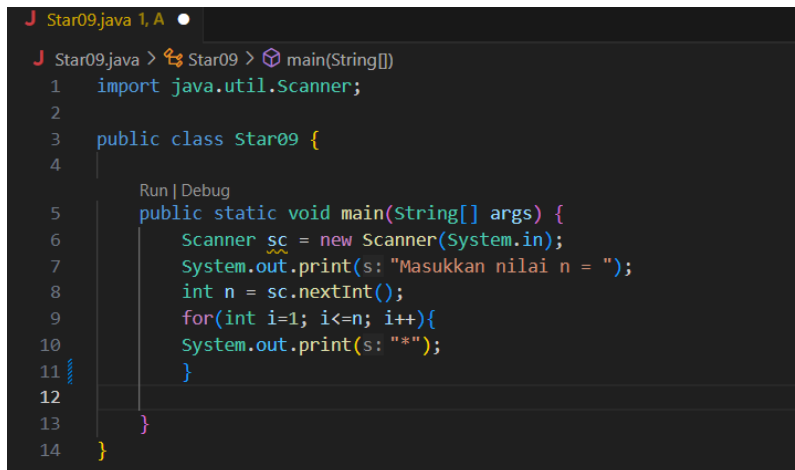


NAMA : FAFIQ LUTFI AZANA

KELAS : TI_1G

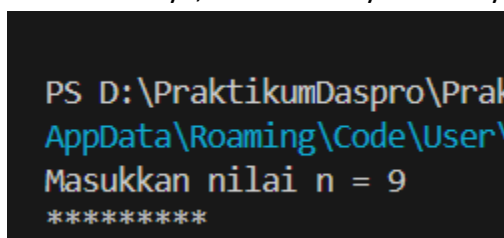
NIM : 254107020058

1. Buat folder baru pada repositori lokal Anda, beri nama jobsheet8
2. Buat file baru StarNoPresensi.java
3. Buat fungsi/method main() di dalamnya.
4. Karena program membutuhkan input dari keyboard, maka perlu import class Scanner. Jadi tambahkan sintaks import di baris atas sendiri program. `import java.util.Scanner;`
5. Di dalam fungsi main() yang telah dibuat, deklarasikan objek Scanner dengan nama sc. `Scanner sc = new Scanner(System.in);`
6. Pada baris selanjutnya, tampilkan instruksi untuk memasukkan nilai yang akan disimpan ke variabel n. `System.out.print("Masukkan nilai n = "); int n = sc.nextInt();`
7. Pada baris selanjutnya, buat sintaks perulangan dengan for seperti di bawah ini. `for(int i=1; i<=n; i++){ System.out.print("*"); }` Catatan: perlu diperhatikan, bahwa yang digunakan adalah perintah print, bukan println karena kita ingin menampilkan tanpa ada baris baru



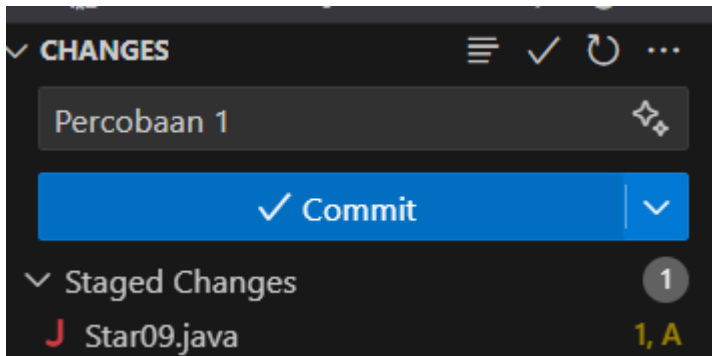
```
J Star09.java 1, A
J Star09.java > Star09 > main(String[])
1  import java.util.Scanner;
2
3  public class Star09 {
4
5      Run | Debug
6      public static void main(String[] args) {
7          Scanner sc = new Scanner(System.in);
8          System.out.print(s: "Masukkan nilai n = ");
9          int n = sc.nextInt();
10         for(int i=1; i<=n; i++){
11             System.out.print(s: "*");
12         }
13     }
14 }
```

8. Compile dan jalan program.
9. Amati hasilnya, maka hasilnya harusnya akan serupa dengan tampilan di bawah ini.



```
PS D:\PraktikumDaspro\Prak
AppData\Roaming\Code\User\
Masukkan nilai n = 9
*****
```

10. Commit program Anda ke Github dengan pesan "Percobaan 1"



Pertanyaan

1. Jika pada perulangan for, inisialisasi $i=1$ diubah menjadi $i=0$, apa akibatnya? Mengapa bisa demikian?
2. Jika pada perulangan for, kondisi $i \leq n$ diubah menjadi $i > n$, apa akibatnya? Mengapa bisa demikian?
3. Jika pada perulangan for, kondisi $i \leq n$ diubah menjadi $i > n$, apa akibatnya? Mengapa bisa demikian?
4. Jika pada perulangan for, kondisi step $i++$ diubah menjadi $i--$ apa akibatnya? Mengapa bisa demikian?
5. Jika pada perulangan for, step $i++$ diubah menjadi $i += 2$, bagaimana pola outputnya jika input $n = 6$? Apa yang menyebabkan perubahan tersebut?

