

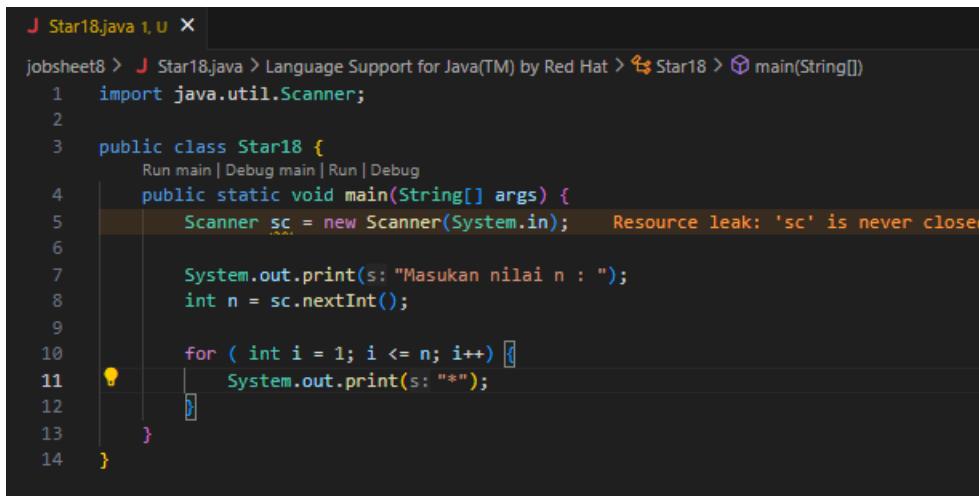
Nama : Muhammad Toriq Januarsyah (18)

kelas : 1-H

NIM : 254107020075

Mata Kuliah : Praktikum Dasar Pemograman (Jobsheet8)

2.1 Percobaan 1



```
J Star18.java 1.0 X
jobsheet8 > J Star18.java > Language Support for Java(TM) by Red Hat > ⚙ Star18 > main(String[])
1 import java.util.Scanner;
2
3 public class Star18 {
4     Run main | Debug main | Run | Debug
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);    Resource leak: 'sc' is never closed
7         System.out.print(s: "Masukan nilai n : ");
8         int n = sc.nextInt();
9
10        for ( int i = 1; i <= n; i++) {
11            System.out.print(s: "*");
12        }
13    }
14 }
```

```
Masukan nilai n : 5
*****
```

Pertanyaan :

1. Jika pada perulangan for, inisialisasi `i=1` diubah menjadi `i=0`, apa akibatnya? Mengapa bisa demikian?
 - Loop akan berjalan satu kali lebih banyak, karena i dimulai dari 0. Misalnya, jika n=5, maka akan mencetak 6 bintang
2. Jika pada perulangan for, kondisi `i<=n` diubah menjadi `i<n`, bagaimana bentuk outputnya jika input n = 5? Mengapa hasilnya berbeda?
 - Hanya akan mencetak 4 bintang, karena loop berhenti saat i=5.
3. Jika pada perulangan for, kondisi `i <= n` diubah menjadi `i > n`, apa akibatnya? Mengapa bisa demikian?
 - Loop tidak akan pernah dijalankan, karena kondisi awal sudah salah.

4. Jika pada perulangan for, kondisi step `i++` diubah menjadi `i--` apa akibatnya? Mengapa bisa demikian?
- Program akan mengalami infinite loop karena nilai i semakin kecil dan tidak pernah memenuhi kondisi berhenti ($i \leq n$).
5. Jika pada perulangan for, step `i++` diubah menjadi `i += 2`, bagaimana pola outputnya jika input $n = 6$? Apa yang menyebabkan perubahan tersebut?
- Akan mencetak 3 bintang, karena i naik 2 setiap pengulangan sehingga $i = 1, 3, 5$, dan 7 (berhenti)

2.2 Percobaan 2 : Bintang Persegi

```

J Square18.java 1, U X
jobsheet8 > J Square18.java > Language Support for Java(TM) by Red Hat > ⚙ Square18.java
1 import java.util.Scanner;
2
3 public class Square18 {
4     Run main | Debug main | Run | Debug
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);    Resource 1
7
8             System.out.print(s: "Masukan 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 }
```

Masukan Nilai n : 5

Pertanyaan :

