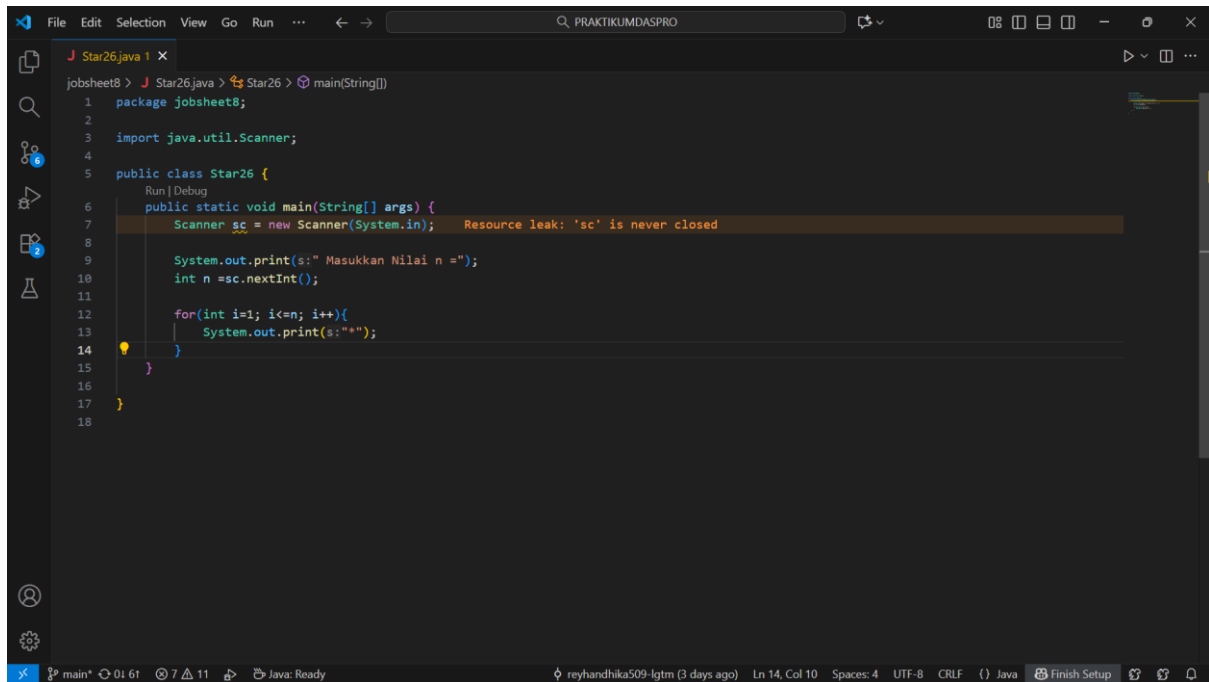


NAMA : REYHANDHIKA ZIKRI PRIJADI

KELAS/NO'ABSEN : TI_1G / 26

NIM : 254107020219

PERCOBAAN 1



The screenshot shows an IDE window titled 'PRAKTIKUMDASPRO'. The code is in a file named 'Star26.java' and contains the following Java code:

```
1 package jobsheet8;
2
3 import java.util.Scanner;
4
5 public class Star26 {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8
9         System.out.print(s: " Masukkan Nilai n =");
10        int n = sc.nextInt();
11
12        for(int i=1; i<=n; i++){
13            System.out.print(s: "");
14        }
15    }
16 }
17
18 }
```

A red error message is displayed on line 7: 'Resource leak: 'sc' is never closed'. The IDE interface includes a sidebar with icons for Explorer, Search, Run and Debug, and a bottom status bar showing 'main' at line 61, column 11, with 7 errors and 11 warnings.

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<=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 <= 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 .

