

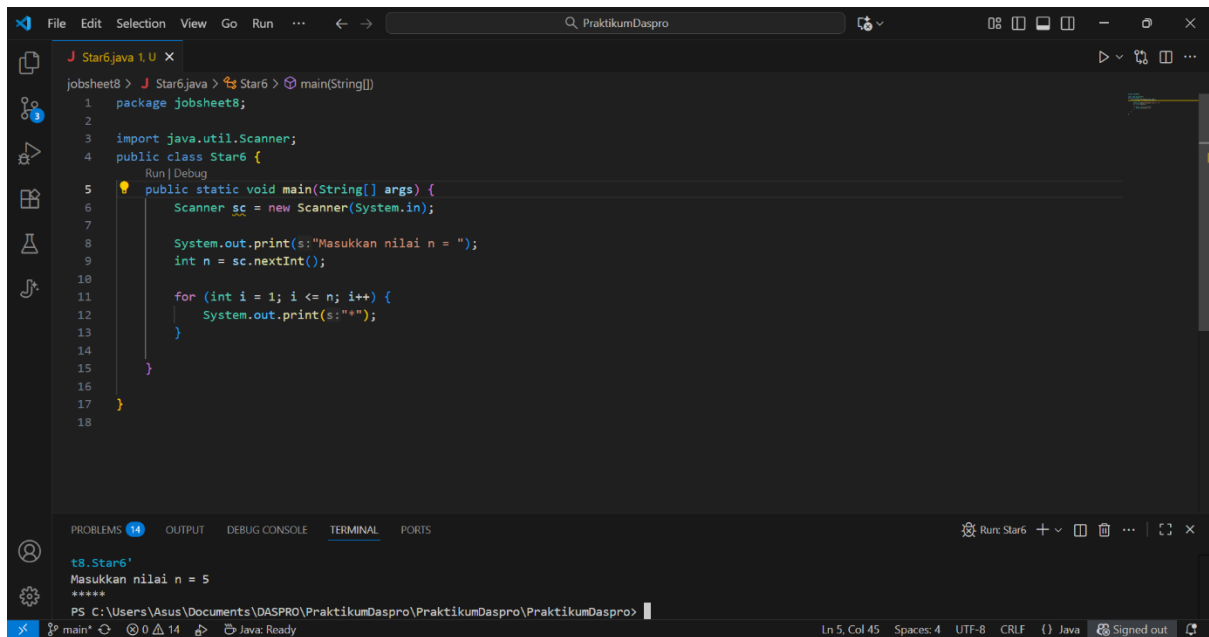
Nama : Deswita Khansa Rafifah

NIM : 254107020151

Kelas : TI_1G

JOBSHEET 8

Percobaan 1



```
1 package jobsheet8;
2
3 import java.util.Scanner;
4 public class Star6 {
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 i = 1; i <= n; i++) {
12             System.out.print(s:"");
13         }
14     }
15 }
16
17
18
```

PROBLEMS: 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

t8.Star6*

Masukkan nilai n = 5

PS C:\Users\Asus\Documents\DASPRO\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 <= 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 $i <= n$ 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.

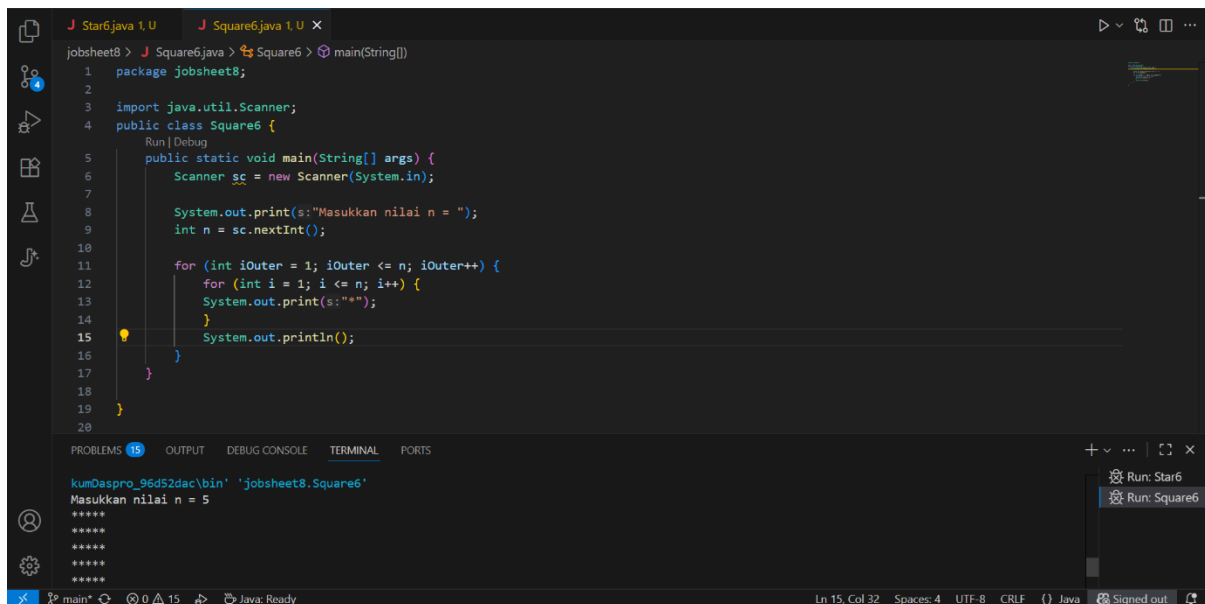
4. Jika pada perulangan for, kondisi step $i++$ diubah menjadi $i--$ apa akibatnya? Mengapa bisa demikian?

