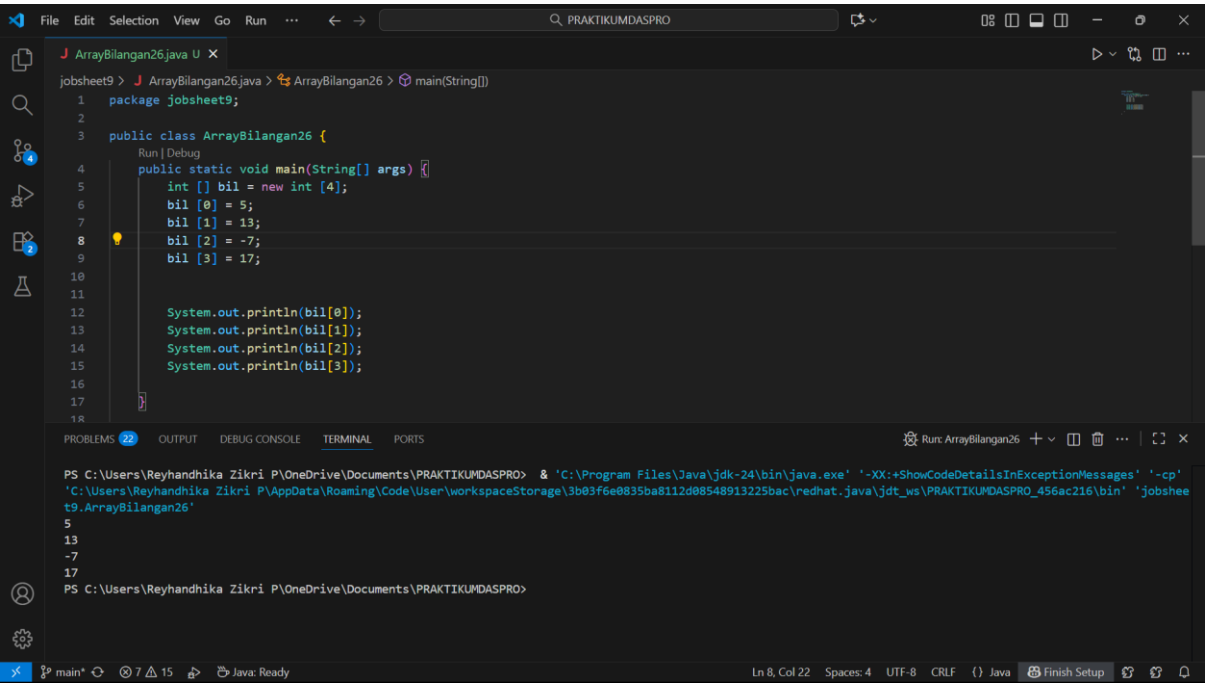


Nama : Reyhandhika Zikri Prijadi

Kelas : TI_1G

Nim : 254107020219

PERCOBAAN 1



PERTANYAAN 1

1. Jika isi masing-masing elemen array bil diubah dengan angka 5.0, 12867, 7.5, 2000000.

Apa yang terjadi? Mengapa bisa demikian?

⇒ Akan ERROR karena, jika nilai- nilai tersebut dimasukkan ke dalam array int , maka akan terjadi erorr karena angka 5,0 dan 7,5. Dikarenakan kedua nilai tersebut Adalah bilangan pecahan (tipe double), sedangkan array bil dideklarasikan sebagai int[] yang hanya bisa menyimpan bilangan bulat.

2. Modifikasi kode program di atas dengan melakukan inisialisasi elemen array sekaligus pada saat deklarasi array.

⇒ dengan cara `int[] bil = {5, 13, -7, 17};` array langsung memiliki nilai awal sejak dideklarasikan. Dan hasil keluaran program tetap sama

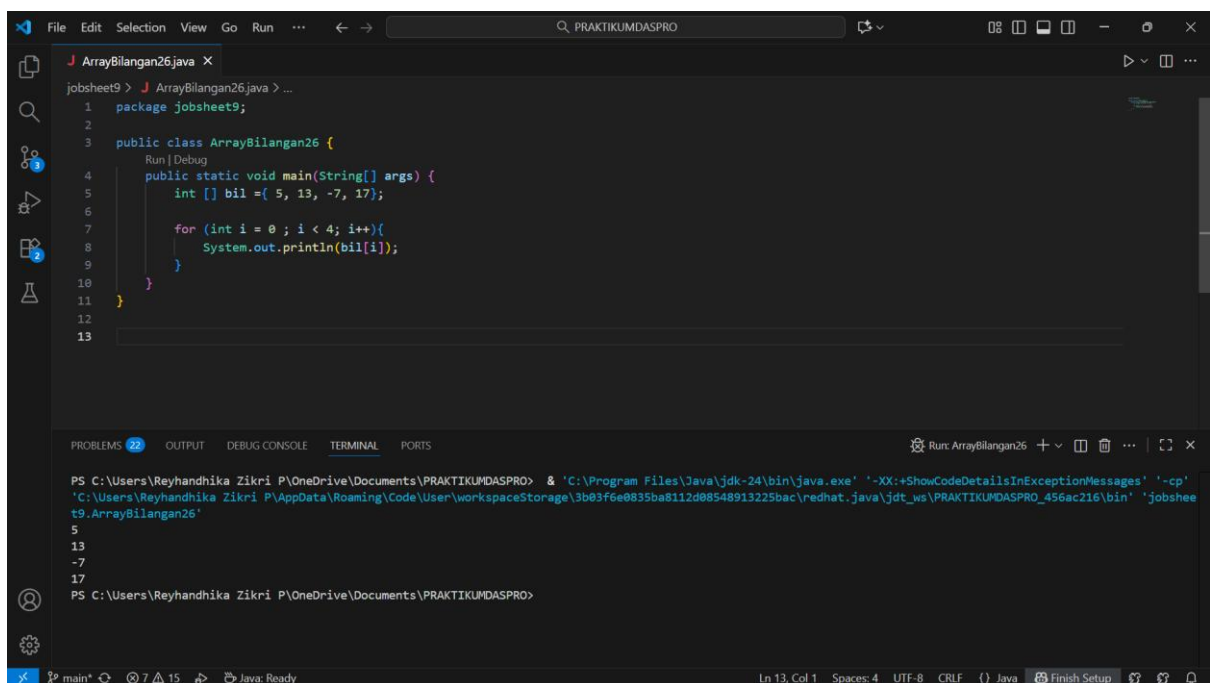
3. Ubah statement pada langkah No 4 menjadi seperti berikut ...

```

for (int i = 0; i < 4; i++){
    System.out.println(bil[i]);
}

