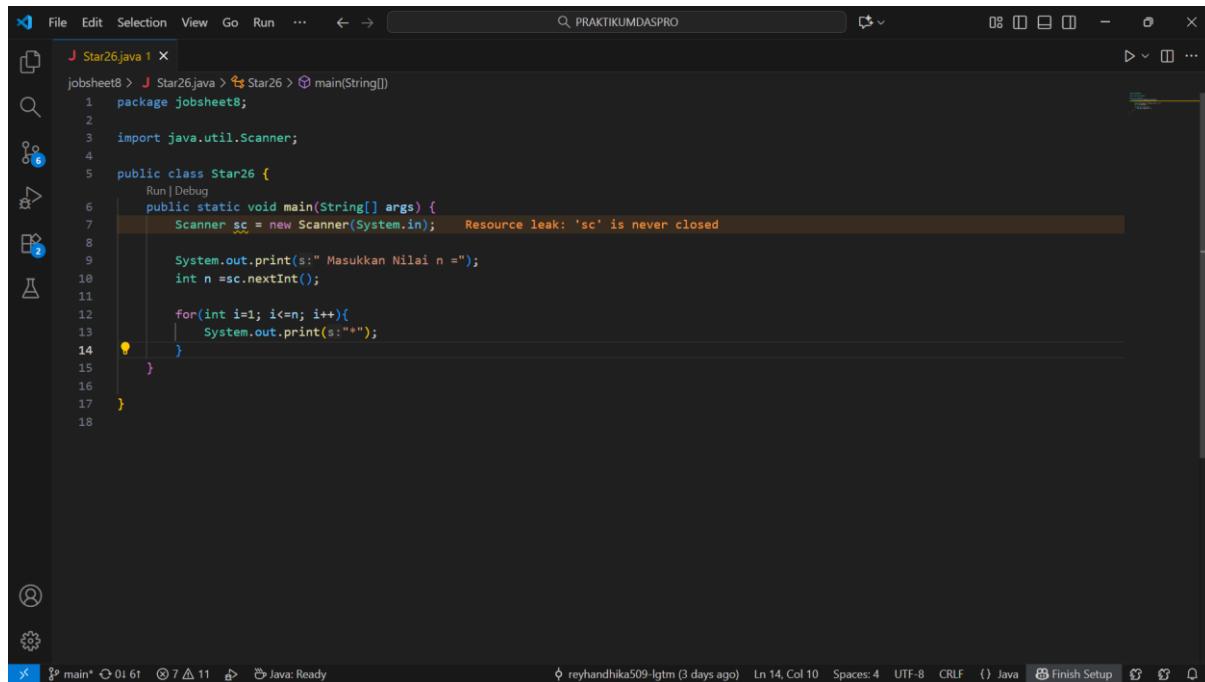


**NAMA** : REYHANDHIKA ZIKRI PRIJADI  
**KELAS/NO'ABSEN** : TI\_1G / 26  
**NIM** : 254107020219

## PERCOBAAN 1



A screenshot of a Java code editor (IntelliJ IDEA) showing a code snippet named Star26.java. The code reads a value from System.in using Scanner, prints a prompt, reads the value, and then prints a series of asterisks. A warning message 'Resource leak: 'sc' is never closed' is displayed in the gutter next to the scanner declaration. The code editor interface includes toolbars, a search bar, and a status bar at the bottom.

```
File Edit Selection View Go Run ... ← → PRAKTIKUMDASPRO
J Star26.java 1 ×
jobsheet8 > J Star26.java > Star26 > main(String[])
1 package jobsheet8;
2
3 import java.util.Scanner;
4
5 public class Star26 {
6     Run | Debug
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in); Resource leak: 'sc' is never closed
9         System.out.print("Masukkan Nilai n =");
10        int n = sc.nextInt();
11
12        for(int i=1; i<=n; i++){
13            System.out.print("*");
14        }
15    }
16
17 }
```

### PERTANYAAN 1 :

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

Mengapa bisa demikian?

↳ Akibatnya jumlah Bintang yang dicetak akan bertambah Satu.

