

Nama : Maharani Wirawan  
NIM : 2341760111  
Kelas : SIB 1C  
No.Abs : 18

## **JOBSHEET 7 PRAKTIKUM DASAR PEMROGRAMAN**

### **• PERCOBAAN 1**

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!

⇒ Tiga komponen perulangan FOR pada kode program tersebut adalah:

1. Inisialisasi: `int i = 1`
2. Kondisi: `i <= 50`
3. Pembaruan: `i++`

2. Jelaskan alur kerja dari potongan kode program berikut!

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

Kode program tersebut melakukan hal berikut:

1. Mulai dengan angka 1.
2. Periksa apakah angka tersebut bisa dibagi (kelipatan) oleh suatu angka tertentu yang disebut kelipatan.
3. Jika angka tersebut memang kelipatan dari kelipatan, tambahkan angka tersebut ke total yang disebut jumlah.
4. Selain itu, hitung berapa banyak angka yang memenuhi syarat tersebut, dan simpan jumlahnya di counter.
5. Ulangi langkah 2 hingga 4 untuk angka berikutnya, yaitu 2, 3, dan seterusnya hingga mencapai angka 50.
6. Setelah selesai, maka akan tahu berapa banyak angka yang memenuhi syarat tersebut (disimpan di counter) dan totalnya adalah jumlah.

Jadi, kode program ini digunakan untuk menghitung jumlah dari angka-angka antara 1 hingga 50 yang merupakan kelipatan dari suatu angka tertentu, dan juga menghitung berapa banyak angka tersebut.

## • PERCOBAAN 2

1. Tunjukkan bagian kode program yang digunakan sebagai syarat untuk menghentikan perulangan WHILE! Berapa kali perulangan dilakukan? Perulangan while dalam kode program ini dihentikan oleh bagian berikut:

⇒ Bagian kode program yang digunakan sebagai syarat untuk menghentikan perulangan while adalah sebagai berikut:

```
while (i < jumlahKaryawan) {  
    // ...  
    i++;  
    // ...  
}
```

Dalam kode program ini, perulangan akan berhenti ketika i mencapai nilai yang sama dengan jumlahKaryawan. Bagian yang mengubah nilai i adalah pernyataan i++; yang terdapat di dalam loop.

Jadi, perulangan dilakukan sebanyak jumlahKaryawan kali sesuai dengan nilai yang dimasukkan oleh pengguna sebagai jumlah karyawan.

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", pernyataan continue akan membuat program melewati perhitungan gaji lembur untuk direktur dan melanjutkan ke input jabatan berikutnya dalam 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!

⇒ Komponen i++ diletakkan di tengah untuk memberikan fleksibilitas dalam mengontrol alur program. Dengan penempatan ini, kita dapat melakukan tindakan sebelum nilai i ditingkatkan.

Dalam kode program yang diperbarui, penginkremen i++ ditempatkan di akhir perulangan. Hal ini memastikan bahwa "direktur" tidak dihitung dalam jumlah karyawan, dan perhitungan gaji lembur dilakukan setelah jenis jabatan diketahui. Ini memungkinkan

program berfungsi dengan benar dalam skenario di mana "direktur" adalah jabatan karyawan pertama.