- Perhatikan perulangan luar. Jika pada sintaks for, inisialisasi `iOuter=1` diubah menjadi `iOuter=0`, apa yang akibatnya? Mengapa bisa demikian?
 - pengulangan akan berjalan satu kali lebih banyak, karena dimulai dari 0 dan berakhir di pengulangan $i = 5$ (6 pengulangan)
- 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?
 - Setiap baris akan mencetak satu bintang lebih banyak (jika $n = 5$), karena dimulai dari 0 dan berakhir di pengulangan $i = 5$ (6 pengulangan)

3. Apakah perbedaan kegunaan antara perulangan luar dengan perulangan yang berada di dalamnya?
 - Perulangan luar untuk mengatur/mengontrol jumlah baris
 - Perulangan dalam untuk mengatur/mengontrol jumlah bintang pada di tiap baris
4. Mengapa perlu ditambahkan sintaks `System.out.println();` di bawah perulangan dalam? Apa akibatnya jika sintaks tersebut dihilangkan?
 - Untuk mencetak hasil pengulangan berikutnya di baris baru. Jika dihilangkan maka hasilnya semua bintang akan tercetak pada satu baris yang sama.
5. **Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan 2”**

2.3 Percobaan 3 : Bintang segitiga

```
J Triangle18.java 1 X
jobsheet8 > J Triangle18.java > ...
1 import java.util.Scanner;
2
3 public class Triangle18 {
4     Run main | Debug main | Run | Debug
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);    Reso
7
8         System.out.print(s: "Masukan nilai n : ");
9         int n = sc.nextInt();
10
11        int i = 0;
12        while ( i <= n ) {
13            int j = 0;
14            while ( j < i ) {
15                System.out.print(s: "*");
16                j++;
17            }
18            i++;
19        }
20    }
21 }
22 }
```

Masukan nilai n : 5

Pertanyaan :

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

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

- Tidak sesuai

2. Jika tidak sesuai, bagian mana saja yang harus diperbaiki/ditambahkan? Jelaskan setiap bagian yang perlu diperbaiki/ditambahkan.
- Jika outputnya bisa menjadi sesuai dengan gambar pada no 1, maka terdapat 2 hal yang perlu diperbaiki/ditambahkan.
 1. Inisialisasi i harus dimulai dari 1, bukan 0
Pada kode awal, int i = 0; menyebabkan loop berjalan 6 kali jika n = 5, karena kondisi while (i <= n) akan menghitung dari 0 hingga 5.
Untuk memperbaikinya Ganti menjadi int i = 1; agar loop hanya berjalan 5 kali, yaitu dari 1 hingga 5.
 - 2.Tambahkan System.out.println(); setelah inner loop
Kode awal tidak mencetak baris baru setelah setiap baris bintang, sehingga semua bintang tercetak dalam satu baris panjang.
Untuk memperbaikinya
Tambahkan System.out.println(); setelah inner loop, agar setiap baris bintang ditampilkan di baris baru dan membentuk pola segitiga siku-siku.

```

Triangle18.java 1. M ×
jobsheet8 > J Triangle18.java > Java > Triangle18 > main(String[] args)
1 import java.util.Scanner;
2
3 public class Triangle18 {
4     Run main | Debug main | Run | Debug
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);    Resourc
7
8             System.out.print("Masukan nilai n : ");
9             int n = sc.nextInt();
10
11            int i = 1;
12            while ( i <= n ) {
13                int j = 0;
14                while ( j < i ) {
15                    System.out.print("*");
16                    j++;
17                }
18                System.out.println();
19                i++;
20            }
21        }
22    }
23

```

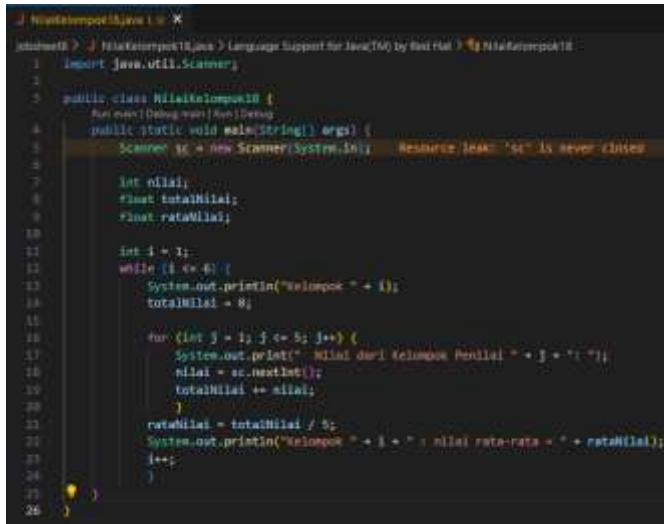
Masukan nilai n : 5

*
**

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 : mengontrol jumlah baris segitiga.
 - J : mengontrol jumlah bintang per baris.
 - j di-reset ke 0 agar setiap baris dimulai dari 0 bintang lagi.
 - Jika j tidak di-reset, bintang akan terus bertambah tanpa reset, sehingga pola segitiga tidak terbentuk.