```

Apa keluaran dari program? Jelaskan maksud dari statement tersebut.

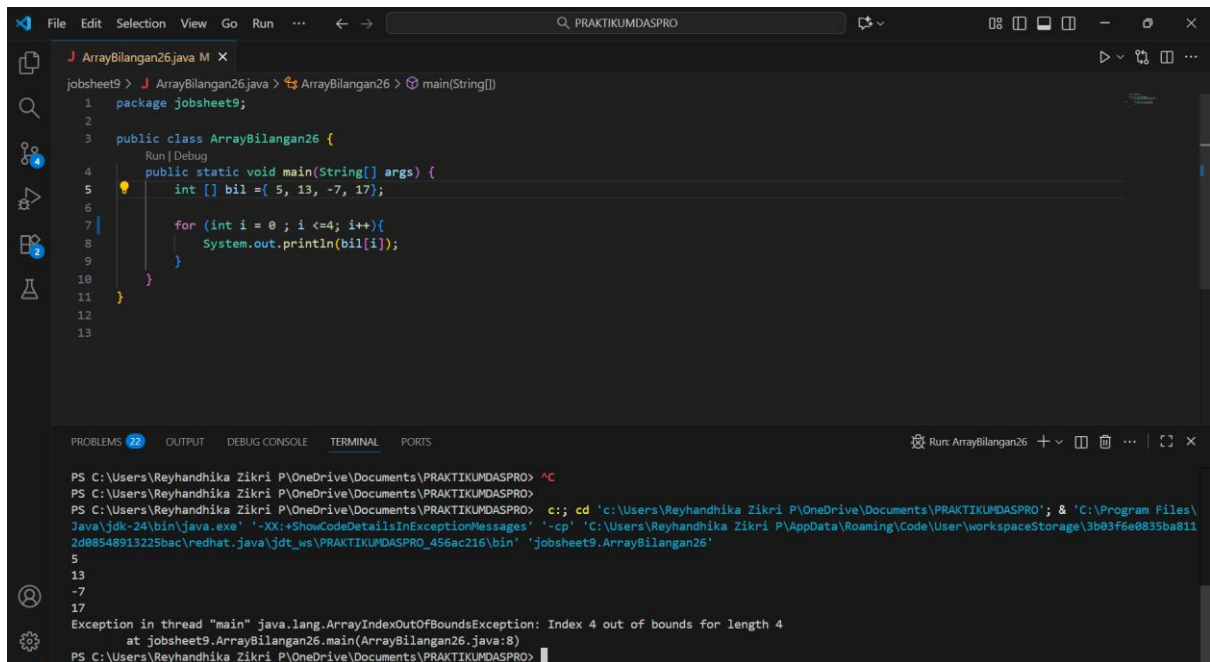


```
1 package jobsheet9;
2
3 public class ArrayBilangan26 {
4     public static void main(String[] args) {
5         int [] bil = { 5, 13, -7, 17};
6
7         for (int i = 0 ; i < 4; i++){
8             System.out.println(bil[i]);
9         }
10    }
11 }
12
13
```

```
PS C:\Users\Reyhandhika Zikri P\OneDrive\Documents\PRAKTIKUMDASPRO> & 'C:\Program Files\Java\jdk-24\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Reyhandhika Zikri P\AppData\Roaming\Code\User\workspaceStorage\3b03f6e0835ba8112d88548913225bac\redhat_java\jdt_ws\PRAKTIKUMDASPRO_456ac216\bin' 'jobsheet9.ArrayBilangan26'
5
13
-7
17
PS C:\Users\Reyhandhika Zikri P\OneDrive\Documents\PRAKTIKUMDASPRO>
```

⇒ for (int i = 0; i < 4; i++) program ini akan mengulang sebanyak 4kali, mulai dari indeks 0 sampai 3, pada setiap pengulangan, bil[i] menampilkan isi array sesuai indeksinya. Di loop ini digunakan agar program tidak perlu menulis System.out.println() berkali-kali secara manual, tetapi dapat menampilkan seluruh isi array dengan lebih efisien dan dinamis.