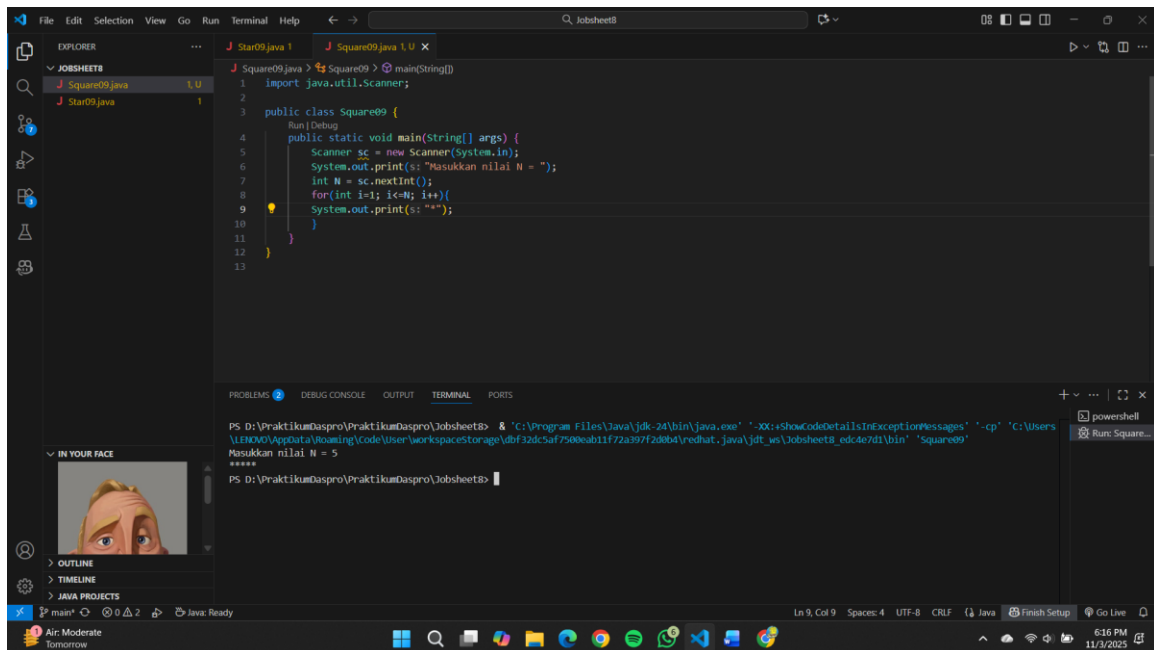
Jawaban

1. Yang terjadi adalah program menghitung tidak mulai dari 1 tapi mulai dari 0. Jika pada praktek saya memasukan nilai $n = 9$, maka ketika i dimulai dari 0, program akan menghitung mulai dari 0 sehingga Bintang yang tercetak berjumlah 10.
2. Yang terjadi adalah program akan menjalankan perintah jika $i < n$ dimana ketika saya memasukan $n = 9$, program akan menjalankan perulangan sebanyak 8 kali karena ketika $i = 9$ program akan berhenti.
3. Yang terjadi adalah program tidak berjalan karena kondisi $i > n$ tidak terpenuhi sehingga perulangan tidak berjalan.
4. Yang terjadi adalah perulangan akan terus dijalankan karena kondisi $i \leq n$ selalu terpenuhi sehingga perulangan terus dijalankan.
5. Jika $i++$ diubah dengan $i += 2$, output yang tercetak adalah 3 bintang karena setiap perulangan yang dijalankan, i ditambah 2 sehingga hanya setengah dari n yang tercetak.

Percobaan 2

1. Pada percobaan ke-2 akan dilakukan percobaan tentang nested loop. Kasus yang akan diselesaikan adalah untuk membuat tampilan bujursangkar *, dengan panjang sisi sebanyak n. Misalkan n dimasukan 5, maka hasilnya adalah: *****

2. Kalau diamati lebih lanjut, sebenarnya mirip dengan kasus percobaan 1. Jika di percobaan 1, misal input n bernilai 5, maka yang akan dihasilkan adalah ***** (bisa dianggap ini sebagai inner loop yang mencetak 5 bintang *****), maka untuk kasus percobaan 2 ini, hasil dari percobaan 1 tersebut hanya perlu diulang lagi sebanyak n kali. Yaitu dengan menambahkan outer loop untuk mengulangi proses inner loop sebanyak n kali.
3. Buat file baru SquareNoPresensi.java
4. Karena program membutuhkan input dari keyboard, maka perlu import class Scanner. Jadi tambahkan sintaks import di baris atas sendiri program. import java.util.Scanner;
5. Buat method main(), dan isikan kode program yang sama dengan isi method main() di percobaan 1.
6. Compile dan jalankan program. Pastikan program berjalan seperti saat menjalankan percobaan 1.



```

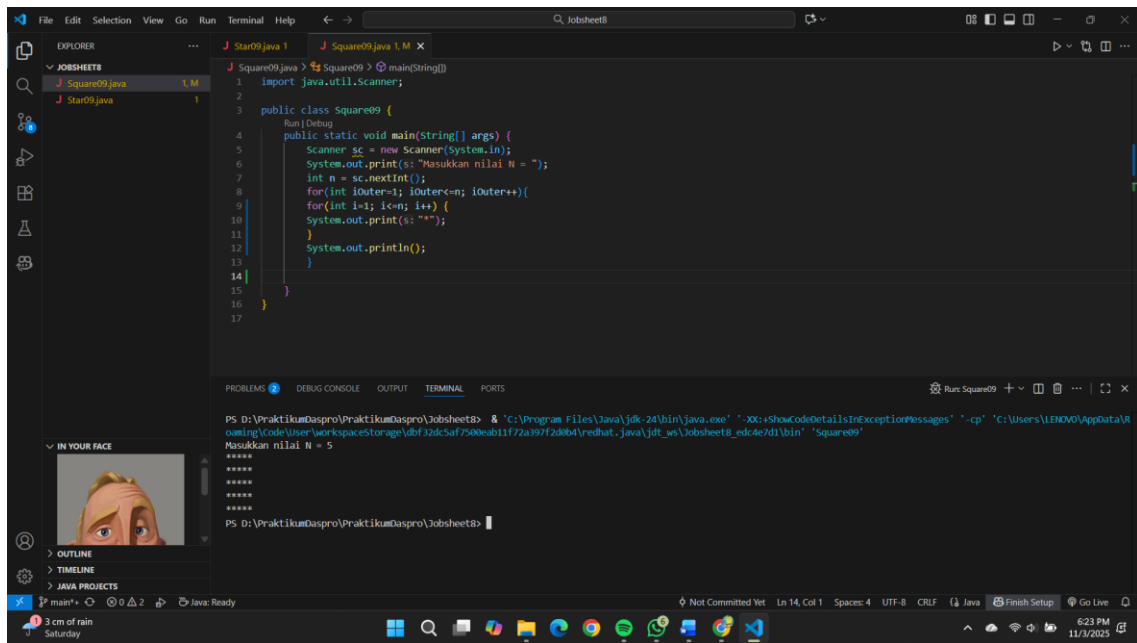
1 Square09.java > Square09 > main(String[])
2 import java.util.Scanner;
3
4 public class Square09 {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7         System.out.print(s: "Masukkan nilai N = ");
8         int N = sc.nextInt();
9         for(int i=1; i<=N; i++){
10             System.out.print(s: "*");
11         }
12     }
13 }

