

**Laporan Jobsheet 8**

**Percobaan 1**

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

**Jawaban :**

Maka perulangan akan dimulai dari angka 0 dan dimulai dari 0 dan angka 0 terhitung. Jadi jika memasukkan  $n=5$ , maka outputnya akan nada 6.

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

**Jawaban :**

maka yang akan keluar menjadi outputnya kurang dari angka yang kita inputkan. Misal input angka 5, maka yang akan di keluarkan outputnya berjumlah kurang dari 5 karena tanda “=” sudah diubah.

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

**Jawaban :**

Maka tidak akan menghasilkan output apapun. Karena syaratnya diubah menjadi  $i > n$ , sedangkan  $i=1$ , maka pengecekan syarat  $i > n$  Ketika  $n$  lebih dari 1 atau 1 maka akan terjadi false.

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

**Jawaban :**

Maka loop tidak akan berhenti, karena Ketika diubah menjadi  $i--$  maka akan membuat nilai  $i$  semakin kecil atau mundur dan jauh dari batas “ $n$ ”.

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

**Jawaban :**

setiap kali satu putaran selesai, nilai  $i$  akan ditambah 2, bukan ditambah 1 seperti biasanya ( $i++$ ). Akibatnya,  $i$  "melompat" lebih cepat menuju batas  $n$ , sehingga jumlah perulangannya jadi lebih sedikit.

## Percobaan 2

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

**Jawaban:**

Jumlah **baris** bintang akan bertambah satu. Karena **iOuter < N**, jika memasukkan **N= 5**, program akan mencetak **5 baris** ke bawah, bukan 4.

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?

**Jawaban :**

Jumlah bintang ke samping (kolom) akan bertambah satu di setiap barisnya. Jika **N = 5**, maka setiap baris akan memiliki 6 bintang

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

**Jawaban :**

- **Perulangan Luar (iOuter):** Bertanggung jawab untuk mengontrol BARIS (arah vertikal), menentukan berapa kali proses pencetakan satu baris penuh akan diulang.
- **Perulangan Dalam (i):** Bertanggung jawab untuk mengontrol KOLOM (arah horizontal), bertugas mencetak isi konten (dalam hal ini tanda bintang \*) berjejer ke samping dalam satu baris tersebut.

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

**Jawaban :**

Perintah **System.out.println();** (tanpa isi) berfungsi sebagai **Enter** atau pindah baris, dipanggil setiap kali *inner loop* selesai mencetak satu deret bintang.

### Percobaan 3

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

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

**Jawaban :**

Tidak Sesuai

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

**Jawaban :**

**Menambahkan System.out.println();** Setelah *inner loop* (while ( $j < i$ )) selesai mencetak deretan bintang untuk satu baris, syntax tersebut berfungsi untuk memerintahkan komputer untuk turun ke baris berikutnya sebelum *outer loop* melanjutkan ke angka  $i$  selanjutnya.

**Mengubah inisialisasi i Di code int i = 0.**

- Saat  $i=0$ , *inner loop* tidak jalan (karena  $0 < 0$  itu False), tapi baris kosong tetap tecetak (jika sudah ada println).
- Agar pas 5 tingkat yang terlihat jelas,  $i$  dimulai dari 1.

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?

**Jawaban :**

- **Variabel i (Outer Loop):** Berperan sebagai **penghitung baris** (vertikal), menentukan sedang berada di baris ke berapa dan juga menjadi **batas jumlah bintang** yang boleh dicetak pada baris tersebut.
- **Variabel j (Inner Loop):** Berperan sebagai **penghitung bintang per baris** (horizontal), bertugas menghitung berapa bintang yang *sedang* dicetak di baris aktif saat itu.

Mengapa  $j$  di-set ulang ke 0 di awal setiap iterasi Outer Loop, Karena setiap kali pindah ke baris baru (baris bawahnya), maka harus mulai mencetak bintang **dari awal lagi (dari kiri)**.

## Percobaan 4

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.

**Jawaban :**

Pada setiap iterasi *outer loop* (setiap kali program pindah ke "Kelompok" baru), nilai variabel totalNilai akan di-reset (dikembalikan) menjadi 0. Mengapa harus di dalam *outer loop*, agar perhitungan total nilai untuk setiap kelompok berdiri sendiri dan tidak bercampur dengan nilai dari kelompok sebelumnya.

Kelompok 1: totalNilai mulai dari 0 -> ditambah nilai-nilai kelompok 1.  
Kelompok 2: totalNilai di-reset jadi 0 lagi -> baru ditambah nilai-nilai kelompok 2. (Benar).

