LEMBAR KERJA 7

PERULANGAN 1

1. Tujuan

Setelah menyelesaikan topik ini, siswa harus mampu:

- Menjelaskan format pemrograman loop bagian 1
- Mengimplementasikan flowchart loop bagian 1 menggunakan bahasa pemrograman Java

2. Laboratorium

2.1 Eksperimen 1: Menghitung Kelipatan Menggunakan FOR

Waktu Eksperimen: 60 menit

Dalam percobaan ini, kode dibuat untuk menampilkan kelipatan suatu bilangan tertentu di dalamnya rentang 1 hingga 50 menggunakan perulangan FOR, dan untuk menghitung total angka-angka ini.

1. Buka editor teks. Buat File Java baru bernama **ForMultiplesStudentIDNumber.java**
2. Buat struktur dasar program Java yang berisi deklarasi **kelas** dan **main()** metode
3. Tambahkan perpustakaan **Pemindai .**
4. Membuat atau mendeklarasikan variabel bernama **input** dari perpustakaan **Pemindai .**
5. Buat variabel **int** bernama **multiple, sum,** dan **counter.** Inisialisasi **jumlah** variabel dan **penghitung** dengan 0.
6. Tambahkan kode berikut untuk mendapatkan input pengguna!

```
System.out.print(s:"Input the multiple = ");
multiple = input.nextInt();
```

7. Buat perulangan FOR dengan kondisi IF untuk mengevaluasi bilangan kelipatan

```
for(int i=1;i<=50;i++){
    if(i%multiple == 0){
        sum = sum + i;
        counter++;
        //System.out.print(i+"-");
    }
}
```

8. Menampilkan jumlah dan penghitung kelipatan angka dalam rentang 1 hingga 50.

```
System.out.printf(format:"There are %d numbers that are multiple of %d in range 1 to 50.\n", counter, multiple);
System.out.printf(format:"The sum from all multiples of %d in range 1 s.d. 50 is %d. \n", multiple, sum);
```

9. Jalankan program dan analisis hasilnya. Hasil Anda harus seperti ini:

```
Input the multiple = 5
There are 10 numbers that are multiple of 5 in range 1 to 50.
The sum from all multiples of 5 in range 1 s.d. 50 is 275.
```

10. Komit dan dorong perubahan ke GitHub.

Pertanyaan

1. Ada 3 komponen utama dalam loop FOR. Berdasarkan percobaan 1 di atas, tentukan dan jelaskan 3 komponen tersebut!
2. Jelaskan cara kerja kode berikut!

```
for(int i=1;i<=50;i++){
    if(i%multiple == 0){
        sum = sum + i;
        counter++;
        //System.out.print(i+"-");
    }
}
```

3. Ubah kode yang ada dengan menambahkan variabel baru untuk menghitung rata-rata semuanya kelipatan tertentu! Dorong dan komit kode program ke GitHub.
4. Buat file program Java baru bernama **WhileMultiplesStudentIDNumber.java.** Membuat kode yang setara dengan menggunakan while loop. Dorong dan komit kode ke GitHub.

2.2 Eksperimen 2: Menghitung Gaji Lembur Karyawan Menggunakan WHILE dan MELANJUTKAN

Waktu Eksperimen: 60 menit

Sebuah perusahaan memberikan upah lembur kepada karyawannya setiap minggu. Upah lemburnya adalah dihitung berdasarkan jabatan pegawai dan jumlah jam lembur dalam a pekan. Karyawan dengan posisi 'direktur' tidak menerima tambahan waktu lembur gaji meskipun mereka bekerja lembur, karyawan dengan posisi 'manajer' menerimanya upah lembur sebesar 100.000 per jam, sedangkan pegawai dengan jabatan 'staf' menerima upah lembur sebesar 75.000 per jam. Pada percobaan ini, kode program dibuat dengan menggunakan SAAT dan LANJUTKAN untuk menghitung pengeluaran perusahaan.

1. Buka editor teks dan buat file Java baru bernama

   WhileOvertimePayStudentIDNumber.java

2. Buat struktur dasar program Java yang berisi deklarasi **kelas** dan **main()**

   metode

3. Tambahkan perpustakaan **Pemindai .**

4. Membuat atau mendeklarasikan variabel bernama **input** dari perpustakaan **Pemindai .**

5. Deklarasikan variabel **int** bernama **numEmployee** dan **overtimeHours,** lalu **overtimePay**

   dan **totalOvertimePay** dengan tipe data **ganda .** Inisialisasi **pembayaran lembur** dan

   total Pembayaran Lembur dengan 0

6. Deklarasikan **posisi** variabel dengan tipe data **String .**

7. Tambahkan kode berikut untuk mendapatkan input pengguna untuk variabel **numEmployee .**

```
System.out.print(s:"Employee number = ");
numEmployee = input.nextInt();
```

8. Buat struktur perulangan WHILE dengan kondisional IF-ELSE dan LANJUTKAN ke

   menentukan upah lembur berdasarkan jabatan pegawai

```
int i=0;
while(i<numEmployee){
    System.out.print("Position of employee "+(i+1)+" (director, manager, staff) = ");
    position = input.next();
    System.out.print("Employee "+(i+1)+" overtime hours = ");
    overtimeHours = input.nextInt();
    i++;

    if(position.equalsIgnoreCase(anotherString:"director")){
        continue;
    }else if(position.equalsIgnoreCase(anotherString:"manager")){
        overtimePay=overtimeHours*100000;
    }else if(position.equalsIgnoreCase(anotherString:"staff")){
        overtimePay=overtimeHours*75000;
    }

    totalOvertimePay += overtimePay;
}
```

