

JOBSHEET 7

PERULANGAN 1

1. Tujuan

- Mahasiswa dapat menjelaskan format penulisan program perulangan bagian 1
- Mahasiswa dapat mengimplementasikan flowchart perulangan bagian 1 menggunakan bahasa pemrograman Java

2. Praktikum

2.1 Percobaan 1: Menghitung Bilangan Kelipatan Menggunakan FOR

Waktu Percobaan: 60 menit

Pada percobaan ini dilakukan pembuatan kode program untuk menampilkan bilangan kelipatan angka tertentu dari rentang 1 sampai dengan 50 menggunakan perulangan FOR, serta menghitung total dari bilangan-bilangan tersebut.

1. Buka text editor. Buat file baru, beri nama **ForKelipatanNoAbsen.java**
2. Tuliskan struktur dasar bahasa pemrograman Java yang terdiri dari fungsi **main()**
3. Tambahkan library Scanner
4. Buat deklarasi **Scanner** dengan nama **scan**
5. Buatlah variabel bertipe **int** dengan nama **kelipatan**, **jumlah**, dan **counter**. Inisialisasi variabel **jumlah** dan **counter** dengan nilai 0
6. Tambahkan kode berikut ini untuk menerima input dari keyboard

```
System.out.print("Masukkan bilangan kelipatan (1-9): ");
kelipatan = scan.nextInt();
```

7. Buat struktur perulangan FOR dengan kondisi pemilihan IF untuk menentukan bilangan kelipatan

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

8. Tampilkan banyaknya bilangan kelipatan dan total bilangan kelipatan pada rentang 1 sampai dengan 50.



```
System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);
9. System.out.printf("Total bilangan kelipatan %d dari 1 sampai 50 adalah %d\n", kelipatan, total);
```

gambar berikut ini

```
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
```

10. Commit dan push kode program ke github

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 :

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

int i = 1 , berfungsi untuk mendeklarasikan variable yang nantinya akan digunakan sebagai acuan saat looping.

i <= 50, berfungsi untuk memberikan statement looping, jika nilai bersifat true akan menjalankan kode didalamnya, namun jika bernilai false maka akan menghentikan perulangan

i++ , berfungsi untuk mengupdate variable i, agar nilainya bertambah 1 (increment)

2. Jelaskan alur kerja dari potongan kode program berikut!

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

Jawaban :

Perulangan akan dijalankan sesuai inisialisasi variable dan statement, dimana dimulai dari 1 hingga 50, kemudian di akhir akan melakukan increment. Di dalam setiap literasi akan terdapat kondisi jika nilai dari variable *i* jika habis dibagi dengan bilangan kelipatan maka variable *total* akan menambahkan value sebelumnya dengan value dari variable *i*, kemudian variable *counter* akan melakukan increment.

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.

```

orkelipatan08.java 7  Forkelipatan08 7  main(String[])
import java.util.Scanner;
public class Forkelipatan08 {
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        int kelipatan, total=0, jumlah=0, counter=0, avg;

        System.out.print("Masukkan bilangan kelipatan (1-9) : ");
        kelipatan = scan.nextInt();

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

        avg = total / counter;

        System.out.printf(format:"Banyak bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);
        System.out.printf(format:"Total bilangan kelipatan %d dari 1 sampai 50 adalah %d\n", kelipatan, total);
        System.out.printf(format:"Rata rata dari seluruh bilangan kelipatan %d dari 1 sampai 50 adalah %d\n", kelipatan, avg);
    }
}

