

## 2.1 Pertanyaan

1. Jika pada perulangan for, inisialisasi  $i=1$  diubah menjadi  $i=0$ , apa akibatnya? Mengapa bisa demikian?

Bintang "\*" akan tercetak sebanyak  $n+1$ , karena dalam perulanganya memiliki kondisi true sebanyak  $n+1$ .

2. Jika pada perulangan for, kondisi  $i \leq n$  diubah menjadi  $i < n$ , bagaimana bentuk outputnya jika input  $n = 5$ ? Mengapa hasilnya berbeda?

Outputnya Adalah "\*\*\*\*", hasilnya berbeda karena dalam perulangan nya memiliki kondisi true sebanyak 4.

3. Jika pada perulangan for, kondisi  $i \leq n$  diubah menjadi  $i > n$ , apa akibatnya? Mengapa bisa demikian?

Tidak ada Bintang yang tercetak, karena kondisi akan selalu false.

4. Jika pada perulangan for, kondisi step  $i++$  diubah menjadi  $i--$  apa akibatnya? Mengapa bisa demikian?

Pada output akan tercetak Bintang terus menerus atau terjadi infinite loop, karena kondisinya selalu true.

5. Jika pada perulangan for, step  $i++$  diubah menjadi  $i += 2$ , bagaimana pola outputnya jika input  $n = 6$ ? Apa yang menyebabkan perubahan tersebut?

Bintang yang tercetak akan sebanyak 3, karena setelah true variable  $i$  di tambah 2 sehingga akan memiliki kondisi true pada saat nilai variable  $i = 1, 3, 5$  dan akan memiliki kondisi false saat kondisi  $i = 7$ . Sehingga akan tercetak Bintang sebanyak 3.

## 2.2 Pertanyaan

1. Perhatikan perulangan luar. Jika pada sintaks for, inisialisasi  $iOuter=1$  diubah menjadi  $iOuter=0$ , apa yang akibatnya? Mengapa bisa demikian?

Baris Bintang yang tercetak akan sebanyak  $n+1$ , karena kondisi true pada outer loop sebanyak  $n+1$

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?

Kolom Bintang yang tercetak akan sebanyak  $n+1$ , karena kondisi `true` pada inner loop sebanyak  $n+1$ .

3. Apakah perbedaan kegunaan antara perulangan luar dengan perulangan yang berada di dalamnya?

Pada algoritmanya akan melakukan perulangan luar lalu melakukan perulangan dalam. Sehingga perulangan dalam akan diulang terus menerus hingga perulangan luar kondisinya menjadi `false`.

4. Mengapa perlu ditambahkan sintaks `System.out.println();` di bawah perulangan dalam? Apa akibatnya jika sintaks tersebut dihilangkan?

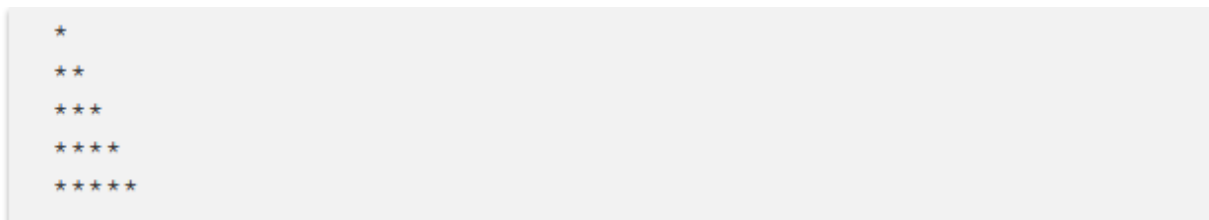
Agar Bintang yang tercetak menjadi baris dan kolom, apabila dihilangkan akan tercetak Bintang sebanyak  $n$  pangkat  $n$  dalam 1 baris.

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



## 2.3 Pertanyaan

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



Tidak

2. Jika tidak sesuai, bagian mana saja yang harus diperbaiki/ditambahkan? Jelaskan setiap bagian yang perlu diperbaiki/ditambahkan?

Perlu tambahan program “System.out.println();” di antara loop 1 dan 2

```
while (i <= n) {  
    int j = 0;  
    while (j < i) {  
        System.out.print(s: "*");  
        j++;  
    }  
    System.out.println();  
    i++;  
}
```

Atau

```
while (i <= n) {  
    int j = 0;  
    System.out.println();  
    while (j < i) {  
        System.out.print(s: "*");  
        j++;  
    }  
    i++;  
}
```

Program tersebut berfungsi untuk memberikan enter(next line) pada saat inner loop berhenti/false.

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?

i berfungsi sebagai kondisi outer loop dan j berfungsi sebagai kondisi pada inner loop. Variable j di set ulang ke 0 agar nilai nya tidak melebihi n. apabila tidak di set Kembali output yang tampil akan menjadi Bintang pada 1 baris sebanyak

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

Percobaan 3”

## 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.

Nilai variable totalNilai akan terus ditambahkan dengan variable nilai. Agar setelah inner loop selesai/false totalNilai menjadi 0 kembali sehingga untuk iterasi loop selanjutnya total nilai dari iterasi loop sebelumnya ter reset.

2. Modifikasi program di atas, sehingga dapat mencari kelompok dengan rata-rata nilai tertinggi dan tampilkan nomor kelompok tersebut.

```
1  import java.util.Scanner;
2
3  public class NilaiKelompok04 {
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6          int nilai;
7          float totalNilai, rataNilai, a = 0;
8          int i = 1;
9          String b = "";
10         while (i <= 6) {
11             System.out.println("Kelompok " + i);
12
13             totalNilai = 0;
14             for (int j = 1; j <= 5; j++) {
15                 System.out.print(" Nilai dari Kelompok Penilai " + j + ": ");
16                 nilai = sc.nextInt();
17                 totalNilai += nilai;
18             }
19
20             rataNilai = totalNilai / 5;
21             if (a < rataNilai) {
22                 a = rataNilai;
23                 b = "Nilai terbaik adalah nilai kelompok " + i + " dengan rata rata nilai " + a;
24             }
25             System.out.println("Kelompok " + i + " Nilai Rata-Rata " + rataNilai);
26             i++;
27         }
28         System.out.println(b);
29     }
30 }
31
32
33
```

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

Percobaan 4”

