

NAMA : REVANI NANDA PUTRI
KELAS : 1B
PRODI : SIB
NO : 24

JOBSHEET 7

PERCOBAAN 1 : Menghitung Bilangan Kelipatan Menggunakan FOR

1. kode

```
jobsheet7 > J ForKelipatan24.java > ForKelipatan24
1  import java.util.Scanner;
2
3  public class ForKelipatan24 {
4      Run | Debug
5      public static void main(String[] args) {
6          Scanner scan = new Scanner(System.in);
7
8          int kelipatan, jumlah = 0, counter = 0, total;
9          System.out.print(s:"Masukkan bilangan kelipatan (1-9): ");
10         kelipatan = scan.nextInt();
11         for (int i = 1; i <= 50; i++) {
12             if (i % kelipatan == 0) {
13                 jumlah += i;
14                 counter++;
15             }
16         }
17         System.out.printf(format:"Banyaknya bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);
18         System.out.printf(format:"Total bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, jumlah);
19     }
20 }
```

2. Output

```
Masukkan bilangan kelipatan (1-9): 5
Banyaknya bilangan 5 dari 1 sampai 50 adalah 10
Total bilangan 5 dari 1 sampai 50 adalah 275
PS D:\dokumenreva\kulyeah\kelas\pemas\P.DASPRO-JOBSHEET> |
```

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!

Jawaban:

1. digunakan untuk inisialisasi variabel sebagai pengontrol perulangan

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

2. digunakan untuk kondisi, batas atau syarat agar perulangan dieksekusi

```
for (int i = 1; i <= 50; i++) {
    if (i % kelipatan == 2) {
        jumlah += i;
    }
}
```

3. digunakan untuk update, perubahan nilai variable counter setiap perputaran perulangan (increment dan decrement)

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

2. Jelaskan alur kerja dari potongan kode program berikut!

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

Jawaban:

Dalam kode tersebut i akan di inisialisasi dengan nilai 1, kemudian loop for akan berjalan selama i kurang dari atau sama dengan 50, setiap program berulang i akan bertambah 1.

di dalam loop for, terdapat pemilihan if dengan kondisi jika i di modulus dengan kelipatan hasilnya 0, maka total akan ditambah dengan i dan counter bertambah 1.

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.

```
jobsheet7 > J ForKelipatan24.java > ForKelipatan24  
1  import java.util.Scanner;  
2  
3  public class ForKelipatan24 {  
4      public static void main(String[] args) {  
5          Scanner scan = new Scanner(System.in);  
6  
7          int kelipatan, jumlah = 0, counter = 0, average = 0;  
8          System.out.print(s:"Masukkan bilangan kelipatan (1-9): ");  
9          kelipatan = scan.nextInt();  
10         for (int i = 1; i <= 50; i++) {  
11             if (i % kelipatan == 0) {  
12                 jumlah += i;  
13                 counter++;  
14                 average = jumlah/counter;  
15             }  
16         }  
17         System.out.printf(format:"Banyaknya bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);  
18         System.out.printf(format:"Total bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, jumlah);  
19         System.out.printf(format:"Rata - rata bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, average);  
20     }  
21 }  
22 }
```

```
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 dari 1 sampai 50 adalah 27  
PS D:\dokumenreva\kulyeah\kelas\pemdas\P.DASPRO-JOBSHEET> █
```

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

```
jobsheet7 > J WhileKelipatan24.java > WhileKelipatan24 > main(String[])
1 import java.util.Scanner;
2
3 public class WhileKelipatan24 {
4
5     Run | Debug
6     public static void main(String[] args) {
7         Scanner scan = new Scanner(System.in);
8
9         int kelipatan, jumlah = 0, counter = 0;
10        double average = 0;
11        System.out.print(s:"Masukkan bilangan kelipatan (1-9): ");
12        kelipatan = scan.nextInt();
13
14        int i = 1;
15        while (i<=50){
16            if (i % kelipatan == 0) {
17                jumlah += i;
18                counter++;
19                average = (double) jumlah/counter;
20            }
21            i++;
22        }
23        System.out.printf(format:"Banyaknya bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);
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 dari 1 sampai 50 adalah %.2f\n", kelipatan, average);
26    }
27 }
```

```
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 dari 1 sampai 50 adalah 27.50
PS D:\dokumenreva\kulyeah\kelas\pemas\P.DASPRO-JOBSHEET> █
```