Hal ini terjadi karena perulangan dimulai dari nilai 0 , sehingga proses perulangan berjalan satu kali lebih lama dibandingkan Ketika dimulai dari 1 dengan kata lain , terdapat satu iterasi tambahan.

Contohnya jika  $n = 5$ , maka hasil yang semula \*\*\*\*\* akan menjadi \*\*\*\*\*.

2. Jika pada perulangan for, kondisi  $i \leq n$  diubah menjadi  $i < n$ , bagaimana bentuk

outputnya jika input  $n = 5$ ? Mengapa hasilnya berbeda?

↳ Akibatnya, jumlah Bintang yang dicetak akan berkurang satu.

Hal ini dikarenakan perulangan berhenti saat nilai  $i$  sudah mencapai  $n$ , sehingga perulangan hanya berjalan sampai  $n - 1$ . jadi jumlah iterasi menjadi lebih sedikit satu kali dibandingkan sebelumnya.

Contohnya jika  $n = 5$ , maka hasil yang semula \*\*\*\*\* akan menjadi \*\*\*.

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

Mengapa bisa demikian?

↳ Akibatnya tidak ada output yang dihasilkan. Hal ini terjadi dikarenakan pada saat perulangan dimulai, dari nilai awal yaitu 1 tidak memenuhi kondisi  $i > n$ . karena kondisi tersebut langsung bernilai salah, maka perulangan tidak dijalankan sama sekali.

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

Mengapa bisa demikian?

↳ Akibatnya, program akan mengalami perulangan tak hingga ( infinite loop). Hal ini dikarenakan nilai  $i$  justru terus berkurang,  $i$  selalu lebih kecil dari  $n$ .

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

↳ Akibatnya jumlah Bintang yang dicetak akan berkurang sekitar setengahnya. Hal ini dikarenakan nilai  $i$  bertambah dua setiap kali perulangan dijalankan, sehingga jumlah perulangan menjadi sedikit dibandingkan jika  $i$  bertambah satu .

Contohnya jika  $n = 6$ , maka perulangan akan dijalankan untuk  $i = 1,3,5$  sehingga hasilnya muncul hanya \*\*\*

## PERCOBAAN 2

```
File Edit Selection View Go Run ... ← → PRACTIKUMDASPRO
J Star26.java 1 J Square26.java 1, M ...
jobsheet8 > J Square26.java > Square26 > main(String[])
1 package jobsheet8;
2
3 import java.util.Scanner;
4
5 public class Square26 {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);    Resource leak: 'sc' is never closed
8         System.out.print("Masukkan Nilai N = ");
9         int N = sc.nextInt();
10
11         for(int iOuter=1; iOuter<=N; iOuter++){
12             for(int i=1; i<=N; i++){
13                 System.out.print("*");
14             }
15             System.out.println();
16         }
17     }
18 }
19
20
```

## Pertanyaan 2

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

↳ Apabila inisialisasi  $iOuter$  di ubah dari 1 menjadi 0, maka jumlah baris Bintang yang dihasilkan akan bertambah satu baris. Hal ini dikarenakan perulangan dimulai dari angka 0 hingga  $N$ , sehingga total perulangannya menjadi ( $N+1$ ) kali. Dengan kata lain, program mencetak satu barisan tambahan dibandingkan kondisi semula.

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?

↳ Apabila inisialisasi  $i$  outer di ubah dari 1 menjadi 0, maka Setiap baris akan berisi satu Bintang lebih banyak, hal ini disebabkan karena perulangan dalam akan berjalan satu kali lebih lama di setiap perulangan, sebab dimulai dari 0, bukan dari 1. Akibatnya, jumlah kolom Bintang dalam setiap baris bertambah satu.

