("Masukkan jabatan karyawan ke- " + (i + 1) + ": ");
19             jabatan = scan.next();
20             System.out.print("Masukkan jumlah jam lembur : ");
21             jmlJamLembur = scan.nextInt();
22             i++;
23
24             if (jabatan.equalsIgnoreCase("direktur")) {
25                 continue;
26             } else if (jabatan.equalsIgnoreCase("manajer")) {
27                 gajiLembur = jmlJamLembur * 100000;
28             } else if (jabatan.equalsIgnoreCase("karyawan")) {
29                 gajiLembur = jmlJamLembur * 75000;
30             }
31
32             totalGajiLembur += gajiLembur;
33         }
34         System.out.println("Total gaji lembur: " + totalGajiLembur);
35     }
36 }
```


Output ;

```
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:\DASPRO\daspro-jobsheet7>
```

Push dan commit ke repository github :



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'main' with files: 'ForKelipatan29.java', 'ForKelipatan29_1.java', 'README.md', 'WhileGaji29.java' (selected), and 'WhileKelipatan29.java'. The code editor shows the following Java code:

```
1  import java.util.Scanner;
2
3  public class WhileGaji29 {
4      public static void main(String[] args) {
5          Scanner scan = new Scanner(System.in);
6          int jmlKaryawan = 0;
7          int jmlJamLembur = 0;
8          double gajiLembur = 0;
9          double totalGajiLembur = 0;
10         String jabatan;
11
12         System.out.print("Masukkan jumlah karyawan : ");
13         jmlKaryawan = scan.nextInt();
14
15         int i = 0;
16         while (i < jmlKaryawan) {
17             System.out.println("Pilihan jabatan - Direktur, Manajer, Karyawan ");
18             System.out.print("Masukkan jabatan karyawan ke- " + (i + 1) + ": ");
19             jabatan = scan.next();
20             System.out.print("Masukkan jumlah jam lembur : ");
21             jmlJamLembur = scan.nextInt();
22             i++;
23
24             if (jabatan.equalsIgnoreCase("direktur")) {
25                 continue;
26             } else if (jabatan.equalsIgnoreCase("manajer")) {
27                 gajiLembur = jmlJamLembur * 100000;
28             } else if (jabatan.equalsIgnoreCase("karyawan")) {
29                 gajiLembur = jmlJamLembur * 75000;
30             }
31
32             totalGajiLembur += gajiLembur;
33         }
34     }
35 }
```

Pertanyaan

1. Tunjukkan bagian kode program yang digunakan sebagai syarat untuk menghentikan perulangan WHILE! Berapa kali perulangan dilakukan?
2. Pada potongan kode berikut, Apa yang sebenarnya terjadi jika variabel jabatan berisi nilai “DIREKTUR”? Apa peran CONTINUE yang dituliskan di dalam sintaks perulangan?
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!
4. Modifikasi kode program untuk handle jabatan yang invalid seperti contoh berikut:

Jawaban :

1. Kode program yang digunakan untuk menghentikan perulangan while adalah pernyataan **continue**.

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

Pernyataan continue ini digunakan untuk mengarahkan eksekusi ke proses berikutnya pada loop yang sama. Pernyataan continue menyebabkan eksekusi menuju kondisi pengujian pengulangan, sehingga program ini akan mengulangi memasukkan input dan baru berakhir setelah input yang dimasukkan negative/ di luar batas. Pengulangan dilakukan sebanyak < jumlah karyawan, dimana dalam studi kasus ini jumlah karyawannya 3. Berarti, perulangan akan dilakukan sebanyak 3 kali.

2. Yang terjadi jika variabel jabatan berisi nilai DIREKTUR adalah program akan terus berjalan dikarenakan ada perintah “equalsIgnoreCase”. Jika ada perintah tersebut maka program akan terus dilanjutkan. Sedangkan fungsi dari CONTINUE di dalam sintaks perulangan berperan untuk menghentikan perulangan dari sintaks yang sudah dilakukan di awal awal.
3. Karena dalam perulangan while i++ diletakkan di bagian tengah adalah untuk proses increment di proses setelah proses i+1 di bagian memasukkan jabatan karyawan yang

fungsinya sama dengan count +1 lalu i++ akan dibaca sehingga terjadi perulangan yang berpengaruh ke penghitungan jabatan karyawan ke berapa. Lalu, yang terjadi jika kita memindah i++ ke bagian akhir dan juga memasukkan “direktur” sebagai jabatan karyawan pertama adalah pengurutan nomer akan menjadi berantakan dan juga perulangan menjadi tidak teratur.

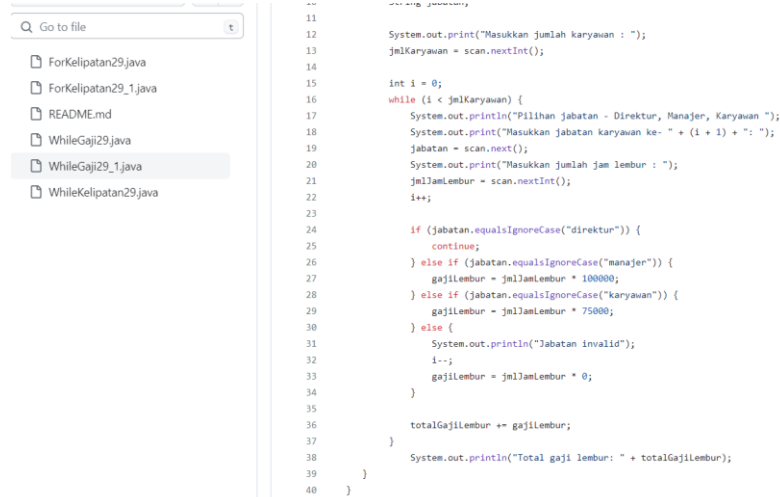
4. Memodifikasi kode program untuk handle jabatan yang invalid seperti contoh Kode program :