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



```
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
20
```

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 inialisasi **iOuter=1**. Kemudian perhatikan perulangan dalam. Jika pada sintaks **for**, inialisasi **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.

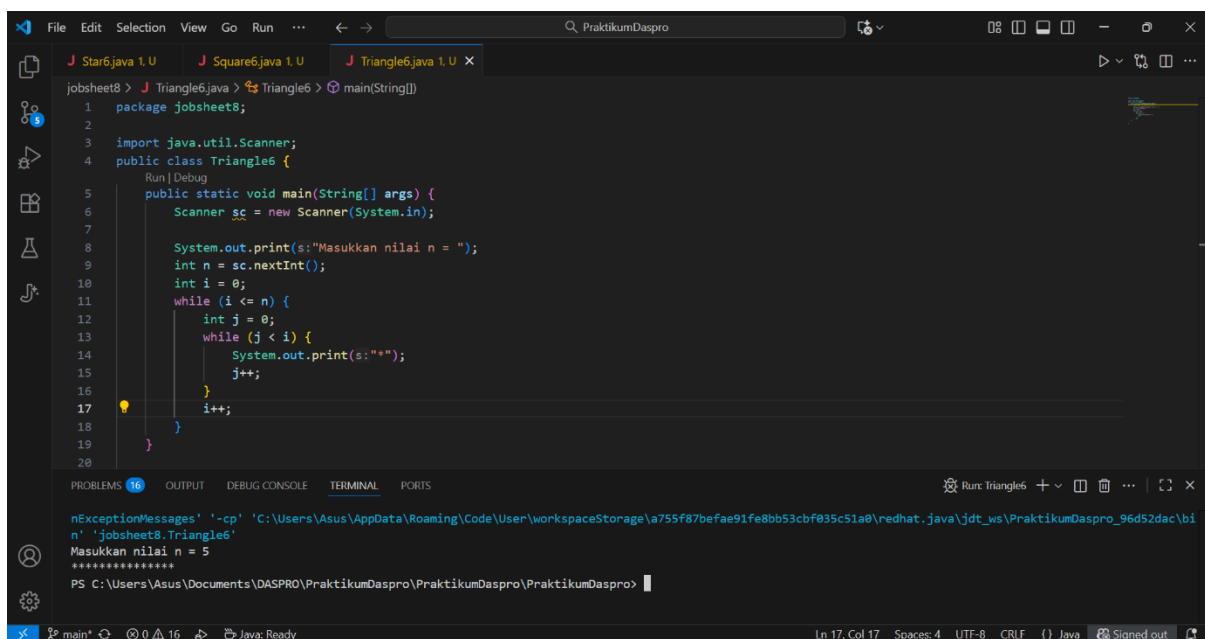
3. Apakah perbedaan kegunaan antara perulangan luar dengan perulangan yang berada di dalamnya?

