

JOB SHEET 8

PERULANGAN 2

1. Tujuan

- Memahami konsep perulangan bersarang pada algoritma
- Mahasiswa mampu menggambarkan flowchart perulangan bersarang
- Mengenal sintaks perulangan bersarang
- Mampu menerapkan perulangan bersarang dalam persoalan

2. Praktikum

2.1 Percobaan 1: Review Perulangan 1

Waktu Percobaan: 30 menit

Percobaan ini ditujukan untuk me-review kembali perulangan yang telah dibahas pada pertemuan sebelumnya. Pada percobaan 1 akan dibuat program untuk membuat tampilan * sebanyak n kali ke arah **samping**.

1. Buat folder baru pada repositori lokal Anda, beri nama **jobsheet8**
2. Buat file baru **StarNoPresensi.java**
3. Buat fungsi/method **main()** di dalamnya.
4. Karena program membutuhkan input dari keyboard, maka perlu import class Scanner. Jadi tambahkan sintaks import di baris atas sendiri program.

```
import java.util.Scanner;
```

5. Di dalam fungsi **main()** yang telah dibuat, deklarasikan objek **Scanner** dengan nama **sc**.

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

6. Pada baris selanjutnya, tampilkan instruksi untuk memasukkan nilai yang akan disimpan ke variabel **n**.

```
System.out.print("Masukkan nilai n = ");  
int n = sc.nextInt();
```

7. Pada baris selanjutnya, buat sintaks perulangan dengan for seperti di bawah ini.

```
for(int i=1; i<=n; i++){  
    System.out.print("*");  
}
```

Catatan: perlu diperhatikan, bahwa yang digunakan adalah perintah **print**, bukan **println** karena kita ingin menampilkan tanpa ada baris baru

8. Compile dan jalan program.
9. Amati hasilnya, maka hasilnya harusnya akan serupa dengan tampilan di bawah ini.



```
Masukkan nilai n = 5
*****
```

10. Commit program Anda ke Github dengan pesan "Percobaan 1"

Pertanyaan

1. Jika pada perulangan for, inisialisasi `i=1` diubah menjadi `i=0`, apa akibatnya? Mengapa bisa demikian?
2. Jika pada perulangan for, kondisi `i<=n` diubah menjadi `i<n`, bagaimana bentuk outputnya jika input n = 5? Mengapa hasilnya berbeda?
3. Jika pada perulangan for, kondisi `i <= n` diubah menjadi `i > n`, apa akibatnya? Mengapa bisa demikian?
4. Jika pada perulangan for, kondisi step `i++` diubah menjadi `i--` apa akibatnya? Mengapa bisa demikian?
5. Jika pada perulangan for, step `i++` diubah menjadi `i += 2`, bagaimana pola outputnya jika input n = 6? Apa yang menyebabkan perubahan tersebut?

2.2 Percobaan 2: Bintang Persegi

Waktu Percobaan: 30 menit

1. Pada percobaan ke-2 akan dilakukan percobaan tentang **nested loop**. Kasus yang akan diselesaikan adalah untuk membuat tampilan bujursangkar *, dengan panjang sisi sebanyak n. Misalkan n dimasukan 5, maka hasilnya adalah:

```
*****
*****
*****
*****
*****
```

2. Kalau diamati lebih lanjut, sebenarnya mirip dengan kasus percobaan 1. Jika di percobaan 1, misal input n bernilai 5, maka yang akan dihasilkan adalah ***** (bisa dianggap ini sebagai **inner loop** yang mencetak 5 bintang *****), maka untuk kasus percobaan 2 ini, hasil dari percobaan 1 tersebut hanya perlu diulang lagi sebanyak n kali. Yaitu dengan menambahkan **outer loop** untuk mengulangi proses **inner loop** sebanyak n kali.
6. Buat file baru **SquareNoPresensi.java**
7. Karena program membutuhkan input dari keyboard, maka perlu import class Scanner. Jadi tambahkan sintaks import di baris atas sendiri program.

```
import java.util.Scanner;
```