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

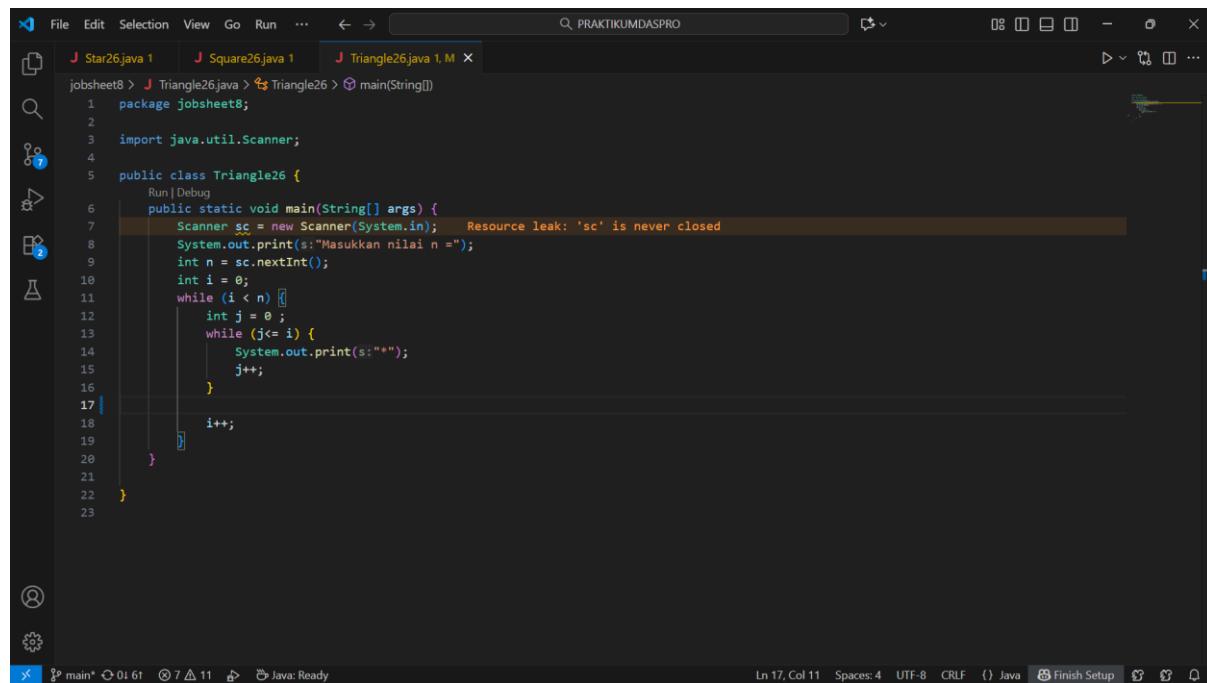
↳ perulangan luar (outer loop) berfungsi untuk mengatur jumlah baris Bintang yang akan dicetak. Sedangkan perulangan dalam (inner loop) berfungsi untuk mencetak jumlah Bintang di setiap baris tersebut. Dengan kata lain, outer loop mengatur vertikal (atas- bawah), sementara inner loop mengatur horizontal (kiri-kanan).

4. Mengapa perlu ditambahkan sintaks `System.out.println();` di bawah

perulangan dalam? Apa akibatnya jika sintaks tersebut dihilangkan?

↳ Sintaks `System.out.println();` berfungsi untuk memindahkan pencetakan ke baris berikutnya setelah satu baris Bintang selesai dicetak. Apabila sintaks tersebut dihilangkan, seluruh Bintang akan tercetak dalam satu baris Panjang tanpa jeda baris baru.

### PERCOBAAN 3



The screenshot shows an IDE interface with a dark theme. In the center is a code editor window displaying Java code for a class named `Triangle26`. The code uses nested loops to print a triangle of asterisks. A tooltip or error message is visible above the code, stating "Resource leak: 'sc' is never closed". The code editor has tabs for other files like `Star26.java` and `Square26.java`. At the bottom, there are status bars showing file names, line numbers, and Java setup information.

```
1 package jobsheet8;
2
3 import java.util.Scanner;
4
5 public class Triangle26 {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in); Resource leak: 'sc' is never closed
8         System.out.print("Masukkan nilai n =");
9         int n = sc.nextInt();
10        int i = 0;
11        while (i < n) {
12            int j = 0;
13            while (j <= i) {
14                System.out.print("*");
15                j++;
16            }
17            i++;
18        }
19    }
20 }
21
22 }
```

### PERTANYAAN 3

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

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

↳ Output yang dihasilkan dari program **belum sesuai** dengan tampilan yang diharapkan. Hal ini disebabkan karena di dalam program belum terdapat perintah untuk **berpindah ke baris baru** setelah satu baris bintang selesai dicetak. Akibatnya, semua bintang tercetak dalam satu baris panjang.

