

JOBSHEET 7
LAPORAN PRAKTIKUM DASAR PEMROGRAMAN



Nama: Riovaldo Alfian Fahmi Rahman

NIM: 2341720209

Kelas: 1B

Prodi: D4 Teknik Informatika

2.1 Menghitung Bilangan Kelipatan Menggunakan FOR

1. Buka code editor anda
2. Kemudian buat file baru dan beri nama ForKelipatan25.java
3. Jangan lupa tambahkan library Scanner di bagian atas class
4. Jika sudah tulis code seperti gambar dibawah ini

```
ForKelipatan25.java > ForKelipatan25 > main(String[])
1  import java.util.Scanner;
2
3  public class ForKelipatan25 {
4      public static void main(String[] args) {
5          Scanner input25 = new Scanner(System.in);
6          int kelipatan, total = 0, counter = 0;
7          System.out.println("Masukkan bilangan kelipatan (1-9) : ");
8          kelipatan = input25.nextInt();
9
10
11          for (int i = 1; i <= 50; i++) {
12              if (i % kelipatan == 0) {
13                  total += i;
14                  counter++;
15              }
16          }
17
18          System.out.printf(format:"Banyaknya bilangan kelipatan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);
19          System.out.printf(format:"Total bilangan kelipatan %d dari 1 sampai 50 adalah %d\n", kelipatan, total);
20      }
21  }
22
```

5. Jika sudah silahkan run dan lihat hasilnya seperti dibawah ini

```
Rio@LAPTOP-2HE8V9VN MINGW64 /d/Kuliah/Praktek Daspro/Semester 1/Meet 7 (main|REBASE)
$ java forkelipatan25.java
Masukkan bilangan kelipatan (1-9) :
5
Banyaknya bilangan kelipatan 5 dari 1 sampai 50 adalah 10
Total bilangan kelipatan 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!

Jawab :

- **Inisialisasi** : Komponen inisialisasi itu terletak pada bagian pertama dari perulangan for, yaitu **int i = 1**; Pada perulangan itu dimulai, variabel i diinisialisasi dengan nilai 1. Inisialisasi ini hanya dilakukan sekali pada awal perulangan.
 - **Kondisi** : Komponen kondisi terletak pada bagian kedua dari perulangan for, yaitu **i <= 50**; Ini adalah kondisi yang menentukan apakah perulangan akan terus berjalan atau berhenti. Selama nilai i kurang dari atau sama dengan 50, perulangan akan terus berlanjut. Jika kondisi ini tidak lagi terpenuhi, perulangan akan berhenti.
 - **Iterasi** : Komponen iterasi terletak pada bagian ketiga dari perulangan for, yaitu **i++**. Pada setiap iterasi perulangan, nilai variabel i akan bertambah satu. Ini adalah langkah yang menjalankan kode perulangan dengan nilai i yang berubah pada setiap iterasi.
2. Jelaskan alur kerja dari potongan kode program berikut!

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

Jawab :

- Jadi variabel i diinisialisasi dengan nilai 1: **int i = 1**; Ini adalah langkah awal perulangan, dan i akan digunakan sebagai penghitung bilangan dari 1 hingga 50.
- Kondisi perulangan: **i <= 50**; Ini adalah kondisi yang menentukan berapa kali perulangan akan berjalan. Selama i kurang dari atau sama dengan 50, perulangan akan terus berlanjut.
- Setiap kali perulangan berjalan, kemudian periksa apakah i merupakan kelipatan dari nilai yang dimasukkan oleh pengguna : **if (i % kelipatan == 0)**. Operasi modulo yang digunakan di sini untuk memeriksa apakah i dapat dibagi habis oleh kelipatan.
- Jika i adalah kelipatan dari kelipatan, maka nilai i akan ditambahkan ke variabel total dan variabel counter akan diinkrementasi. Ini dilakukan untuk menghitung total bilangan kelipatan dan jumlah bilangan kelipatan yang ditemukan selama perulangan.
- Perulangan akan berlanjut, dan nilai i akan ditingkatkan dengan **i++** pada setiap iterasi.
- Perulangan akan terus berjalan hingga i melebihi 50. Setelah kondisi i <= 50 itu tidak terpenuhi, perulangan akan berhenti.

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 :

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

<https://github.com/ckckckcz/kuliah/tree/main/Praktek%20Daspro/Semester%201/Metode%207>

4. Buatlah file baru dengan nama WhileKelipatan25.java. Buatlah kode program dengan tujuan serupa tetapi menggunakan WHILE. Push dan commit kode program ke github.

Jawab :

