

Nama : Deswita Khansa Rafifah

NIM : 254107020151

Kelas : TI_1G

JOBSHEET 8

Percobaan 1

```
Star6.java 1.0
package jobsheet8;
import java.util.Scanner;
public class Star6 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Masukkan nilai n = ");
        int n = sc.nextInt();
        for (int i = 1; i <= n; i++) {
            System.out.print("*");
        }
    }
}

t8.Star6
Masukkan nilai n = 5
*****
PS C:\Users\Asus\Documents\DASPRO\PraktikumDaspro\PraktikumDaspro\PraktikumDaspro>
```

Pertanyaan

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

Jawab: Jika inisialisasi diubah menjadi $i = 0$, jumlah bintang yang dicetak bertambah satu. Karena perulangan dimulai dari 0 hingga n , maka total iterasi menjadi $n + 1$ kali, bukan n kali. Akibatnya output akan menampilkan satu bintang lebih banyak.

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

Jawab:

Program asli menggunakan `for (int i = 1; i <= n; i++)`, sehingga untuk $n = 5$ nilai i berjalan 1, 2, 3, 4, 5 → total 5 iterasi → output 5 bintang.

Jika kondisi diubah menjadi `for (int i = 1; i < n; i++)`, maka untuk $n = 5$ nilai i berjalan 1, 2, 3, 4 → total 4 iterasi → output 4 bintang.

Perbedaannya terjadi karena $i \leq n$ memasukkan nilai n sebagai iterasi terakhir sedangkan $i < n$ mengeluarkan nilai n , sehingga jumlah iterasi (dan jumlah bintang) berkurang satu.

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

Jawab: Jika kondisi menjadi $i > n$, perulangan tidak akan berjalan sama sekali, karena kondisi langsung bernilai FALSE pada awal eksekusi ($1 > 5$ adalah FALSE). Akibatnya tidak ada bintang yang dicetak.

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

Jawab: Jika step diubah menjadi $i--$, maka nilai i akan terus berkurang pada setiap iterasi sehingga bergerak menjauhi nilai batasnya. Akibatnya, kondisi $i \leq n$ tidak akan pernah menjadi false, sehingga perulangan tidak akan berhenti dan menyebabkan infinite loop. Perulangan terus berjalan karena arah perubahan nilai i tidak mendekati kondisi penghentian.

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

Jawab: Dengan step $i += 2$ dan $n = 6$, perulangan hanya menjalankan iterasi pada nilai i : 1, 3, dan 5. Akibatnya, output akan menampilkan 3 bintang. Perubahan terjadi karena nilai i meningkat dua angka setiap iterasi, sehingga jumlah iterasi berkurang menjadi setengah dari jumlah normal, dan bintang yang dicetak juga menjadi lebih sedikit.

Percobaan 2

The screenshot shows the Eclipse IDE interface with two open files: Star6.java and Square6.java. The code for Square6.java is as follows:

```
1 package jobsheet8;
2
3 import java.util.Scanner;
4 public class Square6 {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
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 }
```

The terminal window shows the execution of the code and its output:

```
kumDaspro_96d52dac\bin\jobsheet8.Square6
Masukkan nilai n = 5
*****
*****
*****
*****
*****
```

Pertanyaan

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

Jawab: Jika perulangan luar dimulai dari 0, jumlah baris yang dihasilkan menjadi bertambah satu baris. Hal ini terjadi karena perulangan berjalan mulai dari 0 hingga nilai n, sehingga total iterasi menjadi $n + 1$ kali, bukan n kali. Akibatnya bentuk persegi memiliki satu baris tambahan.

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?

Jawab: Jika perulangan dalam dimulai dari 0, jumlah bintang pada setiap baris bertambah satu. Ini karena inner loop berjalan dari 0 sampai nilai n, sehingga menghasilkan $n + 1$ bintang dalam satu baris. Output menjadi lebih lebar satu bintang dibandingkan program asli.

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

Jawab: Perulangan luar bertugas menentukan jumlah baris pada pola yang dicetak. Perulangan dalam berfungsi menentukan jumlah bintang dalam setiap baris. Dengan kata lain, outer loop mengatur baris, sedangkan inner loop mengatur isi baris.

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

Jawab: **System.out.println()** digunakan untuk memindahkan posisi cetak ke baris baru setelah inner loop selesai mencetak bintang. Jika perintah ini dihilangkan, seluruh bintang akan tercetak dalam satu baris panjang tanpa turun, sehingga bentuk persegi tidak muncul dan pola menjadi tidak terbaca.

