

Laporan Hasil Praktikum Dasar Pemrograman

Latihan-7 : Praktikum Dasar Pemrograman



Nama : Ghoffar Abdul Ja'far

NIM : 41720035

Kelas : 1E

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2023/2024

2.1 Percobaan 1: Menghitung Bilangan Kelipatan Menggunakan FOR

- Menulis syntax

```
Scanner scan = new Scanner(System.in);

int kelipatan, total = 0, counter = 0;

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

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

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

- Hasil

```
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
```

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!
 - Pernyataan `int i = 0`: merupakan inisialisasi variabel
 - Pernyataan `i < 50`: merupakan pernyataan kondisi, apakah perulangan berlanjut atau berhenti
 - Pernyataan `i++`: merupakan fungsi untuk menguiah nilai suatu variabel
2. Jelaskan alur kerja dari potongan kode program berikut!

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

- Dimulai dengan menginisialisasi variabel `i=1`
- Setelah itu menyatakan sebuah kondisi `i<=50`
- Menyatakan penambahan nilai dari variabel `i` pada perulangan dengan operator increment `i++`

- Didalam blok perulangan ada blok percabangan if dengan kondisi ($i \% kelipatan == 0$)
 - Pada blok percabangan, jika i habis dibagi kelipatan sama dengan 0 maka nilai dari variabel total akan di tambah nilai dengan nilai dari variabel i dan kemudian nilai dari counter akan ditambah 1
 - Jika i sudah lebih dari 50 maka 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.
- Menulis syntax

```
Scanner scan = new Scanner(System.in);

int kelipatan, total = 0, counter = 0;
float rataKelipatan = 0f;

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

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

rataKelipatan = jumlah / rataKelipatan;
System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %d\n",
kelipatan, counter);
System.out.printf("total bilangan kelipatan %d dari 1 sampai 50 adalah
%d\n", kelipatan, total);
System.out.printf("Rata-rata bilangan kelipatan %d dari 1 sampai 50
adalah %f\n", kelipatan, rataKelipatan);
```

- Hasil

```
Masukkan bilangan kelipatan (1-9): 2
Banyaknya bilangan 2 dari 1 sampai 50 adalah 25
Total bilangan kelipatan 2 dari 1 sampai 50 adalah 650
Rata-rata bilangan kelipatan 2 dari 1 sampai 50 adalah 26.000000
```

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

- Menulis syntax

```
Scanner scan = new Scanner(System.in);

int kelipatan, total = 0, counter = 0;
float rataKelipatan = 0f;

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

int i = 1;

while (i <= 50) {
    if (i % kelipatan == 0) {
        jumlah += i;
        counter++;
        rataKelipatan++;
    }
    i++;
}
rataKelipatan = jumlah / counter;
System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);
```

- Hasil

```
Masukkan bilangan kelipatan (1-9): 2
Banyaknya bilangan 2 dari 1 sampai 50 adalah 25
Total bilangan kelipatan 2 dari 1 sampai 50 adalah 650
Rata-rata bilangan kelipatan 2 dari 1 sampai 50 adalah 26.000000
```

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

- Menulis syntax

```
Scanner sc = new Scanner(System.in);

int jumlahKaryawan, jumlahJamLembur;
double gajiLembur = 0, totalGajiLembur = 0;
String jabatan;

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

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;
}

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

- Hasil

```
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?

= Pada pernyataan while dan increment i (i++)

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?

- Program akan tetap berjalan karena pada kondisi pertama menggunakan .equalsIgnoreCase() yang mana tidak akan masalah dengan perbedaan besar kecilnya huruf
- Keyword continue berfungsi untuk melanjutkan perulangan ke perulangan selanjutnya dengan langsung melewati kode yang belum dieksekusi

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!

= Yang terjadi bila i++ diletakkan dibagian akhir perulangan dan memasukkan “direktur” akan membuat loop tidak ada kemajuan. Karena dengan mengisi “direktur” akan membuat perulangan lanjut ke perulangan dengan melewati kode yang belum dieksekusi, yang dimana i++ berada dibawah continue

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
```

- Menulis syntax

```
...  
if (jabatan.equalsIgnoreCase("direktur")) {  
    continue;  
} else if (jabatan.equalsIgnoreCase("manajer")) {  
    gajiLembur = jumlahJamLembur * 100000;  
} else if (jabatan.equalsIgnoreCase("karyawan")) {  
    gajiLembur = jumlahJamLembur * 75000;  
} else {  
    System.out.println("Jabatan Invalid");  
    i--;  
    continue;  
}  
...
```

- Hasil

```
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
```

2.3 Percobaan 3: Menghitung Jatah Cuti Menggunakan DO-WHILE

- Menulis syntax

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

- Hasil

```
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?
= Untuk menghentikan perulangan secara paksa
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.

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

3. Push dan commit kode program ke github
4. Pada saat input konfirmasi, ketikkan “t”, apa yang terjadi? Mengapa demikian?
= Program hanya akan terus mengulang ke bagian perulangan awal. Karena tidak ada bagian manapun yang handle huruf selain “y”
5. Modifikasi kode program sehingga saat pengguna mengetikkan “t” sebagai input konfirmasi, maka

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

program akan berhenti

6. Push dan commit kode program ke github

Tugas

```
...
do {
    System.out.println("=====");
    System.out.println("                Tarik Uang Tunai");
    System.out.println("=====");
    System.out.println("isi uang yang ingin ditarik");
    System.out.print("Isi nominal : ");
    tarik = input.nextDouble();
    saldo -= tarik;
    System.out.println("sisa saldo : " + saldo);
    System.out.println("jumlah uang yang ditarik : " + tarik);
    System.out.println("=====");
    System.out.println("\n1.keluar");
    System.out.println("2.Kembali");
    System.out.print("\nPilih Menu : ");
    pilihan2 = input.nextInt();
    switch (pilihan2) {
        case 1:
            System.exit(0);
            break;
    }
    break;
} while (pilihan2 != 2);
...
```