```
WhileKelipatan25.java > WhileKelipatan25 > main(String[])  
1  import java.util.Scanner;  
2  
3  public class WhileKelipatan25 {  
4      public static void main(String[] args) {  
5          Scanner input25 = new Scanner(System.in);  
6          int kelipatan, total = 0, counter = 0;  
7          System.out.println("Masukkan bilangan kelipatan (1-9) : ");  
8          kelipatan = input25.nextInt();  
9          int i = 1;  
10         while (i <= 50) {  
11             if (i % kelipatan == 0) {  
12                 total += i;  
13                 counter++;  
14             }  
15             i++;  
16         }  
17  
18         System.out.printf(format:"Banyaknya bilangan kelipatan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);  
19         System.out.printf(format:"Total bilangan kelipatan %d dari 1 sampai 50 adalah %d\n", kelipatan, total);  
20  
21     }  
22 }
```

2.2 Percobaan 2: Menghitung Gaji Lembur Karyawan Menggunakan WHILE dan CONTINUE

1. Buka code editor anda
2. Kemudian buat file baru dan beri nama WhileGaji25.java
3. Jangan lupa tambahkan library Scanner di bagian atas class
4. Jika sudah tulis code seperti gambar dibawah ini

```
WhileGaji25.java > WhileGaji25 > main(String[])
1  import java.util.Scanner;
2  public class WhileGaji25 {
3      public static void main(String[] args) {
4          Scanner input25 = new Scanner(System.in);
5          int jumlahKaryawan, jumlahJamLembur;
6          double gajiLembur = 0, totalGajiLembur = 0;
7          String jabatan;
8          System.out.println("Masukkan jumlah karyawan : ");
9          jumlahKaryawan = input25.nextInt();
10
11         int i = 0;
12         while (i < jumlahKaryawan){
13             System.out.println("Pilihan jabatan - Direktur, Manajer, Karyawan");
14             System.out.println("Masukkan jabatan karyawan ke-" + (i+1) + " : ");
15             jabatan = input25.next();
16             System.out.println("Masukkan jumlah jam lembur :");
17             jumlahJamLembur = input25.nextInt();
18             i++;
19
20             if (jabatan.equalsIgnoreCase("direktur")){
21                 continue;
22             } else if (jabatan.equalsIgnoreCase("manajer")){
23                 gajiLembur = jumlahJamLembur * 100000;
24             }
25             else if (jabatan.equalsIgnoreCase("karyawan")){
26                 gajiLembur = jumlahJamLembur * 75000;
27             }
28
29             totalGajiLembur += gajiLembur;
30         }
31         System.out.println("Total Gaji Lembur : " + totalGajiLembur);
32     }
33 }
```

5. Jika sudah silahkan run dan lihat hasilnya seperti dibawah ini