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

```
1. import java.util.Scanner;
2.
3. public class WhileGaji18 {
4.     public static void main(String[] args) {
5.         Scanner scan = new Scanner(System.in);
6.
7.         int jumlahKaryawan, jumlahJamLembur;
8.         double gajiLembur = 0.0, totalGajiLembur = 0.0;
9.         String jabatan;
10.
11.         System.out.print("Masukkan jumlah karyawan: ");
12.         jumlahKaryawan = scan.nextInt();
13.         int i = 0;
14.
15.         while (i < jumlahKaryawan) {
16.             System.out.println("Pilihan jabatan - Direktur, Manajer, Karyawan");
17.             System.out.print("Masukkan jabatan karyawan ke-" + (i + 1) + ": ");
18.             jabatan = scan.next();
19.
20.             if (jabatan.equalsIgnoreCase("direktur")) {
21.                 System.out.print("Masukkan jumlah jam lembur: ");
22.                 jumlahJamLembur = scan.nextInt();
23.                 gajiLembur = jumlahJamLembur * 150000;
24.             } else if (jabatan.equalsIgnoreCase("manajer")) {
25.                 System.out.print("Masukkan jumlah jam lembur: ");
26.                 jumlahJamLembur = scan.nextInt();
27.                 gajiLembur = jumlahJamLembur * 100000;
28.             } else if (jabatan.equalsIgnoreCase("karyawan")) {
29.                 System.out.print("Masukkan jumlah jam lembur: ");
30.                 jumlahJamLembur = scan.nextInt();
31.                 gajiLembur = jumlahJamLembur * 75000;
32.             } else {
33.                 System.out.println("Jabatan invalid");
34.                 continue;
35.             }
36.
37.             totalGajiLembur += gajiLembur;
38.             System.out.println("Total gaji lembur: " + totalGajiLembur);
39.             i++;
40.         }
41.     }
42. }
```

```

D:\PDASPRO\Jobsheet-DasarPemrograman>java WhileGaji18
Masukkan jumlah karyawan: 3
Pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-1: manajer
Masukkan jumlah jam lembur: 10
Total gaji lembur: 1000000.0
Pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-2: direktur
Masukkan jumlah jam lembur: 5
Total gaji lembur: 1750000.0
Pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-3: pegawai
Jabatan invalid
Pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-3: 4
Jabatan invalid
Pilihan jabatan - Direktur, Manajer, Karyawan
Masukkan jabatan karyawan ke-3: karyawan
Masukkan jumlah jam lembur: 7
Total gaji lembur: 2275000.0

```

### • PERCOBAAN 3

1. Apa kegunaan sintaks BREAK di dalam sintaks perulangan?  
⇒ Sintaks break digunakan di dalam perulangan untuk menghentikan perulangan secara tiba-tiba dan keluar dari perulangan, bahkan jika kondisi perulangan masih memungkinkan berlanjut. Dalam kode program yang diberikan, break digunakan jika jumlah hari cuti yang diminta oleh pengguna melebihi jatah cuti yang tersedia. Ini digunakan untuk menghentikan perulangan saat jatah cuti tidak mencukupi.
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.

```

WhileCuti18.java > ...

public class DowWhileCuti18 {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int jatahCuti, jumlahHari;
        String konfirmasi;

        System.out.print("Jatah cuti: ");
        jatahCuti = sc.nextInt();

        do {
            System.out.print("Apakah anda ingin mengambil cuti (y/t)? ");
            konfirmasi = sc.next();

            if (konfirmasi.equalsIgnoreCase("y")) {
                System.out.print("Jumlah hari: ");
                jumlahHari = sc.nextInt();

                if (jumlahHari <= jatahCuti) {
                    jatahCuti -= jumlahHari;
                    System.out.println("Sisa jatah cuti: " + jatahCuti);
                } else {
                    System.out.println("Sisa jatah cuti Anda tidak mencukupi");
                    // Hapus pernyataan break di sini
                }
            }
        } while (jatahCuti > 0);
    }
}

```

4. Pada saat input konfirmasi, ketikkan "t", apa yang terjadi? Mengapa demikian?  
⇒ Ketika pengguna memasukkan "t" sebagai konfirmasi, program akan terus menanyakan apakah pengguna ingin mengambil cuti (y/t) setelah mencetak pesan "Apakah anda ingin mengambil cuti (y/t)? ". Ini terjadi karena tidak ada kondisi atau pernyataan yang mengatasi kasus "t" secara khusus. Program hanya berhenti ketika jatah cuti habis.
5. Modifikasi kode program sehingga saat pengguna mengetikkan "t" sebagai input konfirmasi, maka program berhenti

```
        System.out.println(x:"Sisa jatah cuti Anda tidak menc  
        break;  
    }  
    }  
    if (konfirmasi.equalsIgnoreCase(anotherString:"t")) {  
        break;  
    }  
} while (jatahCuti > 0);  
}
```