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

setiap bagian yang perlu diperbaiki/ditambahkan.

↳ Agar sesuai dengan tampilan yang diharapkan, program perlu ditambahkan perintah System.out.println(); setelah perulangan dalam (while ( $j \leq i$ )) selesai. Perintah ini berfungsi untuk memindahkan kursor ke baris berikutnya, sehingga baris Bintang berikutnya dicetak dibawahnya, bukan disamping

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?

↳ - **Variabel i** berfungsi sebagai **pengatur jumlah baris** yang akan dicetak.

Nilai i akan bertambah setiap kali satu baris selesai dicetak.

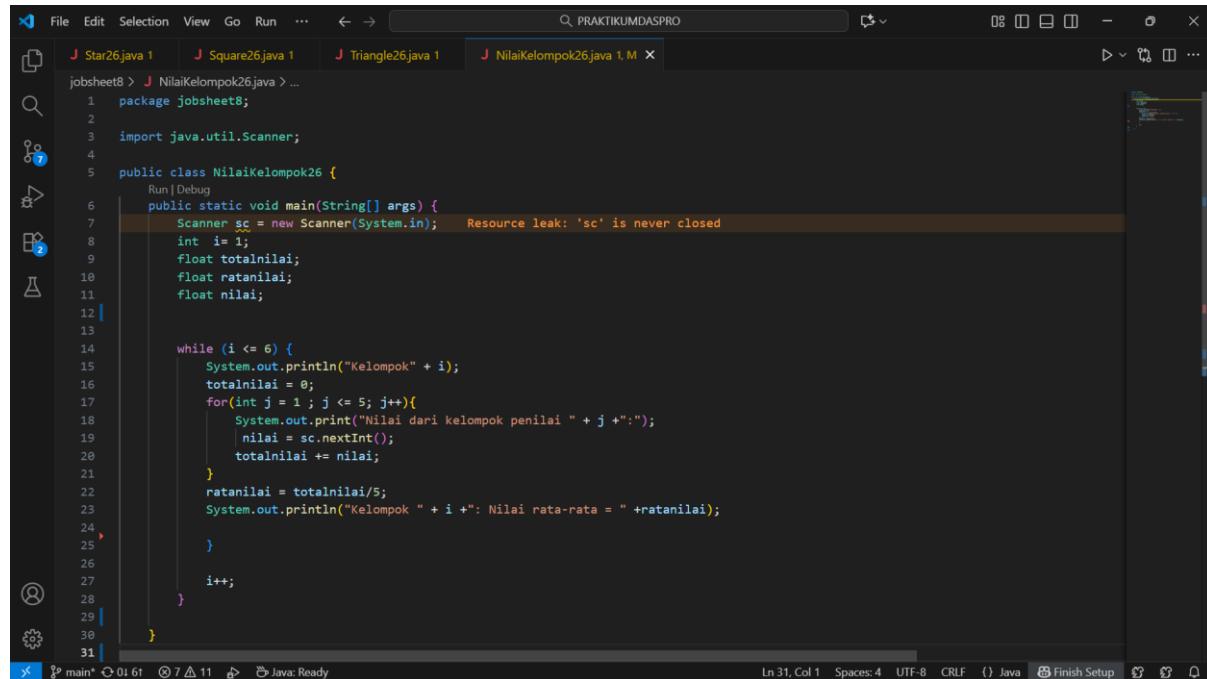
- **Variabel j** berfungsi sebagai **pengatur jumlah bintang di setiap baris**.