```
1  import java.util.Scanner;
2
3  public class WhileGaji29_1 {
4      public static void main(String[] args) {
5          Scanner scan = new Scanner(System.in);
6          int jmlKaryawan = 0;
7          int jmlJamLembur = 0;
8          double gajiLembur = 0;
9          double totalGajiLembur = 0;
10         String jabatan;
11
12         System.out.print("Masukkan jumlah karyawan : ");
13         jmlKaryawan = scan.nextInt();
14
15         int i = 0;
16         while (i < jmlKaryawan) {
17             System.out.println("Pilihan jabatan - Direktur, Manajer, Karyawan ");
18             System.out.print("Masukkan jabatan karyawan ke- " + (i + 1) + ": ");
19             jabatan = scan.next();
20             System.out.print("Masukkan jumlah jam lembur : ");
21             jmlJamLembur = scan.nextInt();
22             i++;
23
24             if (jabatan.equalsIgnoreCase("direktur")) {
25                 continue;
26             } else if (jabatan.equalsIgnoreCase("manajer")) {
27                 gajiLembur = jmlJamLembur * 100000;
28             } else if (jabatan.equalsIgnoreCase("karyawan")) {
29                 gajiLembur = jmlJamLembur * 75000;
30             } else {
31                 System.out.println("Jabatan invalid");
32                 i--;
33                 gajiLembur = jmlJamLembur * 0;
34             }
35
36             totalGajiLembur += gajiLembur;
37         }
38         System.out.println("Total gaji lembur: " + totalGajiLembur);
39     }
40 }
```

Output :

```
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: 130000.0
```

Push dan commit ke repository github ;