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

**Jawaban :**

```
2 import java.util.Scanner;
3
4 public class NilaiKelompok9 {
5     Run main | Debug main | Run | Debug
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);    Convert to try-with-resources
8         int i=1, nilai;
9         float totalNilai, rataNilai;
10
11         float nilaiTertinggi = 0;
12         int kelompokTerbaik = 0;
13         while (i <=6) {
14             System.out.println("Kelompok " + i);
15
16             totalNilai = 0;
17             for (int j =1; j<=5; j++) {
18                 System.out.print("Nilai dari kelompok Penilai " + j + ":");
19                 nilai = sc.nextInt();
20                 totalNilai += nilai;
21             }
22
23             rataNilai = totalNilai / 5;
24             System.out.println("Nilai Rata-Rata" + rataNilai);
25
26             if (rataNilai > nilaiTertinggi) {
27                 nilaiTertinggi = rataNilai;
28                 kelompokTerbaik = i;
29             }
30             i++;
31         }
32         System.out.println("-----");
33         System.out.println("Nilai rata-rata tertinggi diperoleh oleh kelompok" + kelompokTerbaik);
34         System.out.println("Dengan skor rata-rata: " +nilaiTertinggi);
35
36         sc.close();
37     }
38 }
```

## Tugas

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”**

### Jawaban :

```
1 import java.util.Scanner;
2 public class Tugas1Jobsheet8 {
3     Run main | Debug main | Run | Debug
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);    Resource leak:
6
7         System.out.println("Masukkan Nilai n:");
8         int n = sc.nextInt();
9
10        for (int i =1; i <=n; i++) {
11            int total =0;
12            System.out.print("n = "+i + "jumlah kuadrat =");
13
14            for (int j=1; j<=i; j++ ) {
15                int kuadrat = j*j;
16                total += kuadrat;
17                System.out.print(kuadrat);
18                if ( j < i) {
19                    System.out.print(" +");
20                }
21            }
22            System.out.println(" = " + total);
23        }
24    }
25 }
```

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

Commit dan push program Anda ke Github dengan pesan "Tugas 2"

**Jawaban :**

```
import java.util.Scanner;  
  
public class Tugas2Jobsheet8 {  
    Run main | Debug main | Run | Debug  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  Resource leak:  
  
        System.out.println("Masukkan nilai n ( minimal 3):");  
        int n = sc.nextInt();  
  
        if ( n < 3 ) {  
            System.out.println("Nilai n minimal 3 !");  
        }else{  
  
            for ( int i = 1; i <= n; i++ ) {  
  
                for (int j = 1; j <= n; j++) {  
  
                    if ( i == 1 || i == n || j == 1 || j == n ) {  
                        System.out.print(n + "");  
                    }else {  
                        System.out.print(" ");  
                    }  
                }  
                System.out.println();  
            }  
        }  
    }  
}
```

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"

### Jawaban :

```

import java.util.Scanner;

public class Tugas3Jobsheet8 {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);    Resource leak: 'sc' is never closed

        System.out.print("Jumlah Cabang kafe"+ ":");

        int jumlahCabang = sc.nextInt();

        System.out.println(x: "==== Input Penjualan Per Cabang =====");

        int totalPelangganSemuaCabang = 0;
        int totalSemuaItem = 0;

        for (int i =1; i<=jumlahCabang; i++) {
            System.out.print("\n--- Cabang "+ i + " ---");

            System.out.print(s: "Jumlah Pelanggan: ");
            int jumlahPelanggan = sc.nextInt();

            int totalItemCabang =0;

            for ( int j = 1; j<=jumlahPelanggan; j++) {

                System.out.print(" Pelanggan "+ j + " memesan berapa item? ");
                int jumlahItem = sc.nextInt();

                totalItemCabang += jumlahItem;
            }

            System.out.println("Cabang " + i + ":");

            System.out.println("- Pelanggan: " + jumlahPelanggan + "orang");
            System.out.println("- Item terjual: " + totalItemCabang+ " item");

            totalPelangganSemuaCabang += jumlahPelanggan;
            totalSemuaItem += totalItemCabang;
        }

        System.out.println(x: "\n=====");
        System.out.println(x: " Total seluruh Cabang: ");
        System.out.println("Pelanggan : " + totalPelangganSemuaCabang+ " orang");
        System.out.println(" Item terjual : " + totalSemuaItem + " item");
    }
}

```