8. Buat method **main()**, dan isikan kode program yang sama dengan isi method main() di percobaan 1.

```
Scanner sc = new Scanner(System.in);
System.out.print("Masukkan nilai N = ");
int N = sc.nextInt();
for(int i=1; i<=N; i++){
    System.out.print("*");
}
```

9. Compile dan jalankan program. Pastikan program berjalan seperti saat menjalankan percobaan 1.
10. Perhatikan sintaks perulangan yang digunakan untuk mencetak * sebanyak n kali ke arah samping. Di langkah 8 di atas kode **for** (kotak merah) dijadikan sebagai *inner loop*.
11. Perulangan pada inner loop dilakukan sebanyak n kali untuk menghasilkan output seperti pada poin 1. Maka perlu ditambahkan perulangan luar (*outer loop*).

```
for(int iOuter=1; iOuter<=n; iOuter++){
    for(int i=1; i<=n; i++){
        System.out.print("*");
    }
    System.out.println();
}
```

12. Simpan perubahan, compile dan jalankan program.
13. Amati hasilnya, maka hasilnya harusnya akan serupa dengan tampilan di bawah ini.

```
Masukkan nilai n = 5
*****
*****
*****
*****
*****
```

14. Commit program Anda ke Github dengan pesan "Percobaan 2"

Pertanyaan

1. Perhatikan perulangan luar. Jika pada sintaks **for**, inisialisasi **iOuter=1** diubah menjadi **iOuter=0**, apa yang akibatnya? Mengapa bisa demikian?
2. Kembalikan program semula dimana inisialisasi **iOuter=1**. Kemudian perhatikan perulangan dalam. Jika pada sintaks **for**, inisialisasi **i=1** diubah menjadi **i=0**, apa yang akibatnya? Mengapa bisa demikian?
3. Apakah perbedaan kegunaan antara perulangan luar dengan perulangan yang berada di dalamnya?
4. Mengapa perlu ditambahkan sintaks **System.out.println();** di bawah perulangan dalam? Apa akibatnya jika sintaks tersebut dihilangkan?

5. Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan 2”

2.3 Percobaan 3: Bintang Segitiga

Waktu Percobaan: 45 menit

1. Pada percobaan ke-3 akan dilakukan percobaan segitiga *, dengan sama siku dengan tinggi sebesar n. Misalkan n dimasukan 5, maka hasilnya adalah:

```
*
**
***
****
*****
```

2. Buat file baru **TriangleNoPresensi.java**
3. Karena program membutuhkan input dari *keyboard*, maka perlu import class Scanner.
4. Buat method **main()**, dan isikan kode program berikut kedalam method **main()**.

```
Scanner sc = new Scanner(System.in);
System.out.print("Masukkan nilai n = ");
int n = sc.nextInt();
int i = 0;
while(i <= n) {
    int j = 0;
    while(j < i) {
        System.out.print("*");
        j++;
    }
    i++;
}
```

5. Compile dan jalankan program. Amati apa yang terjadi.
6. Commit program Anda ke Github dengan pesan “Percobaan 3”

Pertanyaan

1. Perhatikan, apakah output yang dihasilkan dengan nilai n = 5 sesuai dengan tampilan berikut?

```
*
**
***
****
*****
```