```
11
12     System.out.print("Masukkan jumlah karyawan : ");
13     jmlKaryawan = scan.nextInt();
14
15     int i = 0;
16     while (i < jmlKaryawan) {
17         System.out.println("Pilihan jabatan - Direktur, Manajer, Karyawan ");
18         System.out.print("Masukkan jabatan karyawan ke- " + (i + 1) + ": ");
19         jabatan = scan.next();
20         System.out.print("Masukkan jumlah jam lembur : ");
21         jmlJamLembur = scan.nextInt();
22         i++;
23
24         if (jabatan.equalsIgnoreCase("direktur")) {
25             continue;
26         } else if (jabatan.equalsIgnoreCase("manajer")) {
27             gajiLembur = jmlJamLembur * 100000;
28         } else if (jabatan.equalsIgnoreCase("karyawan")) {
29             gajiLembur = jmlJamLembur * 75000;
30         } else {
31             System.out.println("Jabatan invalid");
32             i--;
33             gajiLembur = jmlJamLembur * 0;
34         }
35         totalGajiLembur += gajiLembur;
36     }
37     System.out.println("Total gaji lembur: " + totalGajiLembur);
38 }
39 }
40 }
```

2.3 Menghitung Jatah Cuti Menggunakan Do-While

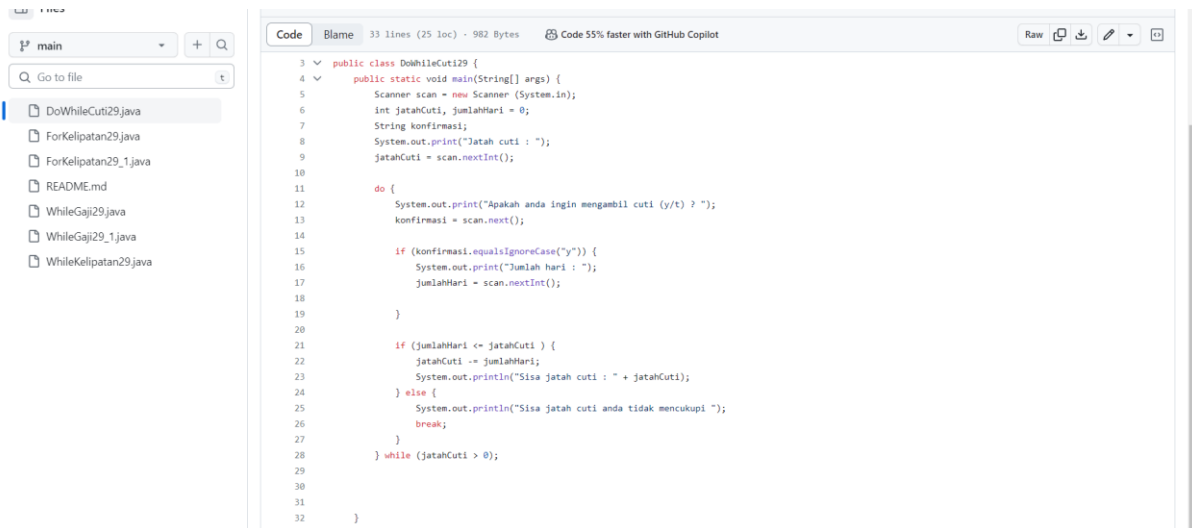
Kode program :

```
1  import java.util.Scanner;
2
3  public class DoWhileCuti29 {
4      public static void main(String[] args) {
5          Scanner scan = new Scanner (System.in);
6          int jatahCuti, jumlahHari = 0;
7          String konfirmasi;
8          System.out.print("Jatah cuti : ");
9          jatahCuti = scan.nextInt();
10
11         do {
12             System.out.print("Apakah anda ingin mengambil cuti (y/t) ? ");
13             konfirmasi = scan.next();
14
15             if (konfirmasi.equalsIgnoreCase("y")) {
16                 System.out.print("Jumlah hari : ");
17                 jumlahHari = scan.nextInt();
18
19             }
20
21             if (jumlahHari <= jatahCuti ) {
22                 jatahCuti -= jumlahHari;
23                 System.out.println("Sisa jatah cuti : " + jatahCuti);
24             } else {
25                 System.out.println("Sisa jatah cuti anda tidak mencukupi ");
26                 break;
27             }
28         } while (jatahCuti > 0);
29
30
31     }
32 }
33 }
```

Output :

```
Jatah cuti : 12
Apakah anda ingin mengambil cuti (y/t) ? y
Jumlah hari : 4
Sisa jatah cuti : 8
Apakah anda ingin mengambil cuti (y/t) ? y
Jumlah hari : 5
Sisa jatah cuti : 3
Apakah anda ingin mengambil cuti (y/t) ? y
Jumlah hari : 4
Sisa jatah cuti anda tidak mencukupi
PS D:\DASPRO\daspro-jobsheet7> █
```

Push dan commit ke repository github :