```

PS D:\PraktikumDaspro\PraktikumDaspro\Jobsheet8> & "C:\Program Files\Java\jdk-24\bin\java.exe" "-XX:+ShowCodeDetailsInExceptionMessages" "-cp" "C:\Users\LEB000\AppData\Roaming\Code\User\workspaceStorage\bf32dc5af7500eab11f72a397f2db04\redhat_java\jdk_us\Jobsheet8_ed4e7d1\bin" "Square09"

Masukkan nilai N = 5

7. Perhatikan sintaks perulangan yang digunakan untuk mencetak * sebanyak n kali ke arah samping. Di langkah 8 di atas kode for (kotak merah) dijadikan sebagai inner loop.
8. Perulangan pada inner loop dilakukan sebanyak n kali untuk menghasilkan output seperti pada poin 1. Maka perlu ditambahkan perulangan luar (outer loop). for(int iOuter=1; iOuter<=n; iOuter++){ for(int i=1; i<=n; i++) { System.out.print("*"); } System.out.println(); }
9. Simpan perubahan, compile dan jalankan program.



```
1 import java.util.Scanner;
2
3 public class Square09 {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         System.out.print("Masukkan nilai N = ");
7         int n = sc.nextInt();
8         for(int iOuter=1; iOuter<=n; iOuter++){
9             for(int i=1; i<=n; i++) {
10                 System.out.print(" *");
11             }
12             System.out.println();
13         }
14     }
15 }
16
17
```

PS D:\PraktikumDaspro\PraktikumDaspro\Jobsheet8> & 'C:\Program Files\Java\jdk-24\bin\java.exe' ^-XX:ShowCodeDetailsInExceptionMessages' ^-cp 'C:\Users\LENOVO\AppData\Local\Temp\Code\User\workspaceStorage\df32dc5af7500eab11f72a397f2d8b4\redhat_java\jdk_ws\Jobsheet8_edc4e7d1\bin' 'Square09'