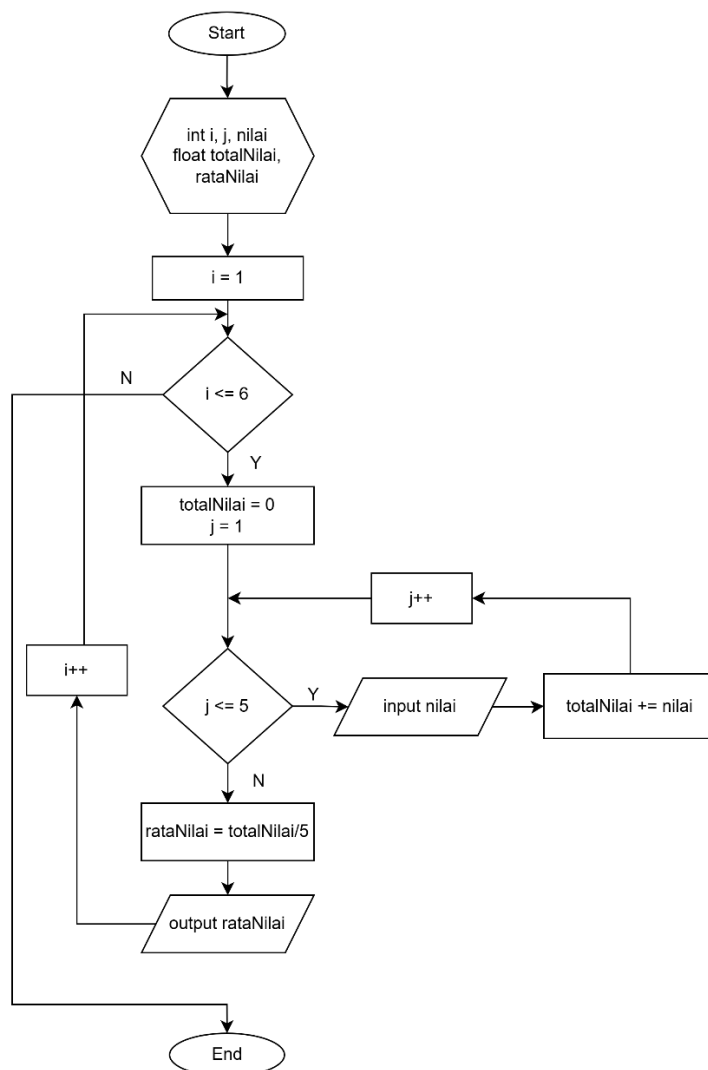
2. Jika tidak sesuai, bagian mana saja yang harus diperbaiki/ditambahkan? Jelaskan setiap bagian yang perlu diperbaiki/ditambahkan.
3. Jelaskan peran masing-masing variabel **i** dan **j** dalam program ini. Mengapa **j** di-set ulang ke 0 di awal setiap iterasi outer loop? Apa yang akan terjadi jika **j** tidak di-reset?

4. Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan 3”

2.4 Percobaan 4: Studi Kasus Nilai Tugas Proyek Kelompok

Waktu Percobaan: 75 menit

Dalam pembelajaran kolaboratif, dosen menerapkan sistem peer review untuk menilai tugas kelompok. Setiap kelompok akan dinilai oleh beberapa kelompok lain sebagai bentuk umpan balik sejawat. Untuk setiap kelompok akan dinilai oleh beberapa kelompok lainnya. Pada percobaan ini, program akan memproses nilai untuk kelas yang berjumlah 6 kelompok. Masing-masing kelompok akan dinilai oleh 5 kelompok yang lain, dalam skala 1-100. Program akan menghitung nilai akhir setiap kelompok sebagai rata-rata dari semua penilai. Berikut adalah flowchart yang menunjukkan algoritma dari penyelesaian masalah tersebut.



Berdasarkan flowchart tersebut, buat program menggunakan bahasa pemrograman Java.

Langkah-langkah Percobaan

1. Buat file baru **NilaiKelompokNoPresensi.java**
2. Buatlah struktur dasar program Java yang terdiri dari fungsi **main()**.