PERCOBAAN 2 : Menghitung Gaji Lembur Karyawan Menggunakan WHILE dan CONTINUE

1. kode

```
jobsheet7 > J WhileGaji24.java > WhileGaji24 > main(String[])
3 public class WhileGaji24 {
4
5     Run | Debug
6     public static void main(String[] args) {
7         Scanner scan = new Scanner(System.in);
8
9         int jumlahKaryawan, jumlahJamLembur;
10        double gajilembur = 0, totalGajilembur = 0;
11        String jabatan;
12
13        System.out.print(s:"Masukkan jumlah karyawan: ");
14        jumlahKaryawan = scan.nextInt();
15
16        int i = 0;
17        while (i < jumlahKaryawan) {
18            System.out.println(x:"Pilihlah jabatan - Direktur, manager, Karyawan");
19            System.out.print("Masukkan jabatan karyawan ke-" + (i+1) + ": ");
20            jabatan = scan.next();
21            System.out.print(s:"Masukkan jumlah jam lembur: ");
22            jumlahJamLembur = scan.nextInt();
23            i++;
24
25            if (jabatan.equalsIgnoreCase(anotherString:"direktur")) {
26                continue;
27            } else if (jabatan.equalsIgnoreCase(anotherString:"manager")){
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    }
36 }
```

2. output

```
Masukkan jumlah karyawan: 3
Pilihlah jabatan - Direktur, manager, Karyawan
Masukkan jabatan karyawan ke-1: manager
Masukkan jumlah jam lembur: 1
Pilihlah jabatan - Direktur, manager, Karyawan
Masukkan jabatan karyawan ke-2: direktur
Masukkan jumlah jam lembur: 10
Pilihlah jabatan - Direktur, manager, Karyawan
Masukkan jabatan karyawan ke-3: karyawan
Masukkan jumlah jam lembur: 5
Total gaji lembur: 475000.0
PS D:\dokumenreva\kulyeah\kelas\pemas\p.DASPRO-JOBSHEET>
```

Pertanyaan

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

Jawaban:

```
int i = 0;
while (i < jumlahKaryawan) {
    System.out.println(x:"Pilihlah jabatan - Direktur, manager, Karyawan");
    System.out.print("Masukkan jabatan karyawan ke-"+ (i+1)+ ": ");
    jabatan = scan.next();
}
```

perulangan akan dilakukan sesuai jumlahKaryawan yang di inputkan.

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?

Jawaban:

- Pemilihan akan tetap berlanjut karena dalam kode tersebut menggunakan equalsIgnoreCase yang berarti kondisi tidak akan melihat besar kecilnya huruf.
- Peran continue digunakan untuk melewati sisa statement dalam loop dan eksekusi loop berjalan ke tahap 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!

Jawaban:

- i++ diletakkan di tengah karena pada dasarnya program akan dijalankan dari atas ke bawah, jadi setelah menginputkan jabatan karyawan dan jam lembur maka variable i akan bertambah 1
- jika i++ dipindah ke bagian akhir, program akan mengalami looping yang tidak selesai dan i akan tidak beraturan

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

```
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
```

Jawaban:

```
jobsheet7 > J WhileGaji24.java > WhileGaji24 > main(String[])
1  import java.util.Scanner;
2
3  public class WhileGaji24 {
4
5      Run | Debug
6      public static void main(String[] args) {
7          Scanner scan = new Scanner(System.in);
8
9          int jumlahKaryawan, jumlahJamLembur;
10         double gajiLembur = 0, totalGajiLembur = 0;
11         String jabatan;
12
13         System.out.print(s:"Masukkan jumlah karyawan: ");
14         jumlahKaryawan = scan.nextInt();
15
16         int i = 0;
17         while (i < jumlahKaryawan) {
18             System.out.println(x:"Pilihlah jabatan - Direktur, manager, Karyawan");
19             System.out.print("Masukkan jabatan karyawan ke-" + (i + 1) + ": ");
20             jabatan = scan.next();
21             System.out.print(s:"Masukkan jumlah jam lembur: ");
22             jumlahJamLembur = scan.nextInt();
23             i++;
24
25             if (jabatan.equalsIgnoreCase(anotherString:"direktur")) {
26                 continue;
27             } else if (jabatan.equalsIgnoreCase(anotherString:"manager")) {
28                 gajiLembur = jumlahJamLembur * 100_000;
29             } else if (jabatan.equalsIgnoreCase(anotherString:"karyawan")) {
30                 gajiLembur = jumlahJamLembur * 75_000;
31             } else {
32                 i -= 1;
33                 System.out.println(x:"Jabatan invalid!");
34                 continue;
35             }
36             totalGajiLembur += gajiLembur;
37         }
38
39         System.out.println("Total gaji lembur: " + totalGajiLembur);
40     }
```