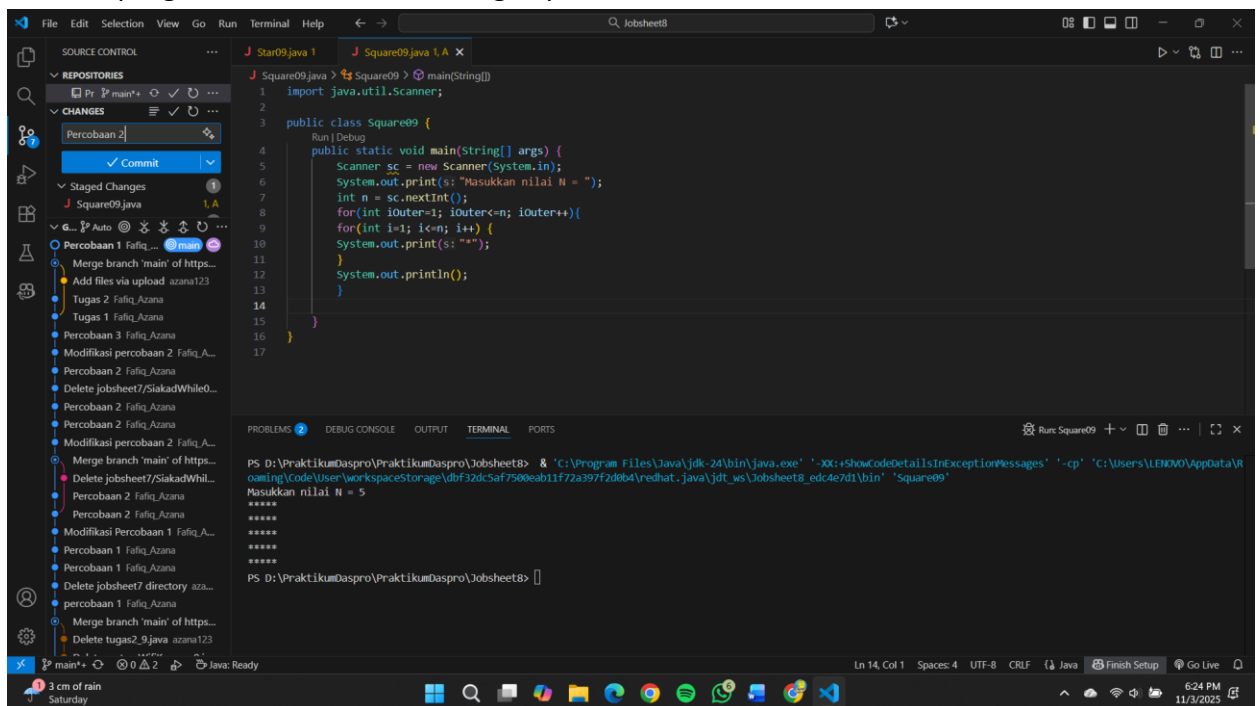
Masukkan nilai N = 5

PS D:\PraktikumDaspro\PraktikumDaspro\Jobsheet8> |

10. Amati hasilnya, maka hasilnya harusnya akan serupa dengan tampilan di bawah ini.

Masukkan nilai n = 5 *****

11. Commit program Anda ke Github dengan pesan “Percobaan 2”



```
1 import java.util.Scanner;
2
3 public class Square09 {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         System.out.print("Masukkan nilai N = ");
7         int n = sc.nextInt();
8         for(int iOuter=1; iOuter<=n; iOuter++){
9             for(int i=1; i<=n; i++) {
10                 System.out.print(" *");
11             }
12             System.out.println();
13         }
14     }
15 }
16
17
```

PS D:\PraktikumDaspro\PraktikumDaspro\Jobsheet8> & 'C:\Program Files\Java\jdk-24\bin\java.exe' ^-XX:ShowCodeDetailsInExceptionMessages' ^-cp 'C:\Users\LENOVO\AppData\Local\Temp\Code\User\workspaceStorage\df32dc5af7500eab11f72a397f2d8b4\redhat_java\jdk_ws\Jobsheet8_edc4e7d1\bin' 'Square09'

Masukkan nilai N = 5

PS D:\PraktikumDaspro\PraktikumDaspro\Jobsheet8> |

Pertanyaan

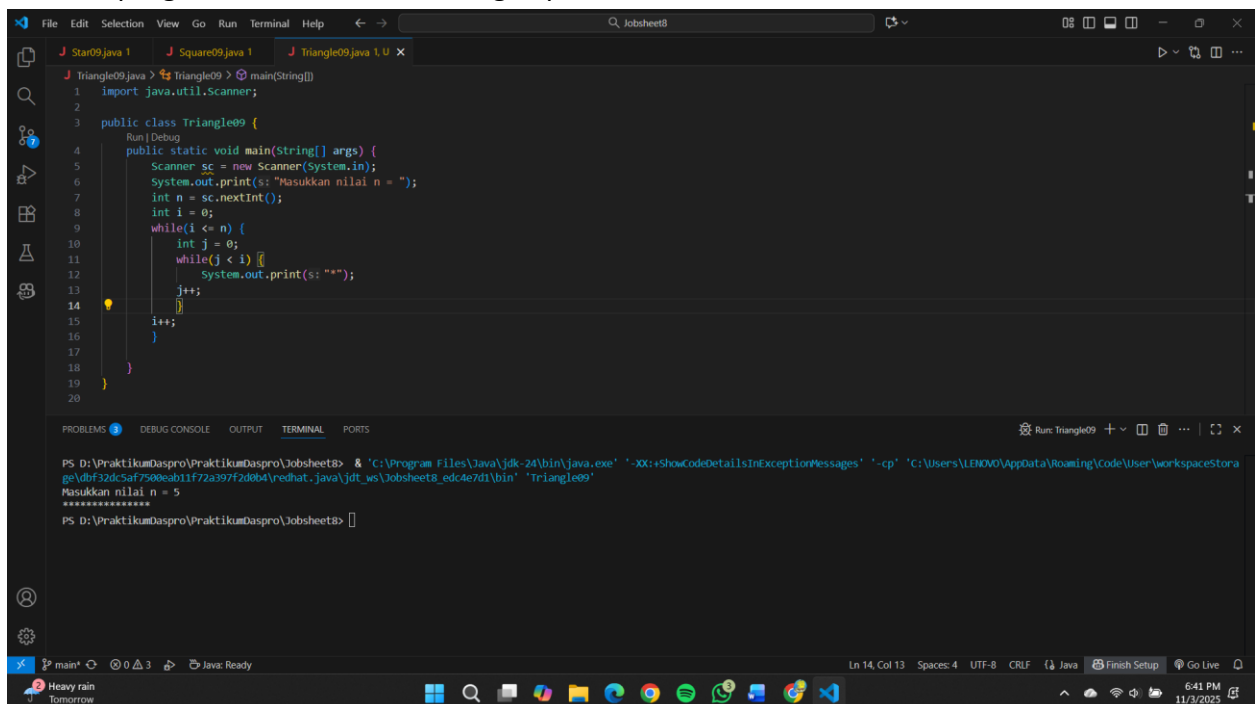
1. Perhatikan perulangan luar. Jika pada sintaks for, inisialisasi iOuter=1 diubah menjadi iOuter=0, apa yang akibatnya? Mengapa bisa demikian?
2. Kembalikan program semula dimana inisialisasi iOuter=1. Kemudian perhatikan perulangan dalam. Jika pada sintaks for, inisialisasi i=1 diubah menjadi i=0, apa yang akibatnya? Mengapa bisa demikian?
3. Apakah perbedaan kegunaan antara perulangan luar dengan perulangan yang berada di dalamnya?
4. Mengapa perlu ditambahkan sintaks System.out.println(); di bawah perulangan dalam? Apa akibatnya jika sintaks tersebut dihilangkan?
5. Commit dan push hasil modifikasi Anda ke Github dengan pesan "Modifikasi Percobaan 2"

Jawaban

1. Ketika diinputkan nilai $n = 5$ outer loop dijalankan sebanyak 6 kali karena counter menghitung dari 0 dan bukan 1 sehingga output yang tercetak adalah 5 bintang sebanyak 6 baris.
2. Ketika diinputkan nilai $n = 5$, inner loop dijalankan sebanyak 6 kali karena counter menghitung dari 0 dan bukan 1 sehingga output yang tercetak adalah 6 bintang sebanyak 5 baris.
3. Kegunaan perulangan dalam adalah untuk menentukan jumlah Bintang yang akan dicetak, sedangkan perulangan luar berfungsi untuk menentukan jumlah baris dari Bintang yang akan dicetak.
4. Supaya program menjalankan perulangan luar pada baris baru. Jika System.out.println(); dihilangkan, output yang keluar adalah bintang yang tercetak dalam satu baris.