3. Tambahkan library Scanner di bagian atas (luar) class
4. Buat deklarasi Scanner dengan nama variabel **sc** di dalam fungsi main()
5. Deklarasikan variabel sesuai pada flowchart
6. Buat struktur perulangan untuk **inner loop** menggunakan **for** untuk memasukkan 5 nilai dari masing-masing kelompok penilai

```
for (j=1;j<=5;j++) {  
    }  
}
```

7. Di dalam perulangan **for** tersebut, tambahkan perintah untuk memasukkan 5 **nilai** dari kelompok penilai dan menghitung total nilainya. Jangan lupa untuk memberikan nilai awal 0 pada totalNilai sebelum perulangan.

```
totalNilai = 0;  
for (int j = 1; j <= 5; j++) {  
    System.out.print(" Nilai dari Kelompok Penilai " + j + ": ");  
    nilai = sc.nextInt();  
    totalNilai += nilai;  
}
```

8. Setelah proses perulangan memasukkan 5 nilai dan diperoleh total nilai, maka hitung rata-rata nilai dengan rumus $\text{rataNilai} = \text{totalNilai} / 5;$
9. Selanjutnya adalah mengulang proses memasukkan nilai tersebut untuk 6 kelompok. Buat struktur perulangan untuk **outer loop** menggunakan **while**.

```
int i = 1;  
while (i <= 6) {  
    System.out.println("Kelompok " + i);  
  
    totalNilai = 0;  
    for (int j = 1; j <= 5; j++) {  
        System.out.print(" Nilai dari Kelompok Penilai " + j + ": ");  
        nilai = sc.nextInt();  
        totalNilai += nilai;  
    }  
  
    rataNilai = totalNilai / 5;  
    i++;  
}
```

10. Tambahkan narasi keterangan untuk kebutuhan masukan dan luaran
11. Compile dan run program.
12. Amati hasilnya, apakah program telah menjalankan perintah memasukkan 5 nilai untuk 6 kelompok dengan tampilan serupa dengan di bawah ini:

```
Kelompok 1
  Nilai dari Kelompok Penilai 1: 88
  Nilai dari Kelompok Penilai 2: 79
  Nilai dari Kelompok Penilai 3: 90
  Nilai dari Kelompok Penilai 4: 82
  Nilai dari Kelompok Penilai 5: 85
Kelompok 1: nilai rata-rata = 84.8
Kelompok 2
  Nilai dari Kelompok Penilai 1: 89
  Nilai dari Kelompok Penilai 2: 85
  Nilai dari Kelompok Penilai 3: 90
  Nilai dari Kelompok Penilai 4: 85
  Nilai dari Kelompok Penilai 5: 82
Kelompok 2: nilai rata-rata = 86.2
Kelompok 3
  Nilai dari Kelompok Penilai 1: 90
  Nilai dari Kelompok Penilai 2: 91
  Nilai dari Kelompok Penilai 3: 86
  Nilai dari Kelompok Penilai 4: 84
  Nilai dari Kelompok Penilai 5: 90
Kelompok 3: nilai rata-rata = 88.2
Kelompok 4
  Nilai dari Kelompok Penilai 1: 77
  Nilai dari Kelompok Penilai 2: 75
  Nilai dari Kelompok Penilai 3: 80
  Nilai dari Kelompok Penilai 4: 79
  Nilai dari Kelompok Penilai 5: 76
Kelompok 4: nilai rata-rata = 77.4
Kelompok 5
  Nilai dari Kelompok Penilai 1: 80
  Nilai dari Kelompok Penilai 2: 82
  Nilai dari Kelompok Penilai 3: 81
  Nilai dari Kelompok Penilai 4: 77
  Nilai dari Kelompok Penilai 5: 83
Kelompok 5: nilai rata-rata = 80.6
Kelompok 6
  Nilai dari Kelompok Penilai 1: 91
  Nilai dari Kelompok Penilai 2: 90
  Nilai dari Kelompok Penilai 3: 85
  Nilai dari Kelompok Penilai 4: 88
  Nilai dari Kelompok Penilai 5: 90
Kelompok 6: nilai rata-rata = 88.8
```