4. Jika kondisi pada statement for-loop di atas diubah menjadi: i <= 4, apa keluaran dari program? Mengapa demikian?



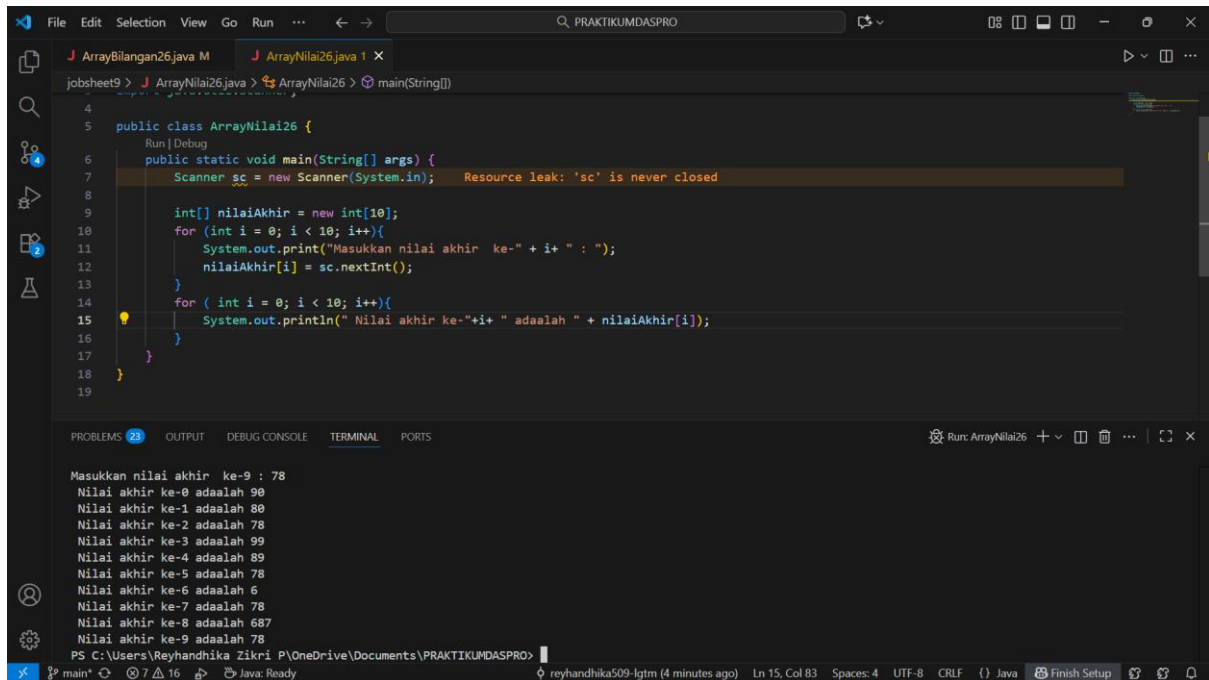
```
1 package jobsheet9;
2
3 public class ArrayBilangan26 {
4     public static void main(String[] args) {
5         int [] bil = { 5, 13, -7, 17};
6
7         for (int i = 0 ; i <= 4; i++){
8             System.out.println(bil[i]);
9         }
10    }
11 }
12
13
```

```
PS C:\Users\Reyhandhika Zikri P\OneDrive\Documents\PRAKTIKUMDASPRO> ^C
PS C:\Users\Reyhandhika Zikri P\OneDrive\Documents\PRAKTIKUMDASPRO>
PS C:\Users\Reyhandhika Zikri P\OneDrive\Documents\PRAKTIKUMDASPRO> c:; cd 'c:\Users\Reyhandhika Zikri P\OneDrive\Documents\PRAKTIKUMDASPRO'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Reyhandhika Zikri P\AppData\Roaming\Code\User\workspaceStorage\3b03f6e0835ba8112d88548913225bac\redhat_java\jdt_ws\PRAKTIKUMDASPRO_456ac216\bin' 'jobsheet9.ArrayBilangan26'
5
13
-7
17
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 4
    at jobsheet9.ArrayBilangan26.main(ArrayBilangan26.java:8)
PS C:\Users\Reyhandhika Zikri P\OneDrive\Documents\PRAKTIKUMDASPRO>
```

⇒ Array bil hanya memiliki indeks dari 0 sampai 3(total 4 elemen). Saat i mencapai 4, program mencoba mengakses bil[4], padahal elemen ke-5 tidak ada akibatnya, java akan menghentikan eksekusi program dan menampilkan pesan kesalahan melampaui batas ukuran array. Kondisi pada perulangan harus berhenti sebelum mencapai jumlah elemen array yaitu i < bil.length.

5. Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan 1”

PERCOBAAN 2



The screenshot shows an IDE with a Java file named `ArrayNilai26.java`. The code defines a class `ArrayNilai26` with a `main` method. It uses a `Scanner` to read 10 integers into an array `nilaiAkhir`. The output window shows the program's execution, where it prompts for 10 values and prints them back. A warning message "Resource leak: 'sc' is never closed" is visible in the IDE's interface.

```
public class ArrayNilai26 {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        int[] nilaiAkhir = new int[10];  
        for (int i = 0; i < 10; i++){  
            System.out.print("Masukkan nilai akhir ke-" + i + " : ");  
            nilaiAkhir[i] = sc.nextInt();  
        }  
        for (int i = 0; i < 10; i++){  
            System.out.println("Nilai akhir ke-" + i + " adalah " + nilaiAkhir[i]);  
        }  
    }  
}
```

Output:

```
Masukkan nilai akhir ke-0 : 78  
Nilai akhir ke-0 adalah 78  
Masukkan nilai akhir ke-1 : 90  
Nilai akhir ke-1 adalah 90  
Masukkan nilai akhir ke-2 : 80  
Nilai akhir ke-2 adalah 80  
Masukkan nilai akhir ke-3 : 78  
Nilai akhir ke-3 adalah 78  
Masukkan nilai akhir ke-4 : 99  
Nilai akhir ke-4 adalah 99  
Masukkan nilai akhir ke-5 : 89  
Nilai akhir ke-5 adalah 89  
Masukkan nilai akhir ke-6 : 78  
Nilai akhir ke-6 adalah 78  
Masukkan nilai akhir ke-7 : 6  
Nilai akhir ke-7 adalah 6  
Masukkan nilai akhir ke-8 : 78  
Nilai akhir ke-8 adalah 78  
Masukkan nilai akhir ke-9 : 687  
Nilai akhir ke-9 adalah 687
```

PERTANYAAN 2

1. Ubah statement pada langkah nomor 5 menjadi seperti berikut ini:

Jalankan program. Apakah terjadi perubahan? Mengapa demikian?

```
for (int i = 0; i < nilaiAkhir.length; i++){  
    System.out.print("Masukkan nilai akhir ke-" + i + " : ");  
    nilaiAkhir[i] = sc.nextInt();  
}
```

⇒ tidak ada perubahan tampilan output maupun cara kerja program di karenakan `nilaiAkhir.length` Adalah Panjang array, yaitu jumlah elemen yang dimilikinya. Karena pada program array `nilaiAkhir` berukuran 10, maka `nilaiAkhir.length` bernilai 10. Jadi perulangan `for (int i = 0; i < nilaiAkhir.length; i++)` artinya sama dengan `for (int i = 0; i < 10; i++)`.

2. Apa yang dimaksud dengan kondisi: `i < nilaiAkhir.length` ?

⇒ Kondisi `i < nilaiAkhir.length` berarti **selama nilai indeks i masih lebih kecil daripada jumlah elemen array**, maka perulangan akan terus dijalankan. kondisi ini memastikan program tidak mengakses indeks di luar batas array, sehingga mencegah error seperti `ArrayIndexOutOfBoundsException`.

3. Ubah statement pada langkah nomor 6 menjadi seperti berikut ini, sehingga program hanya menampilkan nilai Mahasiswa yang lulus saja (yaitu mahasiswa yang memiliki nilai > 70):

```
for (int i = 0; i < nilaiAkhir.length; i++){  
    if (nilaiAkhir[i] > 70){  
        System.out.println("Mahasiswa ke-" + i + " lulus!");  
    }  
}
```

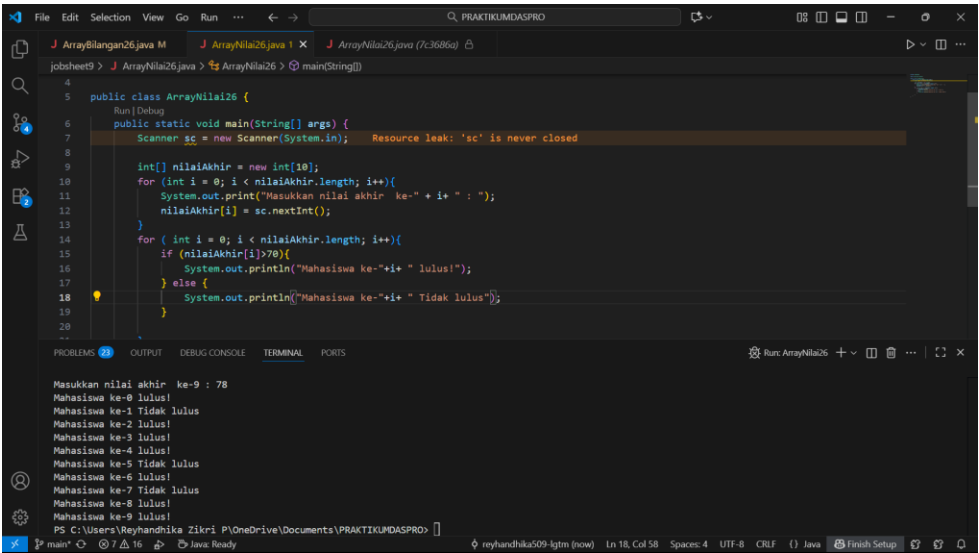
Jalankan program dan jelaskan alur program!

⇒ kode ini menampilkan hanya mahasiswa yang nilainya lebih dari 70, sehingga membantu dalam memfilter data kelulusan secara otomatis.