Percobaan 3

1. Pada percobaan ke-3 akan dilakukan percobaan segitiga *, dengan sama siku dengan tinggi sebesar n. Misalkan n dimasukan 5, maka hasilnya adalah: * * * * *
* * * * *
* * * * *
* * * * *
* * * * *
2. Buat file baru TriangleNoPresensi.java
3. Karena program membutuhkan input dari keyboard, maka perlu import class Scanner.
4. Buat method main(), dan isikan kode program berikut kedalam method main(). Scanner sc = new Scanner(System.in); System.out.print("Masukkan nilai n = "); int n = sc.nextInt(); int i = 0; while(i <= n) { int j = 0; while(j < i) { System.out.print("*"); j++; } i++; }
5. Compile dan jalankan program. Amati apa yang terjadi.
6. Commit program Anda ke Github dengan pesan "Percobaan 3"



The screenshot shows an IDE with the following code in `TriangleNoPresensi.java`:

```
1 import java.util.Scanner;
2
3 public class TriangleNoPresensi {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         System.out.print("Masukkan nilai n = ");
7         int n = sc.nextInt();
8         int i = 0;
9         while(i <= n) {
10             int j = 0;
11             while(j < i) {
12                 System.out.print("*");
13                 j++;
14             }
15             i++;
16         }
17     }
18 }
19
20
```

The terminal output shows the program running and the user inputting 5:

```
PS D:\PraktikumDaspro\PraktikumDaspro\Jobsheet8> java -Xmx+ShowCodeDetailsInExceptionMessages -cp 'C:\Users\LENOVO\AppData\Local\Code\User\workspaceStorage\9b\9bdf32dc5af75993ab11f72a397f2d8b4\redhat_java\jdk_ws\Jobsheet8_edc4e7d1\bin' 'TriangleNoPresensi'
Masukkan nilai n = 5
*****
*****
*****
*****
*****
PS D:\PraktikumDaspro\PraktikumDaspro\Jobsheet8>
```

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

```
★
★ ★
★ ★ ★
★ ★ ★ ★
★ ★ ★ ★ ★
```

2. Jika tidak sesuai, bagian mana saja yang harus diperbaiki/ditambahkan? Jelaskan setiap bagian yang perlu diperbaiki/ditambahkan.
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?
4. Commit dan push hasil modifikasi Anda ke Github dengan pesan "Modifikasi Percobaan 3"

Jawaban

1. Tidak

The screenshot shows an IDE with a Java file named `Triangle09.java`. The code is as follows:

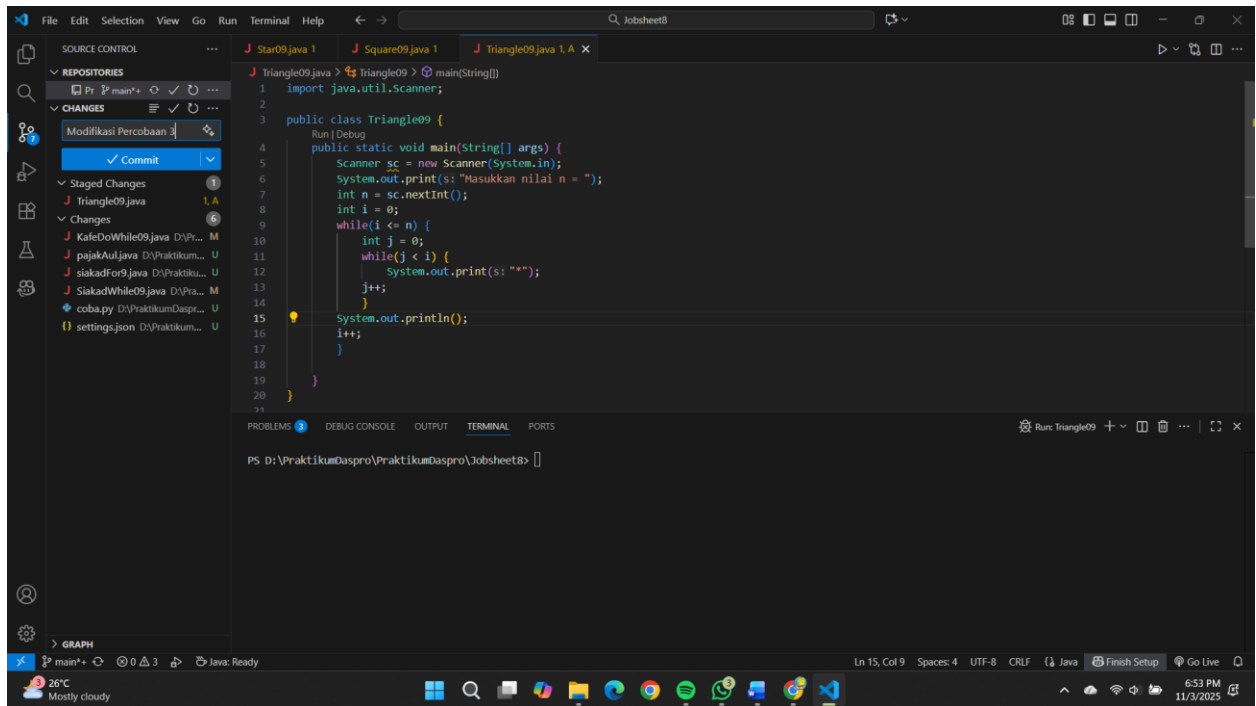
```
1 import java.util.Scanner;
2
3 public class Triangle09 {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         System.out.print(s: "Masukkan nilai n = ");
7         int n = sc.nextInt();
8         int i = 0;
9         while(i <= n) {
10             int j = 0;
11             while(j < i) {
12                 System.out.print(s: "");
13                 j++;
14             }
15             i++;
16         }
17     }
18 }
19
20
```