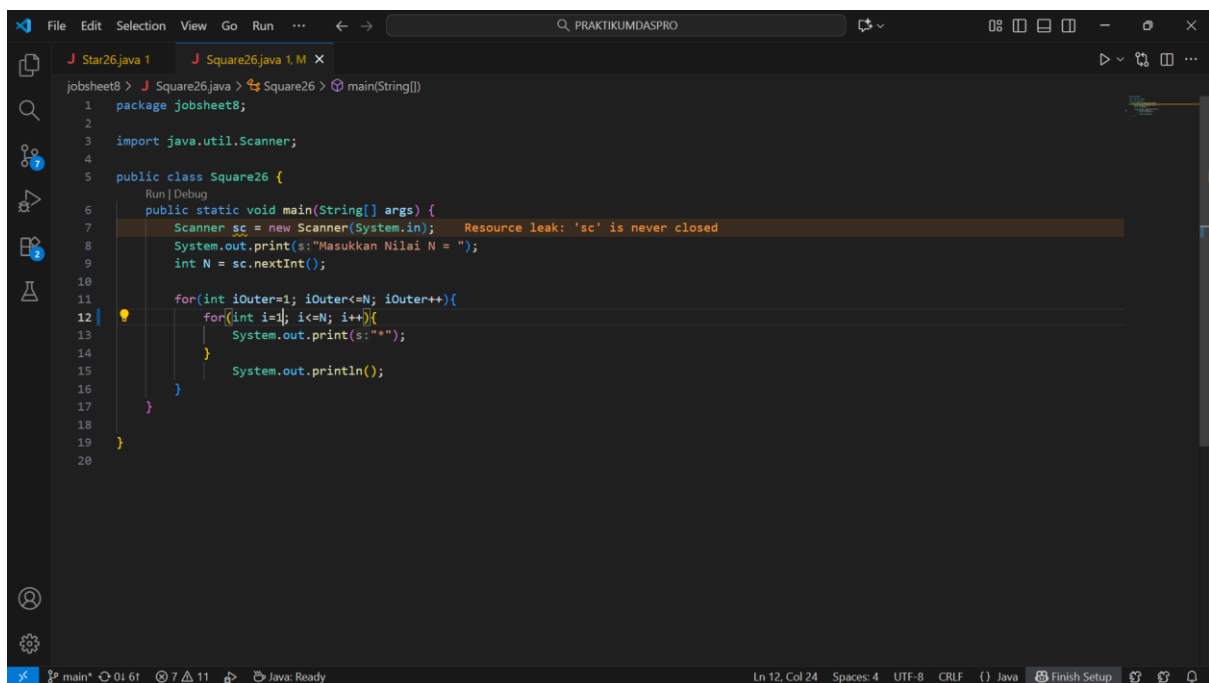
5. Jika pada perulangan for, step $i++$ diubah menjadi $i += 2$, bagaimana pola