Percobaan 3

The screenshot shows the Eclipse IDE interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Project Explorer:** Shows three Java files: Star6.java, Square6.java, and Triangle6.java.
- Code Editor:** Displays the code for Triangle6.java. The code defines a class Triangle6 with a main method. It prompts the user to enter a value for n, initializes variables i and j, and then uses nested loops to print a triangle pattern of asterisks. The code ends with a closing brace for the class.
- Terminal:** Shows the output of running the code. It asks for input "Masukkan nilai n = ", receives "5" as input, and then prints a 5x5 triangle pattern of asterisks: *****
- Status Bar:** Shows the current file is main*, Java Ready, and the terminal output PS C:\Users\Asus\Documents\Daspro\PraktikumDaspro\PraktikumDaspro>.

Pertanyaan

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

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

Jawab: Tidak, Output dengan nilai $n = 5$ belum sesuai. Program menampilkan satu baris kosong di awal dan jumlah baris yang ditampilkan menjadi lebih banyak dari seharusnya. Hal ini membuat bentuk segitiga tidak sesuai dengan pola yang diharapkan.

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

Jawab:

Beberapa bagian yang perlu diperbaiki:

1. Kondisi perulangan while harus disesuaikan agar jumlah baris tidak berlebih.
2. Perintah untuk mencetak baris baru harus ditambahkan setelah perulangan dalam agar bentuk segitiga muncul dengan benar.
3. Inner loop harus memastikan jumlah bintang bertambah sesuai baris, sehingga pola segitiga terbentuk.

Perbaikan ini diperlukan untuk mengatur baris dan jumlah bintang agar sesuai pola yang diinginkan.

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?

Jawab:

- Variabel **i** berfungsi sebagai penghitung jumlah baris (outer loop).
- variabel **j** berfungsi sebagai penghitung jumlah bintang pada setiap baris (inner loop).

Variabel **j** harus di-reset ke 0 setiap kali outer loop berpindah ke baris berikutnya agar perhitungan jumlah bintang dimulai dari awal pada setiap baris. Jika **j** tidak di-reset, maka nilai **j** akan terus meningkat dari iterasi sebelumnya sehingga jumlah bintang menjadi tidak teratur dan pola segitiga tidak terbentuk dengan benar.

Percobaan 4

```
PraktikumDaspro
File Edit Selection View Go Run ...
NilaiKelompok6.java U NilaiKelompok6.java U NilaiKelompok6.java U
jobsheet8 > J NilaiKelompok6.java > NilaiKelompok6 > main(String[])
4 public class NilaiKelompok6 {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         int i = 1, j, nilai;
9         float totalNilai, rataNilai;
10
11        while (i <= 6) {
12            System.out.println("Kelompok " + i);
13
14            totalNilai = 0;
15            for (j = 1; j <= 5; j++) {
16                System.out.print("Nilai dari Kelompok Penilai " + j + ": ");
17                nilai = sc.nextInt();
18                totalNilai += nilai;
19            }
20
21            rataNilai = totalNilai / 5;
22            System.out.println("Kelompok " + i + ": nilai rata-rata = " + rataNilai);
23            i++;
24        }
25    }
26}
PROBLEMS 16 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Run: NilaiKelompok6 + ...
Kelompok 1
Nilai dari Kelompok Penilai 1: 88
Nilai dari Kelompok Penilai 2: 79
Nilai dari Kelompok Penilai 3: 98
Nilai dari Kelompok Penilai 4: 82
Nilai dari Kelompok Penilai 5: 85
Kelompok 1: nilai rata-rata = 84.8
Kelompok 2
...
Ln 22, Col 86 Spaces: 4 UTF-8 CRLF {} Java Signed out
```

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.
Jawab: Variabel totalNilai akan kembali diset menjadi **0** pada setiap iterasi perulangan luar (outer loop). Hal ini dilakukan karena setiap kelompok mahasiswa harus dihitung total nilainya secara terpisah. Jika inisialisasi $totalNilai = 0$ diletakkan di luar outer loop, maka nilai total akan terus bertambah dan tercampur dengan kelompok sebelumnya, sehingga perhitungan rata-rata menjadi salah. Dengan menempatkan inisialisasi di dalam outer loop, setiap kelompok memulai perhitungan dari awal.
2. Modifikasi program di atas, sehingga dapat mencari kelompok dengan rata-rata nilai tertinggi dan tampilkan nomor kelompok tersebut.