9. Menampilkan **total Pembayaran Lembur**

```
System.out.println("Total of Overtime Pay = "+totalOvertimePay);
```

10. Jalankan program dan analisis hasilnya. Hasil Anda harus seperti ini:

```
Employee number = 3
Position of employee 1 (director, manager, staff) = manager
Employee 1 overtime hours = 1
Position of employee 2 (director, manager, staff) = director
Employee 2 overtime hours = 10
Position of employee 3 (director, manager, staff) = staff
Employee 3 overtime hours = 5
Total of Overtime Pay = 475000.0
```

11. Komit dan dorong perubahan ke GitHub.

Pertanyaan

1. Tunjukkan bagian kode program yang digunakan sebagai syarat untuk menghentikan perulangan WHILE! Bagaimana berapa kali loop dieksekusi?
2. Dalam kode ini,

```
if(position.equalsIgnoreCase(anotherString:"director")){
    continue;
```

Apa yang sebenarnya terjadi jika variabel **'posisi'** berisi nilai 'DIRECTOR'? Apa penggunaan CONTINUE dalam struktur loop?

3. Mengapa komponen iterasi 'i++' ditempatkan di tengah, bukan di akhir while memblokir? Pindahkan 'i++' ke akhir blok while, lalu jalankan kembali program dengan mengetik 'DIREKTUR' sebagai **jabatan** bagi karyawan pertama. Apa yang terjadi? Menjelaskan!
4. Ubah kode program untuk menangani posisi yang tidak valid seperti contoh berikut:

```
Employee number = 3
Position of employee 1 (director, manager, staff) = director
Employee 1 overtime hours = 5
Position of employee 2 (director, manager, staff) = manager
Employee 2 overtime hours = 10
Position of employee 3 (director, manager, staff) = programmer
Employee 3 overtime hours = 4
Invalid position!
Position of employee 3 (director, manager, staff) = staff
Employee 3 overtime hours = 4
Total of Overtime Pay = 1300000.0
```

5. Komit dan dorong perubahan ke GitHub.

2.3 Eksperimen 3: Menghitung Hak Cuti Menggunakan DO-WHILE

Waktu Eksperimen: 50 menit

Pada percobaan ini dibuat kode program menggunakan DO-WHILE untuk menghitung **cuti hak** seorang karyawan. Karyawan berhak mendapat cuti selama 5 hari. Hari-hari cuti akan terjadi

dikurangi setiap kali digunakan. Ketika cuti hanya tersisa 2 hari, maka

karyawan menerima peringatan untuk berhenti menggunakan cuti mereka

1. Buka editor teks dan buat file Java baru bernama

   DoWhileLeaveEntitlementStudentIDNumber.java

2. Buat struktur dasar program Java yang berisi deklarasi kelas dan main()

   metode

3. Tambahkan perpustakaan **Pemindai .**

4. Membuat atau mendeklarasikan variabel bernama **input** dari perpustakaan **Pemindai .**

5. Buat variabel **LeaveEntitlement** dan **numLeave** dengan tipe data int.

6. Buat **konfirmasi** variabel dengan tipe data **String .**

7. Buat struktur loop DO-WHILE untuk mendapatkan input pengguna dari keyboard dan

   menghitung hak cuti

```
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");
            break;
        }
    }
} while (jatahCuti > 0);
```

8. Jalankan program dan analisa hasilnya. Itu harus sama dengan yang berikut ini

   keluaran.

```
Number of Leave Entitlement = 12
Do you want to take a leave? (y/n) = y
Leave number = 4
The remaining leave entitlement = 8
Do you want to take a leave? (y/n) = y
Leave number = 5
The remaining leave entitlement = 3
Do you want to take a leave? (y/n) = y
Leave number = 4
The remaining leave entitlement is not sufficient!
```

9. Komit dan dorong perubahan ke GitHub.

Pertanyaan

1. Apa gunanya BREAK dalam sintaks loop?
2. Memodifikasi program sehingga jika jumlah hari cuti yang diminta lebih besar dari jumlah hari cuti yang diminta sisa hak, program tidak berhenti, memungkinkan pengguna untuk masuk jumlah hari sesuai dengan haknya.
3. Komit dan dorong kode program ke GitHub.
4. Saat mengetik "t" sebagai input konfirmasi, apa yang terjadi? Mengapa?
5. Ubah kode program sehingga ketika pengguna memasukkan "t" sebagai input konfirmasi, maka program akan berhenti.
6. Komit dan dorong kode program ke GitHub.

3. Tugas

Waktu Eksperimen: 130 menit

• Mengimplementasikan flowchart yang sudah dibuat pada tugas Dasar Kursus pemrograman yang terkait dengan proyek ke dalam program Anda.

• Dorong dan komit hasil kode program Anda ke repositori proyek Anda.

• Catatan: tugas hanya menerapkan materi dari sesi 1 sampai 7