The terminal output shows the program running with `n = 5`:

```
PS D:\PraktikumDaspro\PraktikumDaspro\Jobsheet8> java -Xmx1024m -Xms128m Triangle09
Masukkan nilai n = 5
*****
PS D:\PraktikumDaspro\PraktikumDaspro\Jobsheet8>
```

2. Menambahkan `System.out.println();` diatas `i++`
3. Variabel i berfungsi sebagai counter yang akan mereset variabel j setiap satu iterasi
Variabel j berfungsi sebagai counter yang akan mencetak Bintang. Ketika $i = 1$ maka variabel j akan di set ke 0 dan mencetak Bintang sebanyak 1 dan terus dijalankan sampai $i \leq n$

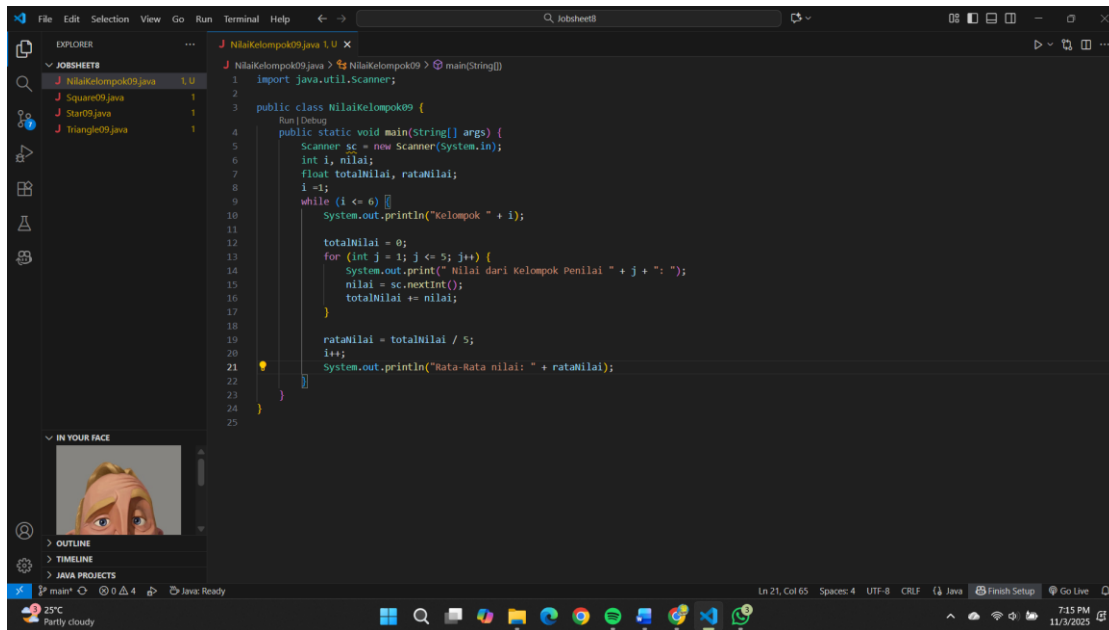
4. Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan 3”



Percobaan 4

Dalam pembelajaran kolaboratif, dosen menerapkan sistem peer review untuk menilai tugas kelompok. Setiap kelompok akan dinilai oleh beberapa kelompok lain sebagai bentuk umpan balik sejawat. Untuk setiap kelompok akan dinilai oleh beberapa kelompok lainnya. Pada percobaan ini, program akan memproses nilai untuk kelas yang berjumlah 6 kelompok. Masing-masing kelompok akan dinilai oleh 5 kelompok yang lain, dalam skala 1-100. Program akan menghitung nilai akhir setiap kelompok sebagai rata-rata dari semua penilai. Berikut adalah flowchart yang menunjukkan algoritma dari penyelesaian masalah tersebut.