4. Modifikasi program agar menampilkan status kelulusan semua mahasiswa berdasarkan nilai, yaitu dengan menampilkan status mana mahasiswa yang lulus dan tidak lulus, seperti ilustrasi output berikut:

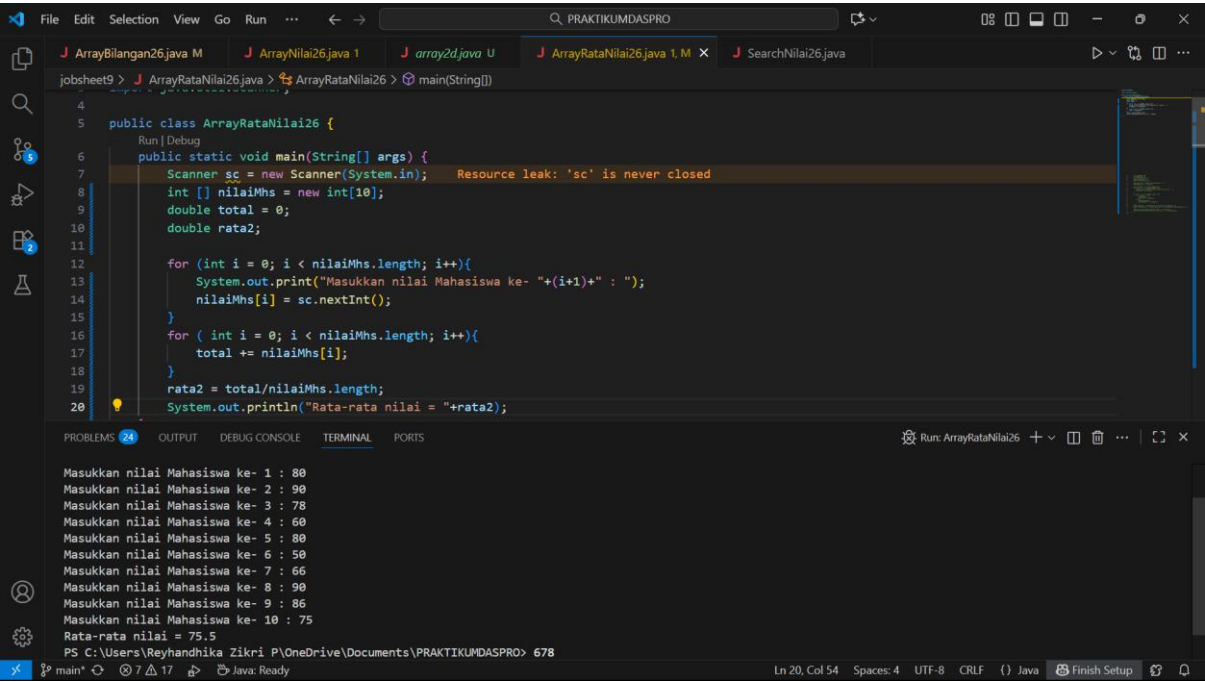
```
Masukkan nilai akhir ke-0 : 87
Masukkan nilai akhir ke-1 : 65
Masukkan nilai akhir ke-2 : 78
Masukkan nilai akhir ke-3 : 95
Masukkan nilai akhir ke-4 : 92
Masukkan nilai akhir ke-5 : 58
Masukkan nilai akhir ke-6 : 89
Masukkan nilai akhir ke-7 : 67
Masukkan nilai akhir ke-8 : 85
Masukkan nilai akhir ke-9 : 78
Mahasiswa ke-0 lulus!
Mahasiswa ke-1 tidak lulus!
Mahasiswa ke-2 lulus!
Mahasiswa ke-3 lulus!
Mahasiswa ke-4 lulus!
Mahasiswa ke-5 tidak lulus!
Mahasiswa ke-6 lulus!
Mahasiswa ke-7 tidak lulus!
Mahasiswa ke-8 lulus!
Mahasiswa ke-9 lulus!
```

JAWABAN :



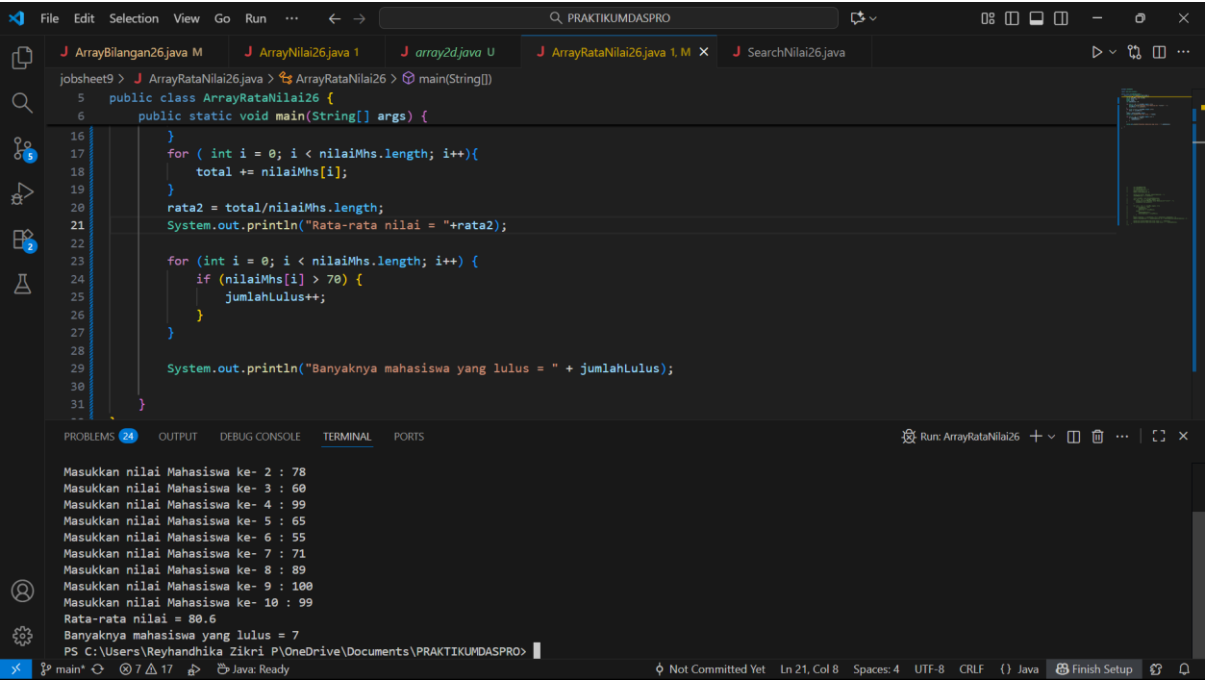
5. Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan 2”