```

```

Rata rata dari seluruh bilangan kelipatan 4 dari 1 sampai 50 adalah 26
PS E:\POLINEMA\1. Ganjil 2023-2024\Daspro\Tugas\jobsheet7\java> git add .
PS E:\POLINEMA\1. Ganjil 2023-2024\Daspro\Tugas\jobsheet7\java> git commit -m "latihan 1 percobaan 2"
[master ca7cdbf] latihan 1 percobaan 2
1 file changed, 4 insertions(+), 1 deletion(-)
PS E:\POLINEMA\1. Ganjil 2023-2024\Daspro\Tugas\jobsheet7\java> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 6 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 377 bytes | 377.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/dzulfikar05/daspro-jobsheet7.git
38e17f1..ca7cdbf master -> master
PS E:\POLINEMA\1. Ganjil 2023-2024\Daspro\Tugas\jobsheet7\java>

```

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 :

```
import java.util.Scanner;

public class WhileKelipatan08 {
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        int kelipatan, total=0, jumlah=0, counter=0, avg;

        System.out.print(s:"Masukkan bilangan kelipatan (1-9) : ");
        kelipatan = scan.nextInt();

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

        avg = total / counter;

        System.out.printf(format:"Banyak bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);
        System.out.printf(format:"Total bilangan kelipatan %d dari 1 sampai 50 adalah %d\n", kelipatan, total);
        System.out.printf(format:"Rata rata dari seluruh bilangan kelipatan %d dari 1 sampai 50 adalah %d\n", kelipatan, avg);
    }
}

PS E:\!POLINEMA\1. Ganjil 2023-2024\Daspro\Tugas\jobsheet7\java> git add .
PS E:\!POLINEMA\1. Ganjil 2023-2024\Daspro\Tugas\jobsheet7\java> git commit -m "larihan 2 percobaan 1"
[master baff083] larihan 2 percobaan 1
1 file changed, 26 insertions(+)
create mode 100644 WhileKelipatan08.java
PS E:\!POLINEMA\1. Ganjil 2023-2024\Daspro\Tugas\jobsheet7\java> git push -u origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 6 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 642 bytes | 642.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/dzulfekar05/daspro-jobsheet7.git
ca7cbbf..baff083 master -> master
branch 'master' set up to track 'origin/master'.
PS E:\!POLINEMA\1. Ganjil 2023-2024\Daspro\Tugas\jobsheet7\java>
```

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

Waktu Percobaan: 60 menit

Sebuah perusahaan memberikan gaji lembur kepada karyawannya setiap minggu. Gaji tersebut dihitung berdasarkan jabatan karyawan dan jumlah jam lembur dalam seminggu. Karyawan dengan jabatan “direktur” tidak mendapatkan tambahan gaji meskipun melakukan

lembur, karyawan dengan jabatan “manager” mendapatkan gaji lembur sebesar 100000 per jam, sedangkan karyawan dengan jabatan “staf” mendapatkan gaji lembur sebesar 75000 per jam. Pada percobaan ini dilakukan pembuatan kode program menggunakan WHILE dan CONTINUE untuk menghitung pengeluaran perusahaan.

1. Buka text editor. Buat file baru, beri nama **WhileGajiNoAbsen.java**
2. Tuliskan struktur dasar bahasa pemrograman Java yang terdiri dari fungsi **main()**
3. Tambahkan library Scanner
4. Buat deklarasi **Scanner** dengan nama **scan**
5. Deklarasikan variabel **jumlahKaryawan** dan **jumlahJamLembur** bertipe int serta **gajiLembur** dan **totalGajiLembur** bertipe double. Inisialisasi variabel **gajiLembur** dan **totalGajiLembur** dengan nilai 0
6. Deklarasikan variabel **jabatan** bertipe String
7. Tambahkan kode berikut ini untuk menerima input dari keyboard guna menentukan jumlah karyawan yang akan dihitung gajinya

```
System.out.print("Masukkan jumlah karyawan: ");
jumlahKaryawan = sc.nextInt();
```