Nilai j dimulai dari 0 di setiap iterasi i dan bertambah hingga mencapai i, sehingga jumlah bintang di setiap baris meningkat satu per satu.

- **j harus di-set ulang ke 0 di awal setiap iterasi outer loop**, karena perulangan dalam harus selalu dimulai dari awal untuk setiap baris baru.

Jika j **tidak di-reset**, maka nilai j akan terus bertambah dari baris sebelumnya, dan akibatnya bintang akan terus bertambah tanpa batas — bentuk segitiga yang diinginkan tidak akan terbentuk.

#### PERCOBAAN 4



The screenshot shows an IDE interface with a dark theme. The top menu includes File, Edit, Selection, View, Go, Run, etc. A tab bar at the top has tabs for Star26.java, Square26.java, Triangle26.java, and NilaiKelompok26.java. The code editor contains the following Java code:

```
1 package jobsheet8;
2
3 import java.util.Scanner;
4
5 public class NilaiKelompok26 {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);    Resource leak: 'sc' is never closed
8         int i = 1;
9         float totalnilai;
10        float ratanilai;
11        float nilai;
12
13
14        while (i <= 6) {
15            System.out.println("Kelompok" + i);
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            ratanilai = totalnilai/5;
23            System.out.println("Kelompok " + i +": Nilai rata-rata = " +ratanilai);
24
25        }
26
27        i++;
28    }
29
30}
31
```

The code calculates the average mark for each group (kelompok) based on 5 student inputs. It uses a nested loop where the inner loop (j) runs 5 times for each group (i), and the outer loop (i) runs 6 times. The variable 'sc' is declared but never closed, which is highlighted by a warning message in the code editor.

#### PERTANYAAN 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.

↳ variable totalnilai digunakan untuk menampung hasil penjumlahan seluruh nilai dalam satu kelompok. Pada setiap iterasi outer loop(perulangan untuk tiap kelompok ) nilai totalnilai akan di hitung ulang mulai dari nol. Karena setiap kelompok memiliki hasil perhitungan yang terpisah, maka inisialisasi totalnilai = 0 ditempatkan didalam outer loop. Tujuannya agar setiap kali program berpindah ke kelompok berikutnya, perhitungan jumlah nilai dimulai dari awal lagi. Apabila inisialisasi totalNilai diletakkan di luar outer loop, maka nilai dari kelompok sebelumnya akan ikut terbawa dan dijumlahkan dengan kelompok berikutnya. Akibatnya, hasil rata-rata menjadi tidak akurat karena data antar kelompok saling tercampur.

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

The screenshot shows a Java code editor in VS Code. The code is named `NilaiKelompok26.java`. It reads five groups of numbers from the console, calculates their averages, and prints the group number with the highest average. The code uses a for loop to iterate through the groups, a variable `ratanilai` to store the current average, and a variable `kelompokterbaik` to store the group number with the highest average found so far. The output window shows the input values and the resulting output.

```
public class NilaiKelompok26 {
    public static void main(String[] args) {
        int ratanilai = totalnilai/5;
        System.out.println("Kelompok" + i +": Nilai rata-rata = " +ratanilai);

        if (ratanilai > nilairatatertinggi ){
            nilairatatertinggi = ratanilai ;
            kelompokterbaik = i;
        }

        i++;
    }

    System.out.println(x: "\n== HASIL TERBAIK ==");
    System.out.println("Kelompok dengan nilai rata-rata tertinggi: kelompok" + kelompokterbaik);
    System.out.println("Dengan nilai : " + nilairatatertinggi);
}

Kelompok6
Nilai dari kelompok penilai 1:70
Nilai dari kelompok penilai 2:70
Nilai dari kelompok penilai 3:80
Nilai dari kelompok penilai 4:80
Nilai dari kelompok penilai 5:90
Kelompok6: Nilai rata-rata = 78.0

== HASIL TERBAIK ==
Kelompok dengan nilai rata-rata tertinggi: kelompok5
Dengan nilai : 90.0
PS C:\Users\Reyhandhika Zikri P\OneDrive\Documents\PRAKTIKUMDASPRO>
```

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