PERCOBAAN 3



PERTANYAAN 3

1. Modifikasi kode program pada praktikum percobaan 3 di atas (ArrayRataNilaiXX.java) agar program dapat menampilkan banyaknya mahasiswa yang lulus, yaitu mahasiswa yang memiliki lebih besar dari 70 (>70).

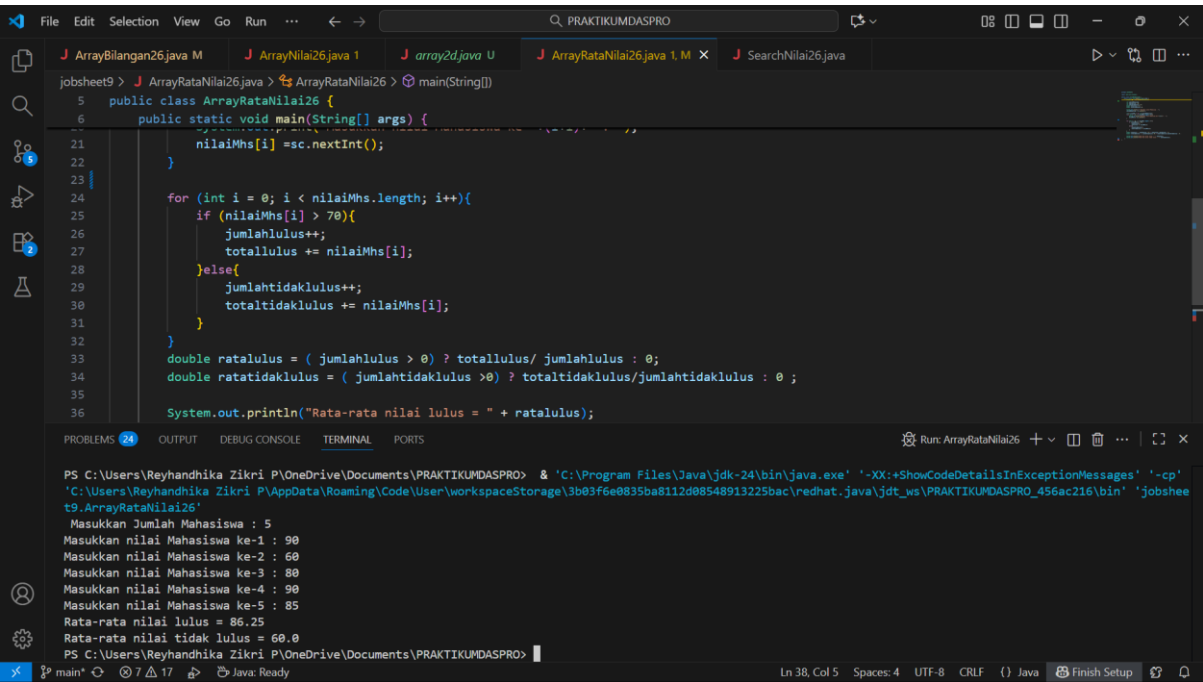


2. Modifikasi program pada praktikum percobaan 3 di atas (ArrayRataNilaiXX.java) sehingga program menerima jumlah elemen berdasarkan input dari pengguna dan mengeluarkan