8. Buat struktur perulangan WHILE dengan kondisi pemilihan IF-ELSE dan CONTINUE untuk menentukan gaji lembur berdasarkan jabatan karyawan

```
int i = 0;

while (i < jumlahKaryawan) {
    System.out.println("Pilihan jabatan - Direktur, Manajer, Karyawan");
    System.out.print("Masukkan jabatan karyawan ke-" + (i+1) + ": ");
    jabatan = sc.next();
    System.out.print("Masukkan jumlah jam lembur: ");
    jumlahJamLembur = sc.nextInt();
    i++;

    if (jabatan.equalsIgnoreCase("direktur")) {
        continue;
    } else if (jabatan.equalsIgnoreCase("manajer")) {
        gajiLembur = jumlahJamLembur * 100000;
    }
    else if (jabatan.equalsIgnoreCase("karyawan")) {
        gajiLembur = jumlahJamLembur * 75000;
    }

    totalGajiLembur += gajiLembur;
}
```

9. Tampilkan hasil perhitungan jumlah gaji

```
System.out.println("Total gaji lembur: " + totalGajiLembur);
```

10. Jalankan program tersebut. Cocokkan hasil compile kode program Anda dengan gambar berikut ini

```
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
```

11. Push dan commit kode program ke github.

Pertanyaan

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

Jawaban :

```
while(i < jumlahKaryawan){
```

Looping akan berhenti jika value dari variable i sudah menyamai dari jumlah karyawan

Looping akan di ulang sebanyak jumlah karyawan yang di inputkan, karena i dimulai dari angka 0 dan kurang dari jumlah karyawan 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 :

Jika pada variable jabatan berisi Direktur, maka program akan berjalan semestinya tanpa membandingkan besar kecil nya karakter pada variable jabatan. Peran dari continue adalah sebagai kondisi yang akan langsung melanjutkan proses 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 posisi Tengah untuk melakukan increment pada variable i.

jika **i++** diletakkan di posisi akhir dan jabatan di isi direktur yang pertama maka yang terjadi looping akan terus berjalan tanpa henti.

Hal ini karena terdapat kondisi dimana jika jabatan == direktur, maka akan melakukan fungsi continue, yang artinya akan mengskip kode kode di bawahnya dan akan Kembali ke perulangan, variable i belum sempat melakukan increment, jadi looping akan terus berjalan tanpa henti.

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 :

```
Masukkan jumlah karyawan :1
Pilih jabatan - direktur, manager, karyawan
Masukkan jabatan karyawan ke-1 : pegawai
Masukkan jumlah jam lembur : 2
Jabatan Invalid
Pilih jabatan - direktur, manager, karyawan
Masukkan jabatan karyawan ke-1 : karyawan
Masukkan jumlah jam lembur : 1
Total gaji lembur : 75000.0
```

5. Push dan commit kode program ke github

Jawaban :

```
PS E:\POLINEMA\1. Ganjil 2023-2024\Daspro\Tugas\jobsheet7\java> git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 6 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 356 bytes | 356.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/dzulfikar05/daspro-jobsheet7.git
   ca50cd4..f8b053d  master -> master
branch 'master' set up to track 'origin/master'.
PS E:\POLINEMA\1. Ganjil 2023-2024\Daspro\Tugas\jobsheet7\java>
```

2.3 Percobaan 3: Menghitung Jatah Cuti Menggunakan DO-WHILE

Waktu Percobaan: 50 menit

Pada percobaan ini dilakukan pembuatan kode program menggunakan DO-WHILE untuk menghitung jatah cuti yang dimiliki oleh pegawai. Pegawai mempunyai jatah cuti sebanyak 5 hari. Jatah cuti akan dikurangi perhati setiap kali digunakan. Saat jatah cuti sisa 2 hari, pegawai mendapat peringatan untuk berhenti menggunakan jatah cutinya.