```
3 public class DoWhileCuti29 {
4     public static void main(String[] args) {
5         Scanner scan = new Scanner (System.in);
6         int jatahCuti, jumlahHari = 0;
7         String konfirmasi;
8         System.out.print("Jatah cuti : ");
9         jatahCuti = scan.nextInt();
10
11         do {
12             System.out.print("Apakah anda ingin mengambil cuti (y/t) ? ");
13             konfirmasi = scan.next();
14
15             if (konfirmasi.equalsIgnoreCase("y")) {
16                 System.out.print("Jumlah hari : ");
17                 jumlahHari = scan.nextInt();
18             }
19
20             if (jumlahHari <= jatahCuti) {
21                 jatahCuti -= jumlahHari;
22                 System.out.println("Sisa jatah cuti : " + jatahCuti);
23             } else {
24                 System.out.println("Sisa jatah cuti anda tidak mencukupi ");
25                 break;
26             }
27         } while (jatahCuti > 0);
28
29
30
31
32 }
```

Pertanyaan

1. Apa kegunaan sintaks BREAK di dalam sintaks 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 mengisi 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?
5. Modifikasi kode program sehingga saat pengguna mengetikkan “t” sebagai input konfirmasi, maka program akan berhenti.
6. Push dan commit kode program ke github.

Jawaban :

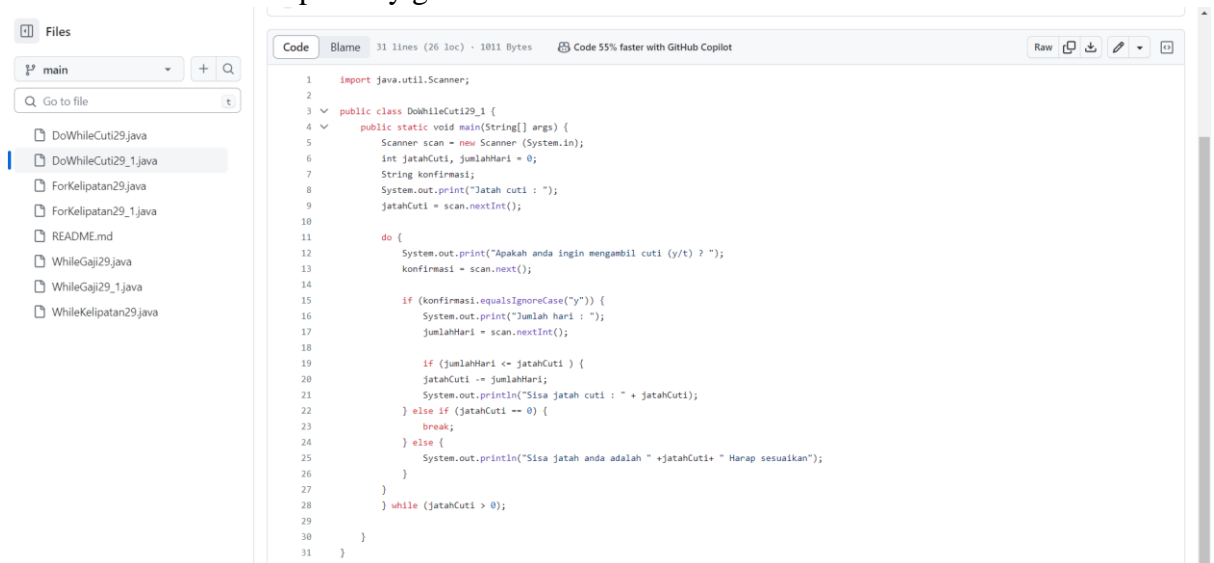
1. Sintaks break dalam sintaks perulangan digunakan untuk memaksa sebuah perulangan berhenti sebelum waktunya. Pada umumnya sintaks break digunakan untuk mengakhiri sebuah eksekusi dalam statement.
2. Kode program :