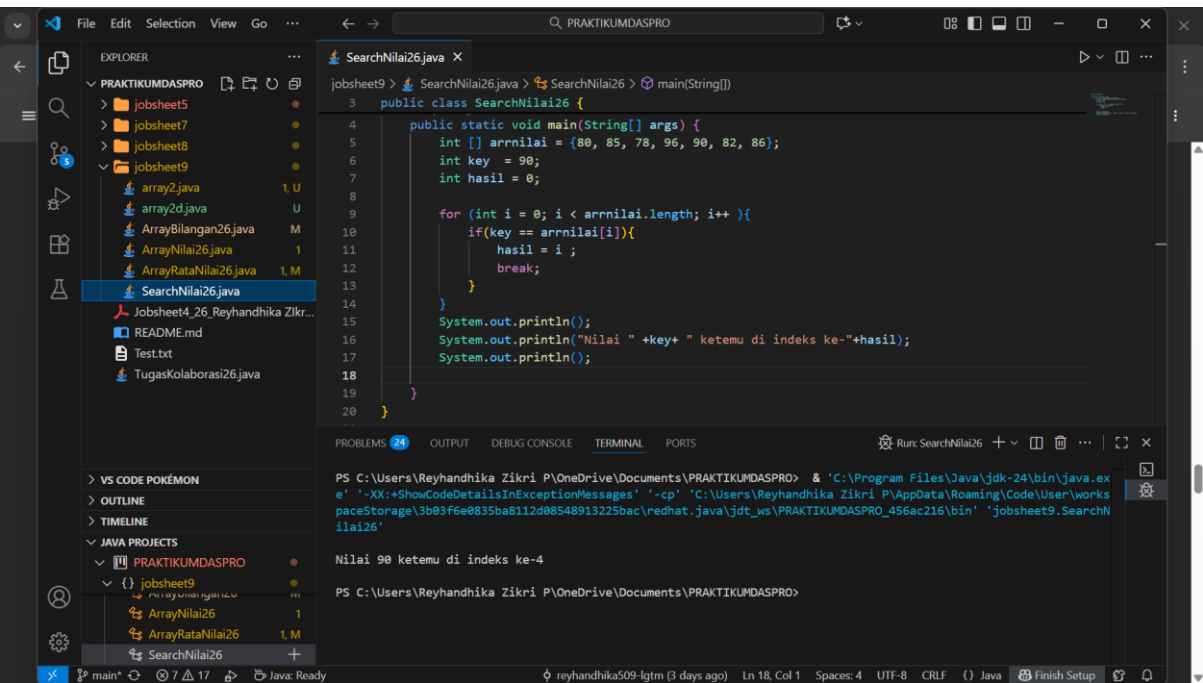
output seperti berikut ini:

```
Masukkan jumlah mahasiswa : 5
Masukkan nilai mahasiswa ke-1 : 80
Masukkan nilai mahasiswa ke-2 : 60
Masukkan nilai mahasiswa ke-3 : 90
Masukkan nilai mahasiswa ke-4 : 85
Masukkan nilai mahasiswa ke-5 : 65
Rata-rata nilai lulus = 85.0
Rata-rata nilai tidak lulus = 62.5
```

output:



PERCOBAAN 4



PERTANYAAN 4

1. Jelaskan maksud dari statement break; pada baris ke-10 kode program percobaan 4 di

atas.

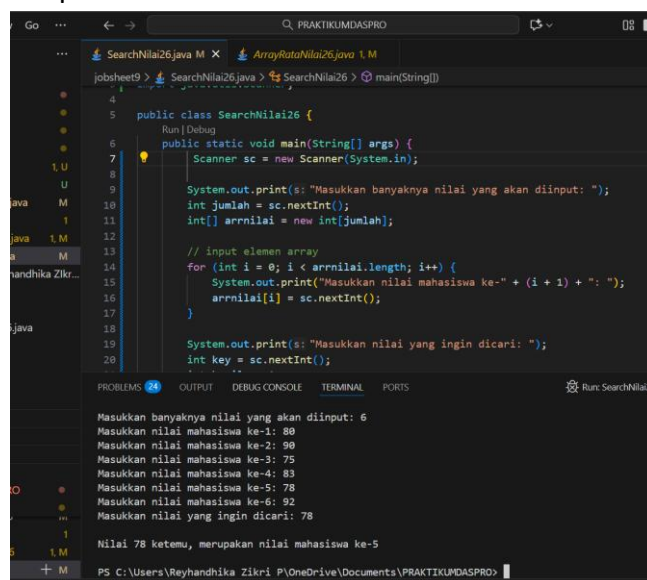
⇒ Statement break; pada barisan ke- 10 berfungsi untuk menghentikan perulangan secara langsung Ketika program sudah menentukan nilai yang di cari di dalam array. Jadi, saat kondisi if (key == arrNilai[i]) terpenuhi (artinya nilai yang dicari sama dengan nilai elemen array), program akan menyimpan posisi nilai tersebut ke variabel hasil, lalu menjalankan break; agar tidak melanjutkan perulangan ke elemen berikutnya.

2. Modifikasi kode program pada percobaan 4 di atas sehingga program dapat menerima input berupa banyaknya elemen array nilai, isi array, dan sebuah nilai (key) yang ingin dicari. Lalu cetak ke layar indeks posisi elemen dari nilai (key) yang dicari. Contoh hasil program:

```
Masukkan banyaknya nilai yang akan diinput: 6
Masukkan nilai mahasiswa ke-1: 80
Masukkan nilai mahasiswa ke-2: 90
Masukkan nilai mahasiswa ke-3: 75
Masukkan nilai mahasiswa ke-4: 83
Masukkan nilai mahasiswa ke-5: 78
Masukkan nilai mahasiswa ke-6: 92
Masukkan nilai yang ingin dicari: 78

Nilai 78 ketemu, merupakan nilai mahasiswa ke-5
```

Output:

The screenshot shows an IDE with a Java file named SearchNilai26.java. The code defines a class SearchNilai26 with a main method that uses a Scanner to take user input. It prompts for the number of values, then loops to take 6 student values (80, 90, 75, 83, 78, 92), and finally asks for a key to search (78). The output window at the bottom shows the execution results, including the search result: 'Nilai 78 ketemu, merupakan nilai mahasiswa ke-5'.

```
public class SearchNilai26 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Masukkan banyaknya nilai yang akan diinput: ");
        int jumlah = sc.nextInt();
        int[] arrnilai = new int[jumlah];

        // input elemen array
        for (int i = 0; i < arrnilai.length; i++) {
            System.out.print("Masukkan nilai mahasiswa ke-" + (i + 1) + ": ");
            arrnilai[i] = sc.nextInt();
        }

        System.out.print("Masukkan nilai yang ingin dicari: ");
        int key = sc.nextInt();

        // ... (search logic) ...

        System.out.println("Nilai 78 ketemu, merupakan nilai mahasiswa ke-5");
    }
}
```

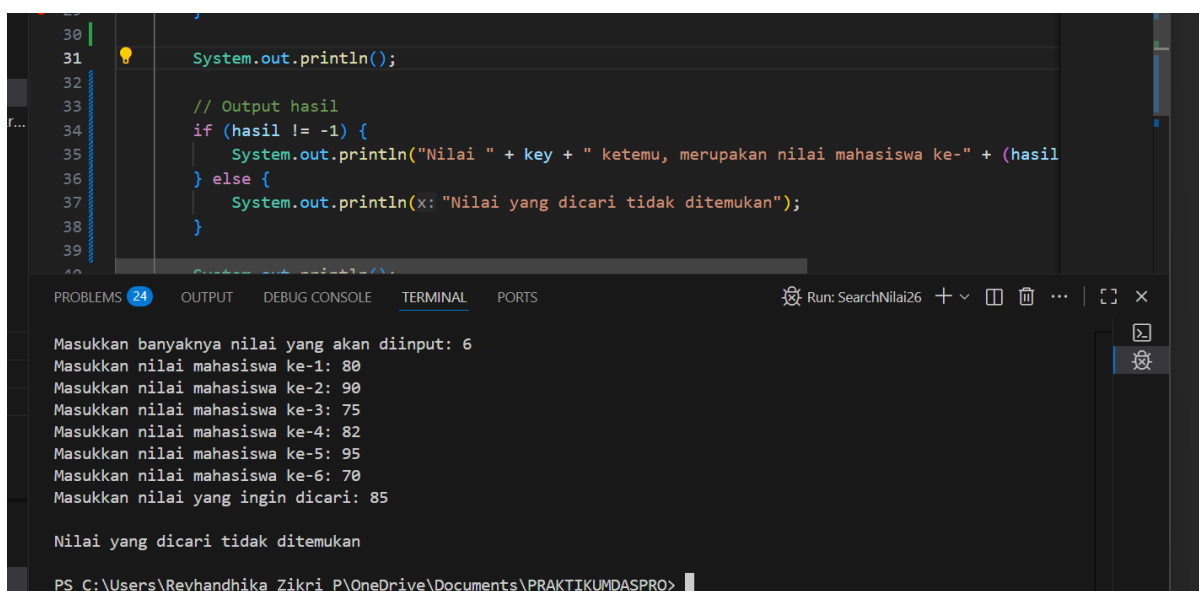
3. Modifikasi program pada percobaan 4 di atas, sehingga program akan memberikan pesan "Nilai yang dicari tidak ditemukan" jika nilai yang dicari (key) tidak ada di dalam array.

Contoh tampilan program sebagai berikut:

```
Masukkan banyaknya nilai yang akan diinput: 6
Masukkan nilai mahasiswa ke-1: 80
Masukkan nilai mahasiswa ke-2: 90
Masukkan nilai mahasiswa ke-3: 75
Masukkan nilai mahasiswa ke-4: 82
Masukkan nilai mahasiswa ke-5: 95
Masukkan nilai mahasiswa ke-6: 70
Masukkan nilai yang ingin dicari: 85
```

```
Nilai yang dicari tidak ditemukan
```

Output :

The screenshot shows a Java IDE with a code editor and a terminal window. The code editor displays a Java program that takes an array of student scores and a target score as input. It uses a loop to check each score against the target. The terminal window shows the program's execution, where the user has entered 6 for the number of students, followed by six scores (80, 90, 75, 82, 95, 70), and then 85 as the target score. The program's output is "Nilai yang dicari tidak ditemukan" (The value being searched for is not found).

```
30 |
31 | System.out.println();
32 |
33 | // Output hasil
34 | if (hasil != -1) {
35 |     System.out.println("Nilai " + key + " ketemu, merupakan nilai mahasiswa ke-" + (hasil
36 | } else {
37 |     System.out.println(x: "Nilai yang dicari tidak ditemukan");
38 | }
39 |
40 |
```

PROBLEMS 24 OUTPUT DEBUG CONSOLE TERMINAL PORTS Run: SearchNilai26

```
Masukkan banyaknya nilai yang akan diinput: 6
Masukkan nilai mahasiswa ke-1: 80
Masukkan nilai mahasiswa ke-2: 90
Masukkan nilai mahasiswa ke-3: 75
Masukkan nilai mahasiswa ke-4: 82
Masukkan nilai mahasiswa ke-5: 95
Masukkan nilai mahasiswa ke-6: 70
Masukkan nilai yang ingin dicari: 85

Nilai yang dicari tidak ditemukan

PS C:\Users\Reyhandhika Zikri P\OneDrive\Documents\PRAKTIKUMDASPRO>
```

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

4"