Jawab:

```
PraktikumDaspro
File Edit Selection View Go Run ...
NilaiKelompok6.java U NilaiKelompok6.java U NilaiKelompok6.java U
jobsheet8 > J NilaiKelompok6.java > NilaiKelompok6 > main(String[])
4 public class NilaiKelompok6 {
5     public static void main(String[] args) {
6         System.out.print("Nilai dari Kelompok Penilai " + j + ": ");
7         nilai = sc.nextInt();
8         totalNilai += nilai;
9     }
10
11    rataNilai = totalNilai / 5;
12    System.out.println("Kelompok " + i + ": nilai rata-rata = " + rataNilai);
13
14    if (rataNilai > rataTertinggi) {
15        rataTertinggi = rataNilai;
16        kelompokTertinggi = i;
17    }
18
19    i++;
20
21}
22
23    System.out.println("=====");
24    System.out.println("Kelompok dengan rata-rata tertinggi:");
25    System.out.println("Kelompok 6 dengan nilai rata-rata 88.8");
26
27PS C:\Users\Asus\Documents\DAASPRO\PraktikumDaspro\PraktikumDaspro>
PROBLEMS 16 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Run: NilaiKelompok6 + ...
=====
Kelompok dengan rata-rata tertinggi:
Kelompok 6 dengan nilai rata-rata 88.8
PS C:\Users\Asus\Documents\DAASPRO\PraktikumDaspro\PraktikumDaspro>
Ln 33, Col 67 Spaces: 4 UTF-8 CRLF {} Java Signed out
```

TUGAS

Tugas 1

The screenshot shows a Java code editor with the following code:

```
File Edit Selection View Go Run ... ← → Q PraktikumDaspro
J Tugas1.java 3 ×
jobsheet8 > J Tugas1.java > Tugas1 > main(String[])
1 package jobsheet8;
2
3 import java.util.Scanner;
4 public class Tugas1 {
    Run | Debug
5     public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
7
8         int n, i, jumlah;
9
10        for (n = 1; n <= 5; n++) {
11            jumlah = 0;
12            System.out.print("n = " + n + " jumlah kuadrat = ");
13
14            for (i = 1; i <= n; i++) {
15                jumlah += i * i;
16                System.out.print(i * i);
17                if (i < n) {
18                    System.out.print(" + ");
19                }
20            }
}
PS C:\Users\Asus\Documents\APLIKASI\PraktikumDaspro\PraktikumDaspro>
```

The code calculates the sum of squares of numbers from 1 to n. The output window shows the results for n values from 1 to 5.

Tugas 2

The screenshot shows a Java code editor with the following code:

```
J Tugas2.java 1 x
jobsheet8 > J Tugas2.java > Tugas2 > main(String[])
1 package jobsheet8;
2
3 import java.util.Scanner;
4 public class Tugas2 {
    Run | Debug
5     public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Masukkan nilai n (minimal 3): ");
        int n = sc.nextInt();

        if (n < 3) {
            System.out.println("Nilai n minimal 3!");
            return;
        }

        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 + " ");
                }
            }
        }
    }
}
```

The code is a Java program that prints a triangular pattern of numbers. It takes an integer input 'n' from the user. If 'n' is less than 3, it prints an error message and exits. Otherwise, it prints a triangle where each row has 'n' elements, and the first and last elements of each row are 'n'. The other elements in each row are also 'n'.

Tugas 3

The screenshot shows a Java code editor interface with a terminal window below it. The code editor displays a file named Tugas3.java with the following content:

```
jobsheet8 > J Tugas3.java > Tugas3 > main(String[])
4  public class Tugas3 {
5      public static void main(String[] args) {
15         for (int i = 1; i <= jumlahCabang; i++) {
16             System.out.println("\n--- Cabang " + i + "---");
17
18             int totalItemCabang = 0;
19
20             System.out.print("Jumlah pelanggan: ");
21             int jumlahPelangganCabang = sc.nextInt();
22
23             for (int j = 1; j <= jumlahPelangganCabang; j++) {
24                 System.out.print("- Pelanggan " + j + " memesan berapa item? ");
25                 int itemDipesan = sc.nextInt();
26
27                 totalItemCabang += itemDipesan;
28             }
29
30             System.out.println("Cabang " + i + ":");
31             System.out.println("- Pelanggan: " + jumlahPelangganCabang + " orang");
32             System.out.println("- Item terjual: " + totalItemCabang);
33     }
}
--- 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:
```

The terminal window shows the execution of the program. It prompts for the number of branches (jumlahCabang), which is 3. For each branch, it asks for the number of customers (jumlahPelanggan) and the number of items ordered by each customer (itemDipesan). The total number of items sold per branch and the total number of items sold across all branches are then printed.