```
1 import java.util.Scanner;
2
3 public class DoWhileCuti29_1 {
4     public static void main(String[] args) {
5         Scanner scan = new Scanner (System.in);
6         int jatahCuti, jumlahHari = 0;
7         String konfirmasi;
8         System.out.print("Jatah cuti : ");
9         jatahCuti = scan.nextInt();
10
11         do {
12             System.out.print("Apakah anda ingin mengambil cuti (y/t) ? ");
13             konfirmasi = scan.next();
14
15             if (konfirmasi.equalsIgnoreCase("y")) {
16                 System.out.print("Jumlah hari : ");
17                 jumlahHari = scan.nextInt();
18
19                 if (jumlahHari <= jatahCuti) {
20                     jatahCuti -= jumlahHari;
21                     System.out.println("Sisa jatah cuti : " + jatahCuti);
22                 } else if (jatahCuti == 0) {
23                     break;
24                 } else {
25                     System.out.println("Sisa jatah anda adalah " + jatahCuti + " Harap sesuaikan");
26                 }
27             }
28         } while (jatahCuti > 0);
29
30     }
31 }
```

Output :

```
Jatah cuti : 12
Apakah anda ingin mengambil cuti (y/t) ? y
Jumlah hari : 9
Sisa jatah cuti : 3
Apakah anda ingin mengambil cuti (y/t) ? y
Jumlah hari : 4
Sisa jatah anda adalah 3 Harap sesuaikan
Apakah anda ingin mengambil cuti (y/t) ? y
Jumlah hari : 3
Sisa jatah cuti : 0
PS D:\DASPRO\daspro-jobsheet7>
```

3. Push dan commit ke repository github :



```
1  import java.util.Scanner;
2
3  public class DoWhileCuti29_1 {
4      public static void main(String[] args) {
5          Scanner scan = new Scanner(System.in);
6          int jatahCuti, jumlahHari = 0;
7          String konfirmasi;
8          System.out.print("Jatah cuti : ");
9          jatahCuti = scan.nextInt();
10
11         do {
12             System.out.print("Apakah anda ingin mengambil cuti (y/t) ? ");
13             konfirmasi = scan.next();
14
15             if (konfirmasi.equalsIgnoreCase("y")) {
16                 System.out.print("Jumlah hari : ");
17                 jumlahHari = scan.nextInt();
18
19                 if (jumlahHari <= jatahCuti) {
20                     jatahCuti -= jumlahHari;
21                     System.out.println("Sisa jatah cuti : " + jatahCuti);
22                 } else if (jatahCuti == 0) {
23                     break;
24                 } else {
25                     System.out.println("Sisa jatah anda adalah " + jatahCuti + " Harap sesuaikan");
26                 }
27             }
28             while (jatahCuti > 0);
29
30         }
31     }
```

4. Yang terjadi ketika kita mengetikkan “t” saat input konfirmasi adalah program tidak akan berhenti dan akan terus berjalan karena belum ada deklarasi untuk inputan t


```
Jatah cuti : 12
Apakah anda ingin mengambil cuti (y/t) ? t
Apakah anda ingin mengambil cuti (y/t) ? t
Apakah anda ingin mengambil cuti (y/t) ? t
Apakah anda ingin mengambil cuti (y/t) ?
```

5. Kode program :