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

2.4 percobaan 4 : Studi Kasus Nilai Tugas Proyek Kelompok



```
J Nilaikelompok10.java X
junit4 > J Nilaikelompok10 > LanguageSupportForJavaCDO by Best Hall > J Nilaikelompok10
1 import java.util.Scanner;
2
3 public class Nilaikelompok10 {
4     //No even debugging available
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in); //Resource leak: 'sc' is never closed
7
8         int nilai;
9         float totalNilai;
10        float rataNilai;
11
12        int i = 1;
13        while (i <= 6) {
14            System.out.println("kelompok " + i);
15            totaNilai = 0;
16
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            i++;
25        }
26    }
}
```

```
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 3: 80
Nilai dari Kelompok Penilai 5: 76
Nilai dari Kelompok Penilai 4: 77
Nilai dari Kelompok Penilai 5: 83
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
```

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.
 - Agar ketika nilai total dijumlahkan, nilai total dari kelompok sebelumnya tidak ikut dijumlahkan sehingga nilai totalnya sesuai.
2. Modifikasi program di atas, sehingga dapat mencari kelompok dengan rata-rata nilai tertinggi dan tampilkan nomor kelompok tersebut.

```
3 Nilaikelompok10.java 1.0 *
josheddi > 3 Nilaikelompok10.java > Java > $3 Nilaikelompok10 > ⑩ main(String[] args)
1 import java.util.Scanner;
2
3 public class Nilaikelompok10 {
4     Run main | Debug main | Run | Debug
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in); // Resource leak: 'sc' is never closed
7
8         int nilai;
9         float totalNilai;
10        float rataNilai;
11        int kelompokTerbaik = 0;
12        float rataTinggi = 0;
13
14        int i = 1;
15        while (i <= 6) {
16            System.out.println("Kelompok " + i);
17            totalNilai = 0;
18
19            for (int j = 1; j <= 5; j++) {
20                System.out.print(" Nilai dari Kelompok " + j + ": ");
21                nilai = sc.nextInt();
22                totalNilai += nilai;
23            }
24            rataNilai = totalNilai / 5;
25            System.out.println("Kelompok " + i + " : nilai rata-rata = " + rataNilai);
26
27            if (rataNilai > rataTinggi) {
28                rataTinggi = rataNilai;
29                kelompokTerbaik = i;
30            }
31        }
32
33        System.out.println("\nKelompok dengan rata-rata nilai tertinggi adalah: Kelompok " + kelompokTerbaik);
34    }
35 }
```

```

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

Kelompok dengan rata-rata nilai tertinggi adalah kelompok 6

```

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

3. 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”