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



```
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);
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(s:"");
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 iOuter 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?

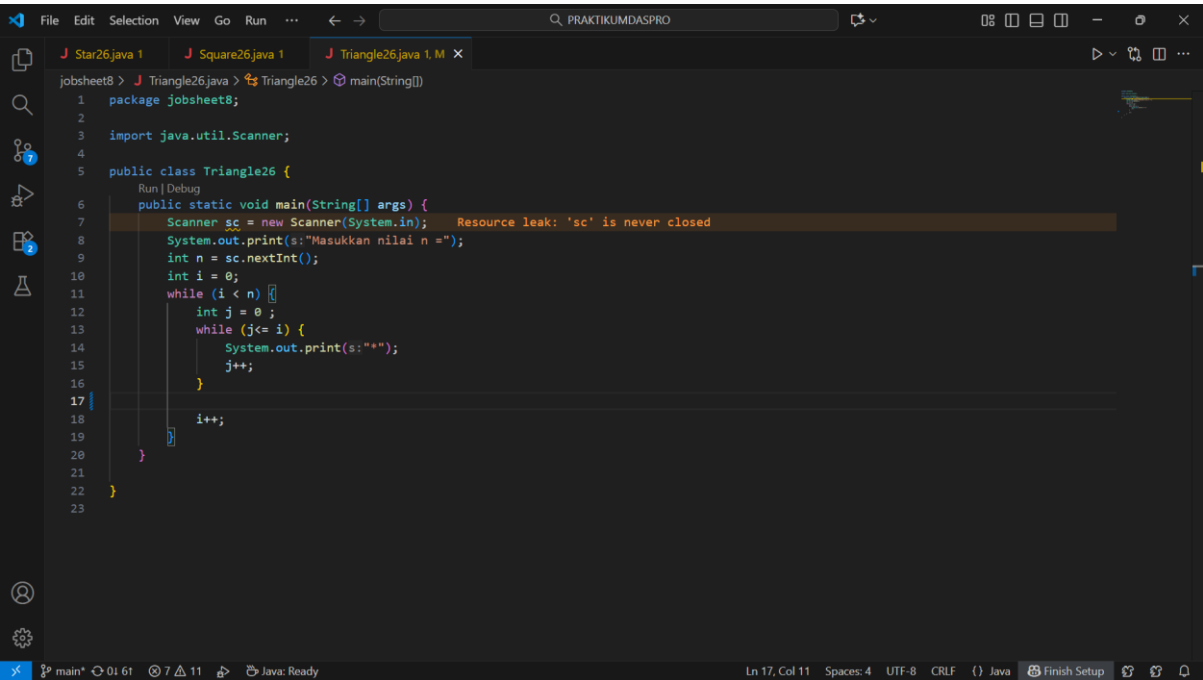
↳ perulangan luar (outer loop) berfungsi untuk mengatur jumlah baris Bintang yang akan dicetak. Sedangkan perulangan dalam (inner loop) berfungsi untuk mencetak jumlah Bintang di setiap baris tersebut. Dengan kata lain , outer loop mengatur vertikal (atas- bawah), sementara inner loop mengatur horizontal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

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

perulangan dalam? Apa akibatnya jika sintaks tersebut dihilangkan?

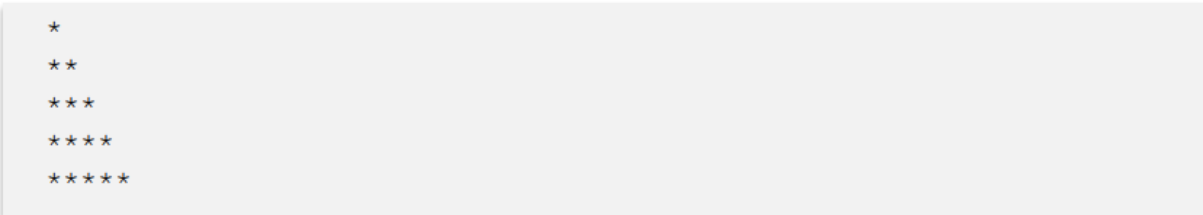