output:

```
Masukkan jumlah karyawan: 3
Pilihlah jabatan - Direktur, manager, Karyawan
Masukkan jabatan karyawan ke-1: manager
Masukkan jumlah jam lembur: 10
Pilihlah jabatan - Direktur, manager, Karyawan
Masukkan jabatan karyawan ke-2: direktur
Masukkan jumlah jam lembur: 5
Pilihlah jabatan - Direktur, manager, Karyawan
Masukkan jabatan karyawan ke-3: pegawai
Masukkan jumlah jam lembur: 4
Jabatan invalid!
Pilihlah jabatan - Direktur, manager, Karyawan
Masukkan jabatan karyawan ke-3: karyawan
Masukkan jumlah jam lembur: 4
Total gaji lembur: 1300000.0
PS D:\dokumenreva\kulyeah\kelas\pemdas\P.DASPRO-JOBSHEET>
```

PERCOBAAN 3 : Menghitung Jatah Cuti Menggunakan DO-WHILE

1. kode

```
jobsheet7 > J DoWhileCuti24.java > DoWhileCuti24 > main(String[])
1  import java.util.Scanner;
2
3  public class DoWhileCuti24 {
4
5      Run | Debug
6      public static void main(String[] args) {
7          Scanner sc = new Scanner (System.in);
8
9          int jatahCuti, jumlahHari;
10         String konfirmasi;
11
12         System.out.print(s:"Jatah cuti: ");
13         jatahCuti = sc.nextInt();
14
15         do {
16             System.out.print(s:"Apakah anda ingin mengambil cuti (y/t)? ");
17             konfirmasi = sc.next();
18             if (konfirmasi.equalsIgnoreCase(anotherString:"y")) {
19                 System.out.print(s:"Jumlah hari: ");
20                 jumlahHari = sc.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             } while (jatahCuti>0);
30         }
31     }
```

2. 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:\dokumenreva\kulyeah\kelas\pemas\P.DASPRO-JOBSHEET> |
```

Pertanyaan

1. Apa kegunaan sintaks BREAK di dalam sintaks perulangan
 - untuk menghentikan program dan keluar dari 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.

```
jobsheet7 > J DoWhileCuti24.java > DoWhileCuti24 > main(String[])
1  import java.util.Scanner;
2
3  public class DoWhileCuti24 {
4
5      Run | Debug
6      public static void main(String[] args) {
7          Scanner sc = new Scanner (System.in);
8
9          int jatahCuti, jumlahHari;
10         String konfirmasi;
11
12         System.out.print(s:"Jatah cuti: ");
13         jatahCuti = sc.nextInt();
14         boolean isEmpty = false;
15         do {
16             System.out.print(s:"Apakah anda ingin mengambil cuti (y/t)? ");
17             konfirmasi = sc.next();
18             if (konfirmasi.equalsIgnoreCase(anotherString:"y")) {
19                 System.out.print(s:"Jumlah hari: ");
20                 jumlahHari = sc.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                     isEmpty = true;
28                 }
29             } while (jatahCuti>0 || isEmpty!=false);
30         }
31     }
```

```
Jatah cuti: 12
Apakah anda ingin mengambil cuti (y/t)? y
Jumlah hari: 10
Sisa jatah cuti: 2
Apakah anda ingin mengambil cuti (y/t)? y
Jumlah hari: 3
Sisa jatah cuti Anda tidak mencukupi
Apakah anda ingin mengambil cuti (y/t)? y
Jumlah hari: 1
Sisa jatah cuti: 1
Apakah anda ingin mengambil cuti (y/t)?
```

3. Push dan commit kode program ke github
4. Pada saat input konfirmasi, ketikkan "t", apa yang terjadi? Mengapa demikian?

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

program tidak berhenti seperti semestinya

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

```
Jatah cuti: 12
Apakah anda ingin mengambil cuti (y/t)? t
Anda tidak jadi mengajukan cuti, program selesai
PS D:\dokumenreva\kulyeah\kelas\pemdas\P.DASPRO-JOBSHEET>
```

6. Push dan commit kode program ke github