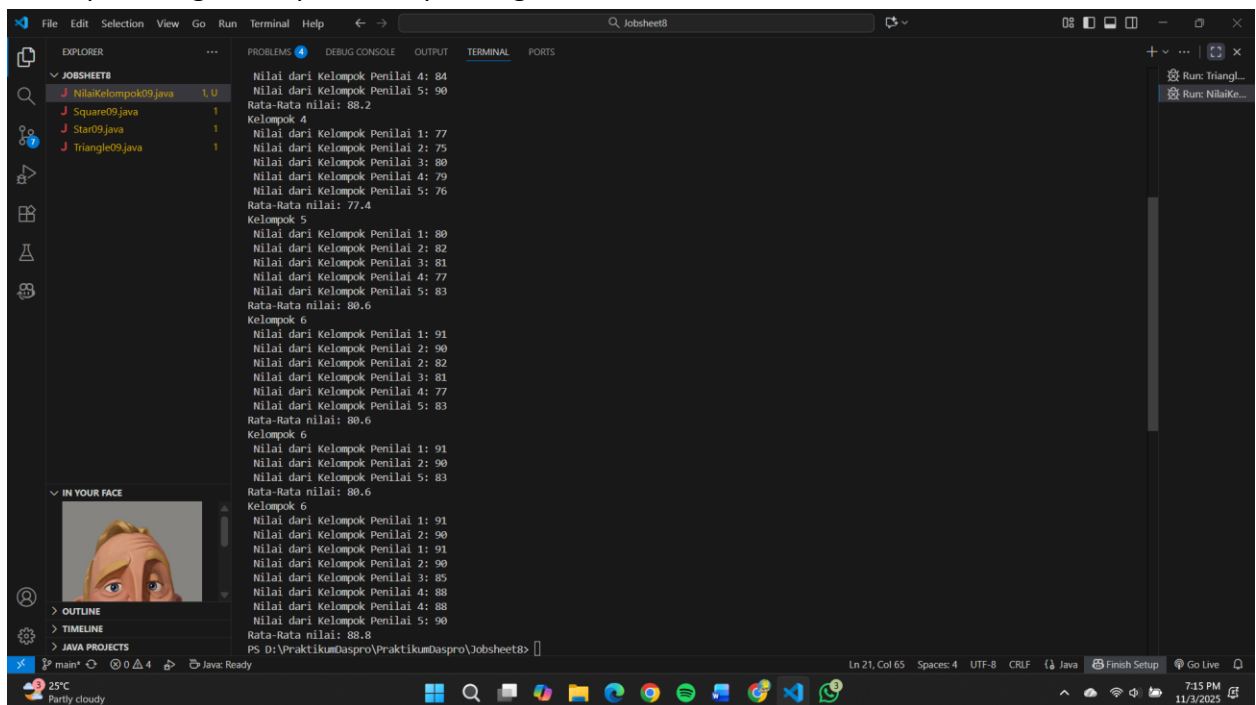
1. Buat file baru NilaiKelompokNoPresensi.java
2. Buatlah struktur dasar program Java yang terdiri dari fungsi main().
3. Tambahkan library Scanner di bagian atas (luar) class
4. Buat deklarasi Scanner dengan nama variabel sc di dalam fungsi main()
5. Deklarasikan variabel sesuai pada flowchart
6. Buat struktur perulangan untuk inner loop menggunakan for untuk memasukkan 5 nilai dari masing-masing kelompok penilai
7. Di dalam perulangan for tersebut, tambahkan perintah untuk memasukkan 5 nilai dari kelompok penilai dan menghitung total nilainya. Jangan lupa untuk memberikan nilai awal 0 pada totalNilai sebelum perulangan.
8. Setelah proses perulangan memasukkan 5 nilai dan diperoleh total nilai, maka hitung rata-rata nilai dengan rumus
9. Selanjutnya adalah mengulang proses memasukkan nilai tersebut untuk 6 kelompok. Buat struktur perulangan untuk outer loop menggunakan while.
10. Tambahkan narasi keterangan untuk kebutuhan masukan dan luaran



```
1 import java.util.Scanner;
2
3 public class NilaiKelompok09 {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         int i, nilai;
7         float totalNilai, rataNilai;
8         i = 1;
9         while (i <= 6) {
10             System.out.println("Kelompok " + i);
11
12             totalNilai = 0;
13             for (int j = 1; j <= 5; j++) {
14                 System.out.print(" Nilai dari Kelompok Penilai " + j + ": ");
15                 nilai = sc.nextInt();
16                 totalNilai += nilai;
17             }
18
19             rataNilai = totalNilai / 5;
20             i++;
21             System.out.println("Rata-Rata nilai: " + rataNilai);
22         }
23     }
24 }
25
```

11. Compile dan run program.

12. Amati hasilnya, apakah program telah menjalankan perintah memasukkan 5 nilai untuk 6 kelompok dengan tampilan serupa dengan di bawah ini:



```
Nilai dari Kelompok Penilai 1: 84
Nilai dari Kelompok Penilai 2: 90
Rata-Rata nilai: 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
Rata-Rata nilai: 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
Rata-Rata nilai: 80.6
Kelompok 6
Nilai dari Kelompok Penilai 1: 91
Nilai dari Kelompok Penilai 2: 90
Nilai dari Kelompok Penilai 3: 81
Nilai dari Kelompok Penilai 4: 77
Nilai dari Kelompok Penilai 5: 83
Rata-Rata nilai: 80.6
Kelompok 6
Nilai dari Kelompok Penilai 1: 91
Nilai dari Kelompok Penilai 2: 90
Nilai dari Kelompok Penilai 3: 81
Nilai dari Kelompok Penilai 4: 77
Nilai dari Kelompok Penilai 5: 83
Rata-Rata nilai: 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
Rata-Rata nilai: 88.8
PS D:\PraktikumDaspro\PraktikumDaspro\Jobsheet8>
```