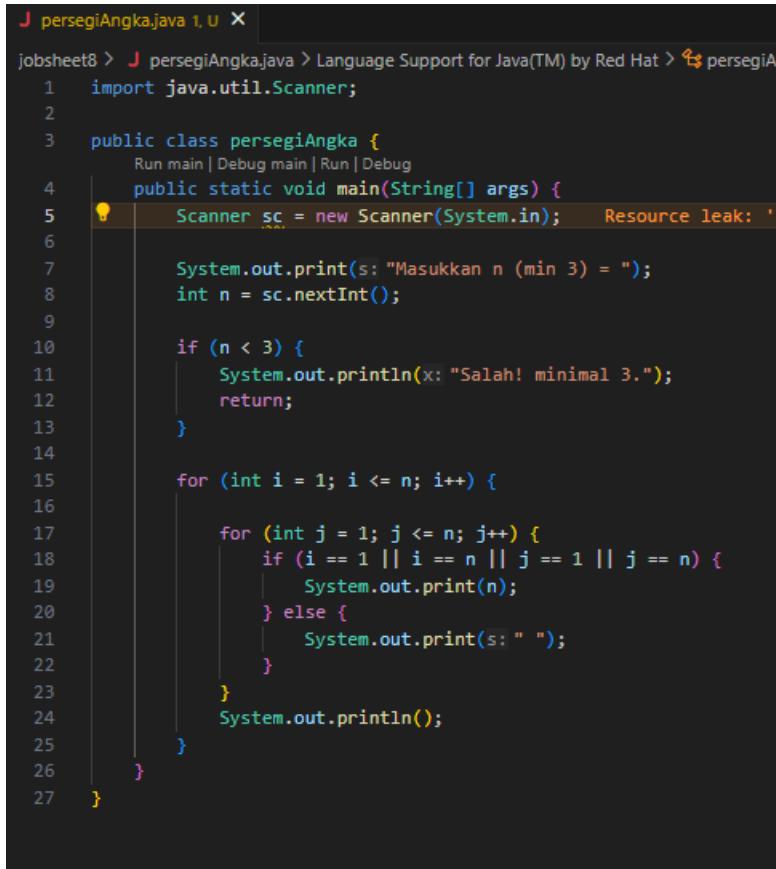
```
J_mihKuadrat.java | 0 X
jobsheet1 > J_mihKuadrat.java > Language Support for Java(TM) by Red Hat > J_mihKuadrat
1 import java.util.Scanner;
2
3 public class JmlHkuadrat {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in); Resource leak: "sc"
6
7
8         System.out.print("Masukkan n = ");
9         int n = sc.nextInt();
10
11        for (int i = 1; i <= n; i++) {
12            int jumlah = 0;
13
14            for (int j = 1; j <= i; j++) {
15                jumlah += (j * j);
16            }
17
18            System.out.print("n = " + i + " -> jumlah kuadrat = ");
19            for (int j = i; j <= i; j++) {
20                System.out.print(j * j);
21                if (j < i) {
22                    System.out.print(" + ");
23                }
24            }
25            System.out.println(" " + jumlah);
26        }
27    }
28 }
```

Masukkan n = 5
n = 1 -> jumlah kuadrat = 1 = 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

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

3 3 3	5 5 5 5 5
	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”



```
J persegiAngka.java 1, U X
jobsheet8 > J persegiAngka.java > Language Support for Java(TM) by Red Hat > persegiA
  1 import java.util.Scanner;
  2
  3 public class persegiAngka {
  4     Run main | Debug main | Run | Debug
  5     public static void main(String[] args) {
  6         Scanner sc = new Scanner(System.in);      Resource leak: 'sc' is never closed
  7         System.out.print(s: "Masukkan n (min 3) = ");
  8         int n = sc.nextInt();
  9
 10        if (n < 3) {
 11            System.out.println(x: "Salah! minimal 3.");
 12            return;
 13        }
 14
 15        for (int i = 1; i <= n; i++) {
 16
 17            for (int j = 1; j <= n; j++) {
 18                if (i == 1 || i == n || j == 1 || j == n) {
 19                    System.out.print(n);
 20                } else {
 21                    System.out.print(s: " ");
 22                }
 23            }
 24            System.out.println();
 25        }
 26    }
 27 }
```

Masukkan n (min 3) = 3 Masukkan n (min 3) = 5
333 5 5
3 3 5 5
333 55555

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”

The screenshot shows a terminal window with two panes. The left pane displays the Java code for a program named `KopiSeri.java`. The right pane shows the program's execution and its output.

```

KopiSeri.java:11:X
jennifermacbook: ~ % java KopiSeri
import java.util.Scanner;

public class KopiSeri {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in); // Resource Java "sc" is never closed

        System.out.print("Jumlah cabang kafe: ");
        int cabang = sc.nextInt();

        int totalPelanggan = 0;
        int totalItem = 0;

        System.out.println("\n==== Input Penjualan Per Cabang ===");

        for (int c = 1; c <= cabang; c++) {
            System.out.print("Cabang " + c + ": ");
            System.out.print("Jumlah pelanggan: ");
            int pelanggan = sc.nextInt();

            int itemCabang = 0;

            for (int p = 1; p <= pelanggan; p++) {
                System.out.print("- Pelanggan " + p + " memesan berapa item? ");
                int item = sc.nextInt();
                itemCabang += item;
            }

            System.out.println("Cabang " + c + ":");
            System.out.println("- Pelanggan: " + pelanggan + " orang");
            System.out.println("- Item terjual: " + itemCabang);

            totalPelanggan += pelanggan;
            totalItem += itemCabang;
        }

        System.out.println("\nTotal seluruh Cabang:");
        System.out.println("Pelanggan: " + totalPelanggan + " orang");
        System.out.println("Item terjual: " + totalItem);
    }
}

```

The right pane shows the program's output:

```

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

```