1. Buka text editor. Buat file baru, beri nama **DoWhileCutiNoAbsen.java**
2. Tuliskan struktur dasar bahasa pemrograman Java yang terdiri dari fungsi **main()**
3. Tambahkan library Scanner
4. Buat deklarasi **Scanner** dengan nama **sc**
5. Buatlah variabel **jatahCuti** dan **jumlahHari** bertipe **int**
6. Buatlah variabel **konfirmasi** bertipe **String**
7. Buat struktur perulangan DO-WHILE untuk menerima input dari keyboard dan menghitung jatah 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 tersebut. Cocokkan hasil *running* program yang sudah Anda buat apakah sudah sesuai dengan tampilan berikut

```
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
```

9. Push dan commit kode program ke github

Pertanyaan

1. Apa kegunaan sintaks BREAK di dalam sintaks perulangan?

Jawaban :

Break di gunakan untuk menghentikan 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.

Jawaban :

```
import java.util.Scanner;

public class DowhileCuti08 {
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int jatahCuti, jumlahHari;
        String konfirmasi;

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

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

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

                if(jumlahHari <= jatahCuti){
                    jatahCuti -= jumlahHari;
                    System.out.print("Sisa jatah cuti : "+jatahCuti);

                }else{
                    System.out.println(x:"Sisa jatah cuti Anda tidak mencukupi");
                    // break;
                }
            }
        }while(jatahCuti > 0);
    }
}
```

3. Push dan commit kode program ke github

```
PS E:\POLINEMA\1. Ganjil 2023-2024\Daspro\Tugas\jobsheet7\java> git add .
PS E:\POLINEMA\1. Ganjil 2023-2024\Daspro\Tugas\jobsheet7\java> git commit -m "remove break"
[master 6bb22c7] remove break
 1 file changed, 1 insertion(+), 1 deletion(-)
PS E:\POLINEMA\1. Ganjil 2023-2024\Daspro\Tugas\jobsheet7\java> git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 6 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 301 bytes | 301.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/dzulfikar05/daspro-jobsheet7.git
 64a6727..6bb22c7 master -> master
branch 'master' set up to track 'origin/master'.
PS E:\POLINEMA\1. Ganjil 2023-2024\Daspro\Tugas\jobsheet7\java>
```

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

Jawaban :

```
Jatah cuti : 4
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
```

Hal ini terjadi karena tidak aksi yang dilakukan ketika menginputkan value selain "y". Jadi program akan terus berjalan.

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

Jawaban :

```
import java.util.Scanner;

public class DowhileCuti08 {
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int jatahCuti, jumlahHari;
        String konfirmasi;

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

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

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

                if(jumlahHari <= jatahCuti){
                    jatahCuti -= jumlahHari;
                    System.out.print("Sisa jatah cuti : "+jatahCuti);

                }else{
                    System.out.println(x:"Sisa jatah cuti Anda tidak mencukupi");
                    // break;
                }
            }else{
                break;
            }
        }while(jatahCuti > 0);
    }
}
```

6. Push dan commit kode program ke github

```
PS E:\POLINEMA\1. Ganjil 2023-2024\Daspro\Tugas\jobsheet7\java> git add .
PS E:\POLINEMA\1. Ganjil 2023-2024\Daspro\Tugas\jobsheet7\java> git commit -m "add break"
[master 41b7793] add break
1 file changed, 2 insertions(+)
PS E:\POLINEMA\1. Ganjil 2023-2024\Daspro\Tugas\jobsheet7\java> git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 6 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 306 bytes | 306.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/dzulfikar05/daspro-jobsheet7.git
6bb22c7..41b7793 master -> master
branch 'master' set up to track 'origin/master'.
```

3. Tugas

Waktu Percobaan : 130 Menit

- Implementasikan flowchart yang telah dibuat pada tugas pertemuan 7 Matakuliah Dasar Pemrograman terkait project ke dalam kode program



- Push dan commit hasil kode program anda ke repository project Anda
- Catatan: tugas hanya boleh menerapkan materi dari pertemuan 1 hingga pertemuan 7.

Jawaban :

Link Github

<https://github.com/dzulfikar05/kasirCafe>