```
Rio@LAPTOP-2HE8V9VN MINGW64 /d/Kuliah/Praktek Das
REBASE)
$ java whilegaji25.java
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
```

Pertanyaan

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

Jawab :

Jadi untuk menghentikan perulangan `while` yaitu dengan `break`, namun pada code yang di program tidak ada `break`, jadi perulangan akan terus berlanjut sampai `i` mencapai nilai yang sama dengan `jumlahKaryawan`. Dengan begitu, perulangan dilakukan sebanyak `jumlahKaryawan` kali sesuai dengan jumlah karyawan yang dimasukkan oleh pengguna.

2. Apa yang sebenarnya terjadi jika variabel jabatan berisi nilai "DIREKTUR"? Apa peran CONTINUE yang dituliskan di dalam sintaks perulangan ?

Jawab :

Jadi begini untuk peran dari `continue` dalam konteks ini adalah untuk mengabaikan sisa kode di dalam blok while dan melanjutkan iterasi berikutnya dari perulangan while. Dalam kata lain, jika seorang karyawan memiliki jabatan "`DIREKTUR`", maka perulangan akan melompati bagian kode yang menghitung gaji lembur dan lanjut ke iterasi berikutnya untuk memproses karyawan selanjutnya.

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 :

Komponen `i++` (penambahan nilai `i`) diletakkan di tengah blok while karena dalam kode tersebut, `i` digunakan untuk dua tujuan : Mengganti nomor karyawan saat menampilkan pesan "Masukkan jabatan karyawan ke- " + `(i+1)`.
Sebagai bagian dari kontrol perulangan while untuk memeriksa apakah perulangan harus dilanjutkan atau tidak (`i < jumlahKaryawan`).

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

Jawab :

```

1  import java.util.Scanner;
2
3  public class WhileGaji25_Sesudah {
4      public static void main(String[] args) {
5          Scanner input25 = new Scanner(System.in);
6          // Deklarasi variabel
7          int jumlahKaryawan, jumlahJamLembur;
8          double gajilembur = 0.0, totalGajilembur = 0.0;
9          String jabatan;
10         System.out.print(s:"Masukkan jumlah karyawan: ");
11         jumlahKaryawan = input25.nextInt();
12         int i = 0;
13
14         while (i < jumlahKaryawan) {
15             System.out.println(x:"Pilihan jabatan - Direktur, Manajer, Karyawan");
16             System.out.print("Masukkan jabatan karyawan ke-" + (i+1) + ": ");
17             jabatan = input25.next();
18             System.out.print(s:"Masukkan jumlah jam lembur: ");
19             jumlahJamLembur = input25.nextInt();
20
21             if (jabatan.equalsIgnoreCase(anotherString:"Direktur")) {
22                 i++;
23                 continue;
24             } else if (jabatan.equalsIgnoreCase(anotherString:"Manajer")) {
25                 gajilembur = jumlahJamLembur * 100000;
26             } else if (jabatan.equalsIgnoreCase(anotherString:"Karyawan")) {
27                 gajilembur = jumlahJamLembur * 75000;
28             } else {
29                 System.out.println(x:"Jabatan invalid");
30                 continue;
31             }
32             totalGajilembur += gajilembur;
33             i++;
34         }
35         System.out.println("Total Gaji Lembur : " + totalGajilembur);
36     }
37 }
38

```

Output :

```

Rio@LAPTOP-2HE8V9VN MINGW64 /d/Kuliah/Praktek Daspr
ter 1/Meet 7 (main|REBASE)
$ java whilegaji25_sesudah.java
Masukkan jumlah karyawan: 3
Pilihan jabatan - Direktur, Manajer, Karyawan
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 : 1300000.0

```

5. Push dan commit kode program ke github

Jawab :

<https://github.com/ckckckcz/kuliah/tree/main/Praktek%20Daspro/Semester%201/Met%207>

2.3 Percobaan 3: Menghitung Jatah Cuti Menggunakan DO-WHILE

1. Buka code editor anda
2. Kemudian buat file baru dan beri nama DoWhileCuti25.java
3. Jangan lupa tambahkan library Scanner di bagian atas class
4. Jika sudah tulis code seperti gambar dibawah ini

```
DoWhileCuti25_Sebelum.java > DoWhileCuti25_Sebelum > main(String[])
1  import java.util.Scanner;
2  public class DoWhileCuti25_Sebelum {
3      public static void main(String[] args) {
4          Scanner input25 = new Scanner(System.in);
5          int jatahCuti, jumlahHari;
6          String konfirmasi;
7          System.out.print(s:"Jatah Cuti : ");
8          jatahCuti = input25.nextInt();
9          do {
10             System.out.print(s:"Apakah anda ingin mengambil cuti (y/t) ?");
11             konfirmasi = input25.next();
12
13             if(konfirmasi.equalsIgnoreCase(anotherString:"y")){
14                 System.out.print(s:"Jumlah hari :");
15                 jumlahHari = input25.nextInt();
16
17                 if (jumlahHari <= jatahCuti){
18                     jatahCuti -= jumlahHari;
19                     System.out.println("Sisa Jatah Cuti : " + jatahCuti);
20                 }else{
21                     System.out.println(x:"Sisa jatah cuti anda tidak mencukupi");
22                     break;
23                 }
24             }
25         } while (jatahCuti > 0);
26     }
27 }
```


6. Jika sudah silahkan run dan lihat hasilnya seperti dibawah ini