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



```
File Edit Selection View Go Run ... < > PraktikumDaspro
J Star6.java 1.U J Square6.java 1.U J Triangle6.java 1.U X
jobsheet8 > J Triangle6.java > Triangle6 > main(String[])
1 package jobsheet8;
2
3 import java.util.Scanner;
4 public class Triangle6 {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         System.out.print(s:"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(s:"*");
15                j++;
16            }
17            i++;
18        }
19    }
20}

PROBLEMS 16 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Run: Triangle6
nExceptionMessages' '-cp' 'C:\Users\Asus\AppData\Roaming\Code\User\workspaceStorage\4755f87befae91fe8bb53cbf035c51a0\redhat.java\jdt_ws\PraktikumDaspro_96d52dac\bin' 'jobsheet8.Triangle6'
Masukkan nilai n = 5
*****
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.

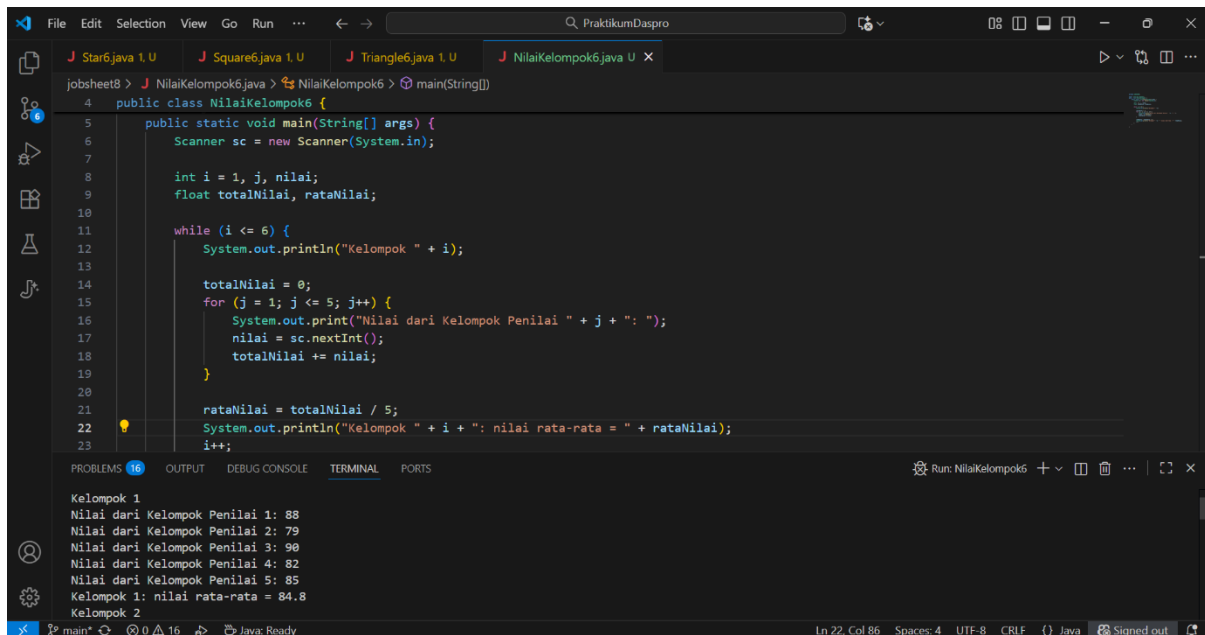
3. Jelaskan peran masing-masing variabel **i** dan **j** dalam program ini. Mengapa **j** di-set ulang ke 0 di awal setiap iterasi outer loop? Apa yang akan terjadi jika **j** tidak di-reset?

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



```
public class NilaiKelompok6 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int i = 1, j, nilai;
        float totalNilai, rataNilai;

        while (i <= 6) {
            System.out.println("Kelompok " + i);

            totalNilai = 0;
            for (j = 1; j <= 5; j++) {
                System.out.print("Nilai dari Kelompok Penilai " + j + ": ");
                nilai = sc.nextInt();
                totalNilai += nilai;
            }

            rataNilai = totalNilai / 5;
            System.out.println("Kelompok " + i + ": nilai rata-rata = " + rataNilai);
            i++;
        }
    }
}
```

Output:

```
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
```

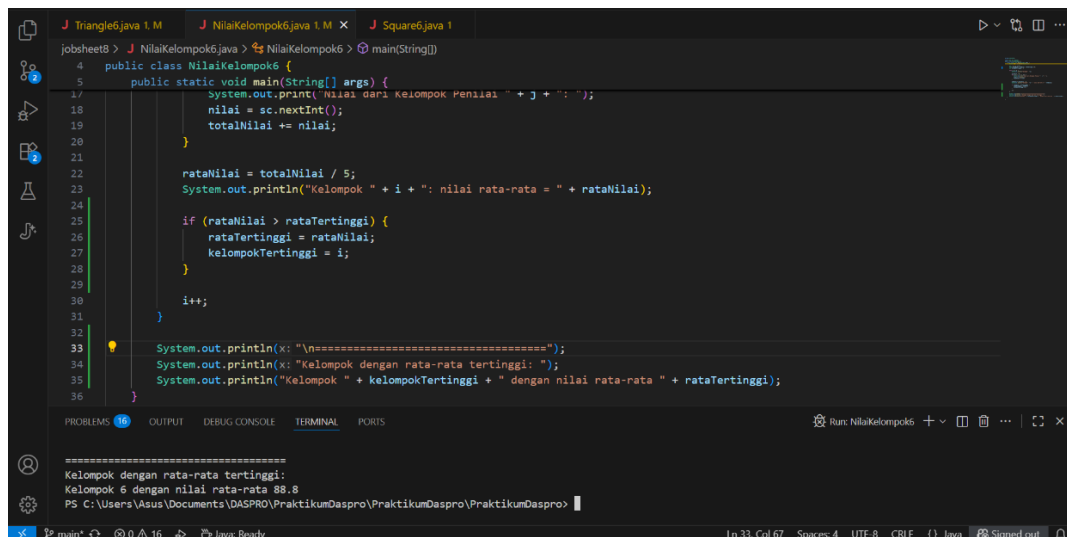
Pertanyaan

1. Jelaskan apa yang terjadi pada variabel **totalNilai** di setiap iterasi *outer loop* dan mengapa inisialisasinya ($\text{total} = 0$) 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:



```
public class NilaiKelompok6 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int i = 1, j, nilai;
        float totalNilai, rataNilai;
        float rataTertinggi = 0;
        int kelompokTertinggi = 0;

        while (i <= 6) {
            System.out.println("Nilai dari Kelompok Penilai " + j + ": ");
            nilai = sc.nextInt();
            totalNilai += nilai;

            rataNilai = totalNilai / 5;
            System.out.println("Kelompok " + i + ": nilai rata-rata = " + rataNilai);

            if (rataNilai > rataTertinggi) {
                rataTertinggi = rataNilai;
                kelompokTertinggi = i;
            }

            i++;
        }

        System.out.println("\n=====");
        System.out.println("Kelompok dengan rata-rata tertinggi: ");
        System.out.println("Kelompok " + kelompokTertinggi + " dengan nilai rata-rata " + rataTertinggi);
    }
}
```

Output:

```
=====
Kelompok dengan rata-rata tertinggi:
Kelompok 6 dengan nilai rata-rata 88.8
PS C:\Users\Asus\Documents\Daspro\PraktikumDaspro>
```

Tugas 1

The screenshot shows an IDE with a Java file named `Tugas1.java`. The code defines a class `Tugas1` with a `main` method that calculates the sum of squares of the first 5 natural numbers. The output in the terminal shows the calculation for `n = 1` to `n = 5`.

```

package jobsheet8;

import java.util.Scanner;

public class Tugas1 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int n, i, j, jumlah;

        for (n = 1; n <= 5; n++) {
            jumlah = 0;
            System.out.print("n = " + n + " jumlah kuadrat = ");

            for (i = 1; i <= n; i++) {
                jumlah += i * i;
                System.out.print(i + " ");
                if (i < n) {
                    System.out.print(": " + " ");
                }
            }

            System.out.println(jumlah);
        }
    }
}

```

The terminal output shows the calculation for `n = 1` to `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

```

Tugas 2

The screenshot shows an IDE with a Java file named `Tugas2.java`. The code is as follows:

```

1 package jobsheet8;
2
3 import java.util.Scanner;
4 public class Tugas2 {
5     Run | Debug
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8
9         System.out.print(s: "Masukkan nilai n (minimal 3): ");
10        int n = sc.nextInt();
11
12        if (n < 3) {
13            System.out.println(x: "Nilai n minimal 3!");
14            return;
15        }
16
17        for (int i = 1; i <= n; i++) {
18            for (int j = 1; j <= n; j++) {
19                if (i == 1 || i == n || j == 1 || j == n) {
20                    System.out.print(n + " ");
21                }
22            }
23        }
24    }
25 }

```

The terminal output shows the program running and the user inputting 3, resulting in three '3 3' lines being printed:

```

t8.Tugas2'
Masukkan nilai n (minimal 3): 3
3 3
3 3
3 3

```

The status bar at the bottom indicates the file is located at `PS C:\Users\Asus\Documents\Daspro\PraktikumDaspro\PraktikumDaspro\PraktikumDaspro>` and the Java version is 11.0.12.

Tugas 3

The screenshot displays an IDE with a Java file named `Tugas3.java`. The code implements a cashier system with the following logic:

- It prompts the user for the number of items (`jumlahCabang`).
- A loop processes each item, asking for the item name (`itemDipesan`) and its price (`itemDipesan`).
- It calculates the total item price (`totalItemCabang`).
- It prompts for the number of customers (`jumlahPelangganCabang`).
- A loop processes each customer, asking for the number of items they want to buy (`itemDipesan`).
- It calculates the total item price for each customer and adds it to the total item price.
- Finally, it prints the total item price and the total number of customers.

The terminal output shows the program's execution with sample input:

```

--- 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:

```