```
1  import java.util.Scanner;
2
3  public class DoWhileCuti29_2 {
4      public static void main(String[] args) {
5          Scanner scan = new Scanner (System.in);
6          int jatahCuti, jumlahHari = 0;
7          String konfirmasi;
8          System.out.print("Jatah cuti : ");
9          jatahCuti = scan.nextInt();
10
11      do {
12          System.out.print("Apakah anda ingin mengambil cuti (y/t) ? ");
13          konfirmasi = scan.next();
14
15          if (konfirmasi.equalsIgnoreCase("y")) {
16              System.out.print("Jumlah hari : ");
17              jumlahHari = scan.nextInt();
18
19              if (jumlahHari <= jatahCuti ) {
20                  jatahCuti -= jumlahHari;
21                  System.out.println("Sisa jatah cuti : " + jatahCuti);
22
23              } else if (jatahCuti == 0) {
24                  break;
25              } else {
26                  System.out.println("Sisa jatah anda adalah " +jatahCuti+ " Harap sesuaikan");
27              }
28
29              } else {
30                  break;
31              }
32      } while (jatahCuti > 0);
33
34
35  }
36
37 }
38
```

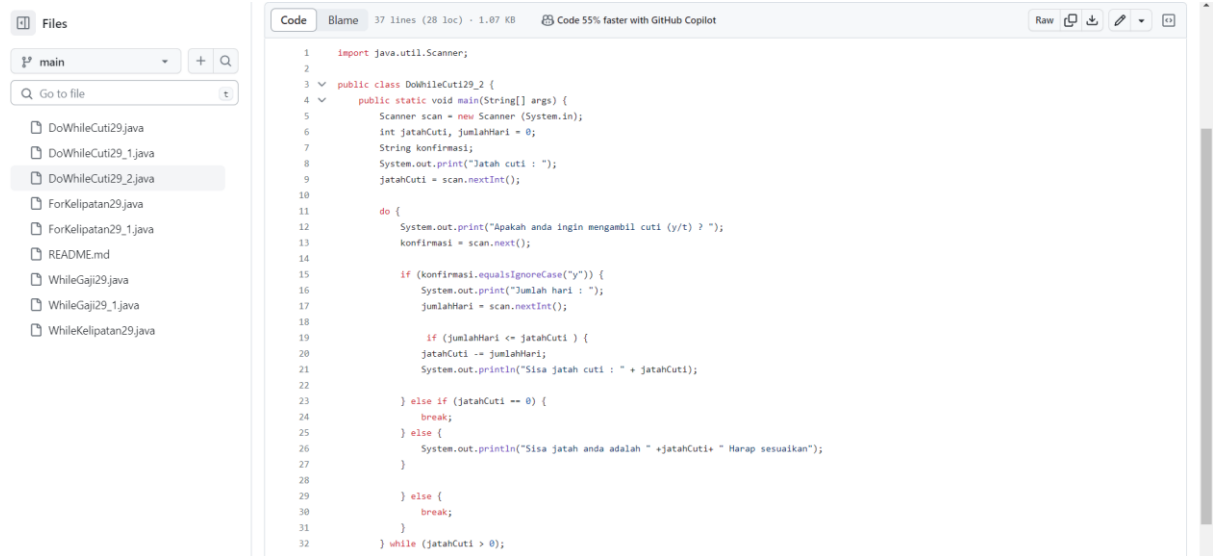
Output :

```

Jatah cuti : 12
Apakah anda ingin mengambil cuti (y/t) ? y
Jumlah hari : 9
Sisa jatah cuti : 3
Apakah anda ingin mengambil cuti (y/t) ? y
Jumlah hari : 5
Sisa jatah anda adalah 3 Harap sesuaikan
Apakah anda ingin mengambil cuti (y/t) ? y
Jumlah hari : 3
Sisa jatah cuti : 0
PS D:\DASPRO\daspro-jobsheet7>

```

6. Push dan commit kode program ke repository github :



The screenshot shows a GitHub repository interface. On the left, a file explorer shows a list of files: `main`, `DoWhileCuti29.java`, `DoWhileCuti29_1.java`, `DoWhileCuti29_2.java` (selected), `ForkKelipatan29.java`, `ForkKelipatan29_1.java`, `README.md`, `WhileGaji29.java`, `WhileGaji29_1.java`, and `WhileKelipatan29.java`. The main area displays the code for `DoWhileCuti29_2.java`, which is a Java program using a `do-while` loop to calculate remaining vacation days based on user input.

```

1  import java.util.Scanner;
2
3  public class DoWhileCuti29_2 {
4      public static void main(String[] args) {
5          Scanner scan = new Scanner(System.in);
6          int jatahCuti, jumlahHari = 0;
7          String konfirmasi;
8          System.out.print("Jatah cuti : ");
9          jatahCuti = scan.nextInt();
10
11         do {
12             System.out.print("Apakah anda ingin mengambil cuti (y/t) ? ");
13             konfirmasi = scan.next();
14
15             if (konfirmasi.equalsIgnoreCase("y")) {
16                 System.out.print("Jumlah hari : ");
17                 jumlahHari = scan.nextInt();
18
19                 if (jumlahHari <= jatahCuti) {
20                     jatahCuti -= jumlahHari;
21                     System.out.println("Sisa jatah cuti : " + jatahCuti);
22
23                 } else if (jatahCuti == 0) {
24                     break;
25                 } else {
26                     System.out.println("Sisa jatah anda adalah " + jatahCuti + " Harap sesuaikan");
27                 }
28
29                 } else {
30                     break;
31                 }
32         } while (jatahCuti > 0);
33     }
34 }

```