13. Commit program Anda ke Github dengan pesan "Percobaan 4"

Pertanyaan

1. Jelaskan apa yang terjadi pada variabel **totalNilai** di setiap iterasi *outer loop* dan mengapa inisialisasinya ($total = 0$) berada di dalam *outer loop*, bukan di luar.
2. Modifikasi program di atas, sehingga dapat mencari kelompok dengan rata-rata nilai tertinggi dan tampilkan nomor kelompok tersebut.
3. **Commit dan push hasil modifikasi Anda ke Github dengan pesan "Modifikasi Percobaan 4"**

3. Tugas

Waktu Percobaan : 120 Menit

1. Buatlah program untuk menghitung dan menampilkan jumlah kuadrat bilangan 1 s.d n. Gunakan perulangan bersarang. Berikut output yang diharapkan jika n pada rentang 1 s.d 5.

```
n = 1 → jumlah kuadrat = 1
n = 2 → jumlah kuadrat = 1 + 4 = 5
n = 3 → jumlah kuadrat = 1 + 4 + 9 = 14
n = 4 → jumlah kuadrat = 1 + 4 + 9 + 16 = 30
n = 5 → jumlah kuadrat = 1 + 4 + 9 + 16 + 25 = 55
```

Commit dan push program Anda ke Github dengan pesan “Tugas 1”

2. Buatlah program untuk mencetak tampilan persegi angka seperti di bawah ini berdasarkan input n (nilai n minimal 3). Contoh n = 3, dan n = 5

```

          5 5 5 5 5
          5      5
3 3 3    5      5
3 3 3    5      5
3 3 3    5 5 5 5 5
```

Commit dan push program Anda ke Github dengan pesan “Tugas 2”

3. Sebuah jaringan kafe “Kopi Senja” memiliki beberapa cabang di berbagai lokasi kota. Untuk memantau kinerja operasional harian, manajemen membutuhkan data sederhana mengenai aktivitas penjualan di setiap cabang. Setiap hari, setiap cabang melayani sejumlah pelanggan, dan setiap pelanggan memesan satu atau lebih item, baik berupa makanan maupun minuman. Buatlah program untuk mencatat dan menghitung banyak pelanggan dan total item yang terjual pada masing-masing cabang. Karena sistem yang digunakan masih sederhana dan tidak menyimpan data historis, program hanya perlu mencatat secara langsung berapa banyak pelanggan yang dilayani dan berapa total item yang terjual di masing-masing cabang, tanpa perlu menyimpan nama pelanggan, jenis menu, atau detail lainnya. Program akan meminta input jumlah cabang terlebih dahulu. Lalu untuk setiap cabang, operator memasukkan jumlah pelanggan yang datang hari ini. Selanjutnya, untuk setiap pelanggan, operator memasukkan jumlah item yang dipesan. Program kemudian menghitung dan menampilkan ringkasan penjualan per cabang, serta total keseluruhan dari seluruh cabang pada akhir sesi. Berikut adalah contoh input dan output program:


```
Jumlah cabang kafe: 2

=== Input Penjualan Per Cabang ===

--- Cabang 1 ---
Jumlah pelanggan: 3
- Pelanggan 1 memesan berapa item? 2
- Pelanggan 2 memesan berapa item? 4
- Pelanggan 3 memesan berapa item? 1
Cabang 1:
- Pelanggan: 3 orang
- Item terjual: 7

--- Cabang 2 ---
Jumlah pelanggan: 4
- Pelanggan 1 memesan berapa item? 3
- Pelanggan 2 memesan berapa item? 5
- Pelanggan 3 memesan berapa item? 1
- Pelanggan 4 memesan berapa item? 2
Cabang 2:
- Pelanggan: 4 orang
- Item terjual: 11

Total seluruh Cabang:
Pelanggan: 7 orang
Item terjual: 18 item
```

Commit dan push program Anda ke Github dengan pesan "Tugas 3"