```
Rio@LAPTOP-2HE8V9VN MINGW64 /d/Kuliah/Praktek Daspro/
ter 1/Meet 7 (main|REBASE)
$ java dowhilecuti25_sebelum.java
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
```

Pertanyaan

1. Apa kegunaan sintaks BREAK di dalam sintaks perulangan?

Jawab :

Sintaks **break** digunakan dalam struktur perulangan untuk **menghentikan perulangan secara tiba-tiba atau keluar dari perulangan saat kondisi tertentu terpenuhi**. Dengan kata lain, break digunakan untuk mengakhiri eksekusi perulangan sebelum kondisi perulangan benar-benar selesai.

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 :

```
Rio@LAPTOP-2HE8V9VN MINGW64 /d/Kuliah/Praktek
ter 1/Meet 7 (main|REBASE)
$ java dowhilecuti25_sebelum2.java
Jatah Cuti : 5
Apakah anda ingin mengambil cuti (y/t) ?y
Jumlah hari :3
Sisa Jatah Cuti : 2
Apakah anda ingin mengambil cuti (y/t) ?y
Jumlah hari :1
Sisa Jatah Cuti : 1
Apakah anda ingin mengambil cuti (y/t) ?y
Jumlah hari :2
Sisa jatah cuti anda tidak mencukupi
Apakah anda ingin mengambil cuti (y/t) ?y
Jumlah hari :10
Sisa jatah cuti anda tidak mencukupi
Apakah anda ingin mengambil cuti (y/t) ?
```

3. Push dan commit kode program ke github

Jawab :

<https://github.com/ckckckcz/kuliah/tree/main/Praktek%20Daspro/Semester%201/Met%207>

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

Jawab :

```
Rio@LAPTOP-2HE8V9VN MINGW64 /d/Kuliah/Prakt
$ java dowhilecuti25_sebelum2.java
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) ?t
Apakah anda ingin mengambil cuti (y/t) ?t
Apakah anda ingin mengambil cuti (y/t) ?t
Apakah anda ingin mengambil cuti (y/t) ?
```

Jadi pada kondisi ini terjadi karena dalam perulangan **do-while**, program akan menjalankan blok perintah di dalamnya setidaknya sekali sebelum memeriksa kondisi while. Oleh karena itu, bahkan jika pengguna memilih “t”, **program akan tetap menjalankan iterasi saat itu**, tidak mengurangi jatah cuti, dan kemudian **akan kembali meminta konfirmasi di iterasi berikutnya**

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

Jawab :

```
if (konfirmasi.equalsIgnoreCase(anotherString:"y")) {
    System.out.print(s:"Jumlah hari :");
    jumlahHari = input25.nextInt();

    if (jumlahHari <= jatahCuti) {
        jatahCuti -= jumlahHari;
        System.out.println("Sisa Jatah Cuti : " + jatahCuti);
    } else {
        System.out.println(x:"Sisa jatah cuti anda tidak mencukupi");
    }
} else if (konfirmasi.equalsIgnoreCase(anotherString:"t")) {
    break;
}
```

Output :

```
Rio@LAPTOP-2HE8V9VN MINGW64 /d/Kuliah/Prak
$ java dowhilecuti25_sesudah2.java
Jatah Cuti : 12
Apakah anda ingin mengambil cuti (y/t) ?t

Rio@LAPTOP-2HE8V9VN MINGW64 /d/Kuliah/Prak
$
```

6. Push dan commit kode program ke github

Jawab :

<https://github.com/ckckckcz/kuliah/tree/main/Praktek%20Daspro/Semester%201/Metode%207>