↳ Sintaks System.out.println(); berfungsi untuk memindahkan pencetakan ke baris berikutnya setelah satu baris Bintang selesai dicetak. Apabila sintaks tersebut dihilangkan, seluruh Bintang akan tercetak dalam satu baris Panjang tanpa jeda baris baru.

PERCOBAAN 3



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 <= 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.

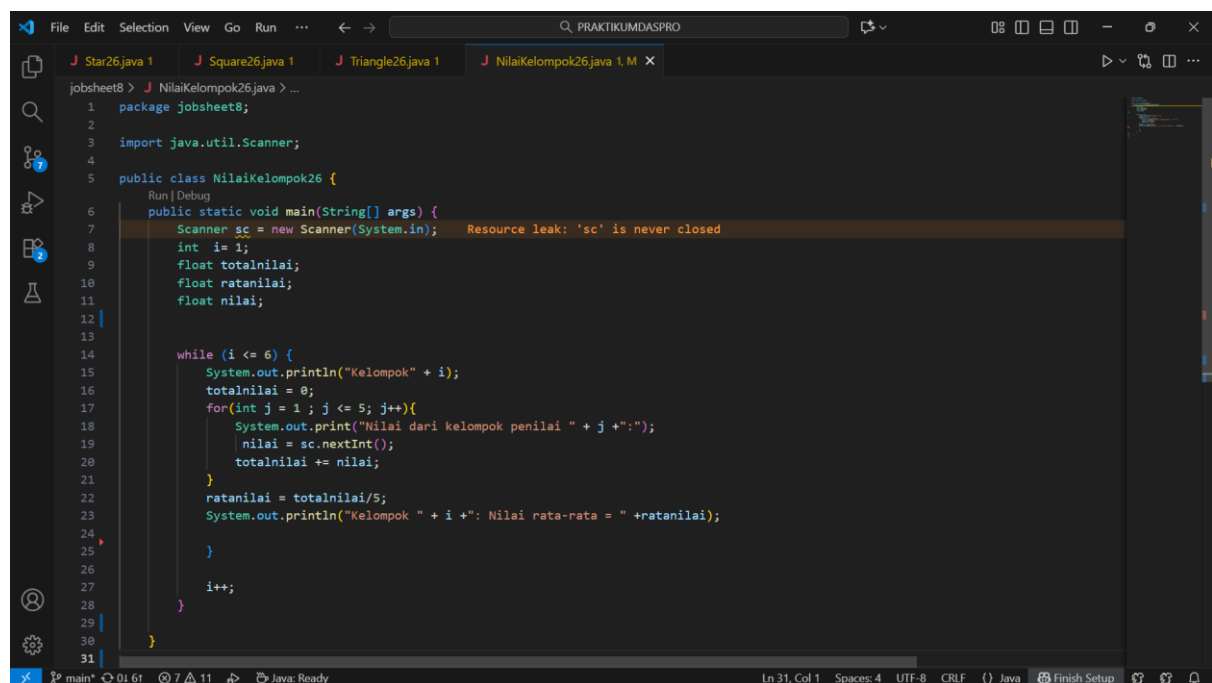
- **Variabel `j`** berfungsi sebagai **pengatur jumlah bintang di setiap baris**.

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.

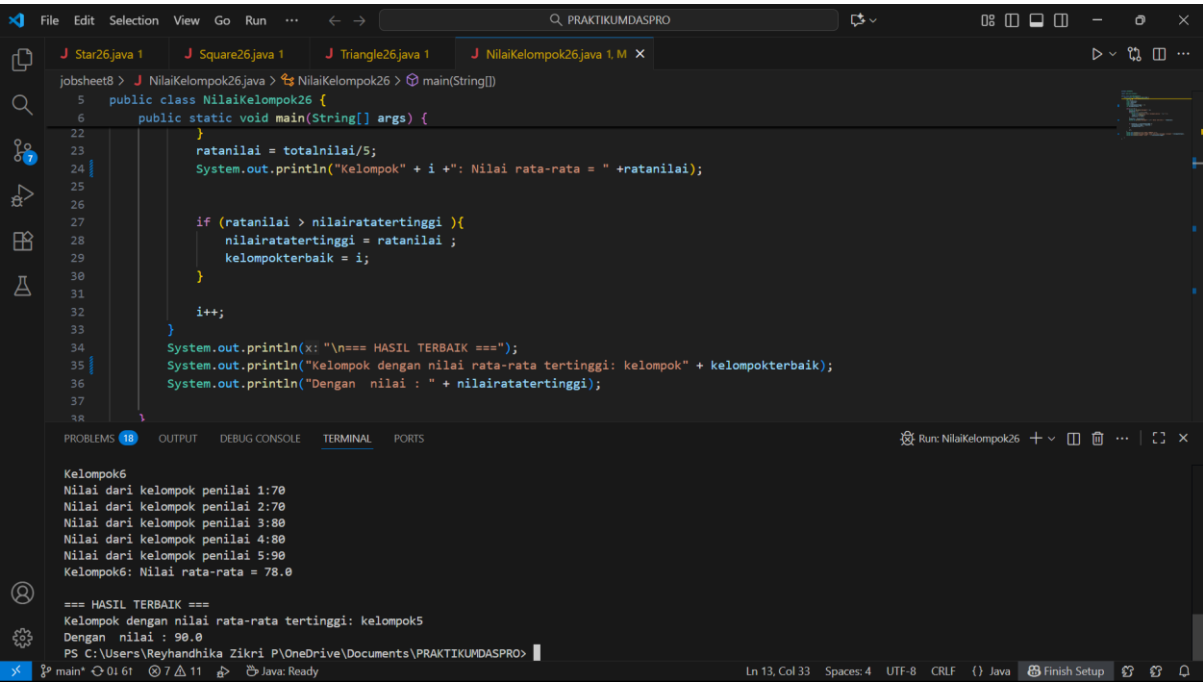
PERCOBAAN 4



```
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);
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    i++;
27 }
28
29
30
31 }
```

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.



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

Percobaan 4”