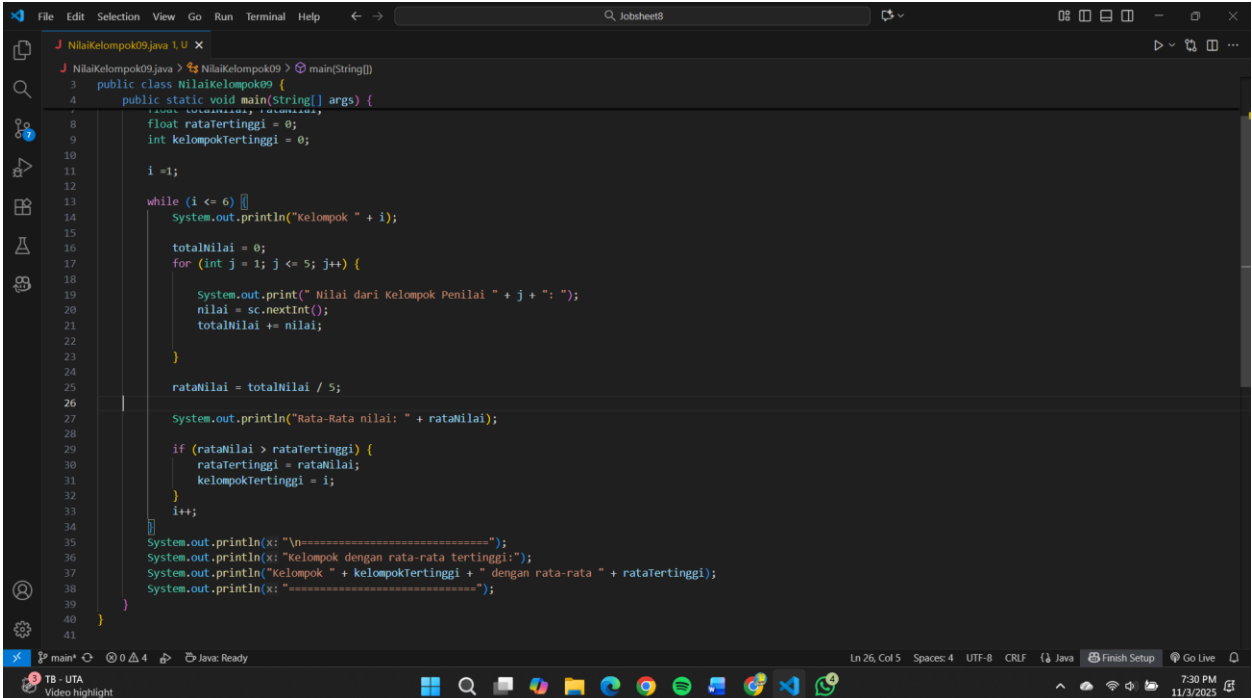
Pertanyaan

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.
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"

Jawaban

1. Variabel totalNilai diset 0 di awal iterasi kemudian user melakukan input nilai untuk setiap kelompok yang kemudian pada program totalNilai += nilai akan melakukan penjumlahan seluruh nilai. Pertama nilai 0 yang diset di awal iterasi akan ditambahkan dengan nilai pertama dan menyimpan kembali pada variabel totalNilai. Kemudian penjumlahan dilakukan lagi hingga seluruh nilai pada kelompok tersebut selesai ditotal. Kemudian setelah selesai ditotal, variabel akan di set kembali ke 0 untuk kelompok selanjutnya sehingga nilai dari kelompok sebelumnya tidak tercampur dengan kelompok lain. Kemudian kenapa inialisasi total = 0 dilakukan didalam outer loop dan bukan diluar outer loop adalah karena jika inialisasi dilakukan di luar outer loop, maka nilai tidak akan di set kembali ke 0 pada setiap iterasi, sehingga nilai tercampur antar kelompok.

2.



```
1  public class NilaiKelompok09 {
2      public static void main(String[] args) {
3          Scanner sc = new Scanner(System.in);
4          float rataTertinggi = 0;
5          int kelompokTertinggi = 0;
6
7          while (true) {
8              System.out.println("Kelompok " + i);
9
10             totalNilai = 0;
11             for (int j = 1; j <= 5; j++) {
12                 System.out.print("Nilai dari Kelompok Penilai " + j + ": ");
13                 nilai = sc.nextInt();
14                 totalNilai += nilai;
15             }
16
17             rataNilai = totalNilai / 5;
18
19             System.out.println("Rata-Rata nilai: " + rataNilai);
20
21             if (rataNilai > rataTertinggi) {
22                 rataTertinggi = rataNilai;
23                 kelompokTertinggi = i;
24             }
25             i++;
26
27             System.out.println("\n=====");
28             System.out.println("Kelompok dengan rata-rata tertinggi:");
29             System.out.println("Kelompok " + kelompokTertinggi + " dengan rata-rata " + rataTertinggi);
30             System.out.println("=====");
31         }
32     }
33 }
```

3.

```
1 public class NilaiKelompok09 {
2     public static void main(String[] args) {
3         float rataTertinggi = 0;
4         int kelompokTertinggi = 0;
5
6         i = 1;
7
8         while (i <= 6) {
9             System.out.println("Kelompok " + i);
10
11             totalNilai = 0;
12             for (int j = 1; j <= 5; j++) {
13                 System.out.print(" Nilai dari Kelompok Penilai " + j + ": ");
14                 nilai = sc.nextInt();
15                 totalNilai += nilai;
16             }
17
18             rataNilai = totalNilai / 5;
19
20             System.out.println("Rata-Rata nilai: " + rataNilai);
21
22             if (rataNilai > rataTertinggi) {
23                 rataTertinggi = rataNilai;
24                 kelompokTertinggi = i;
25             }
26             i++;
27         }
28         System.out.println(x: "\n=====");
29         System.out.println(x: "Kelompok dengan rata-rata tertinggi:");
30         System.out.println("Kelompok " + kelompokTertinggi + " dengan rata-rata " + rataTertinggi);
31         System.out.println(x: "=====");
32     }
33 }
```

The screenshot shows an IDE with a Java file named 'NilaiKelompok09.java'. The code implements a loop to process six groups. For each group, it prints the group number and then prompts for five individual scores. It calculates the average for each group and keeps track of the highest average and the corresponding group number. The program uses a Scanner object 'sc' to read input. The output shows the average for each group and the final group with the highest average.