Nama: Kamila Zahwa (13)

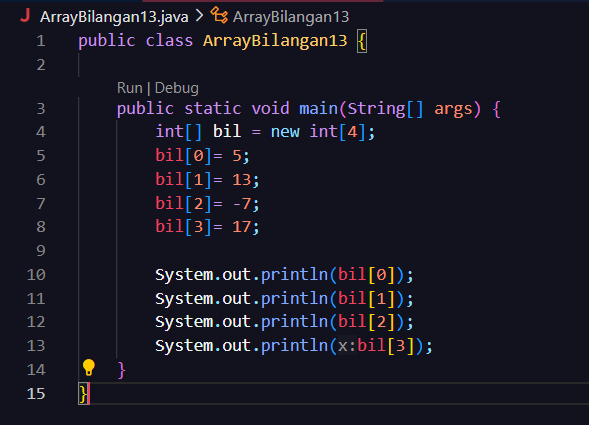
NIM: 244107020111

Kelas: TI 1D

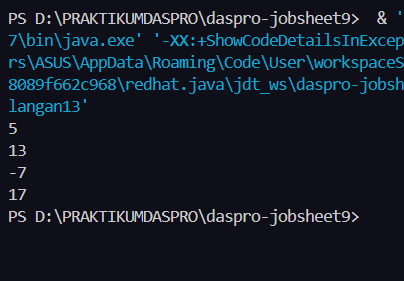
**JOBSHEET 9**

Percobaan 1

1. Buka text editor, buat class Java baru dengan nama ArrayBilanganXX.java. (XX=nomor absen)
2. Buat array bertipe integer dengan nama bil dengan kapasitas 4 elemen.
3. Isi masing-masing elemen array bil tadi dengan angka 5, 13, -7, 17.
4. Tampilkan ke layar semua isi elemennya:



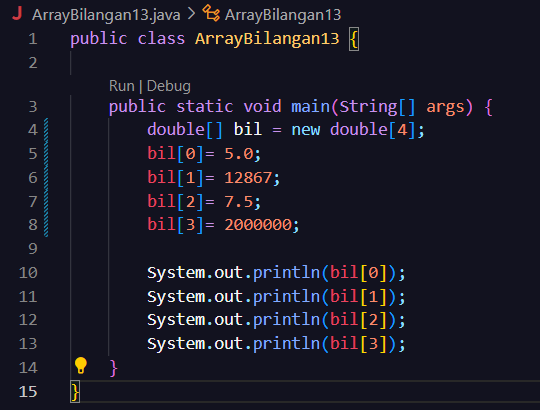
1. Cocokkan dan amati hasilnya dengan gambar berikut ini



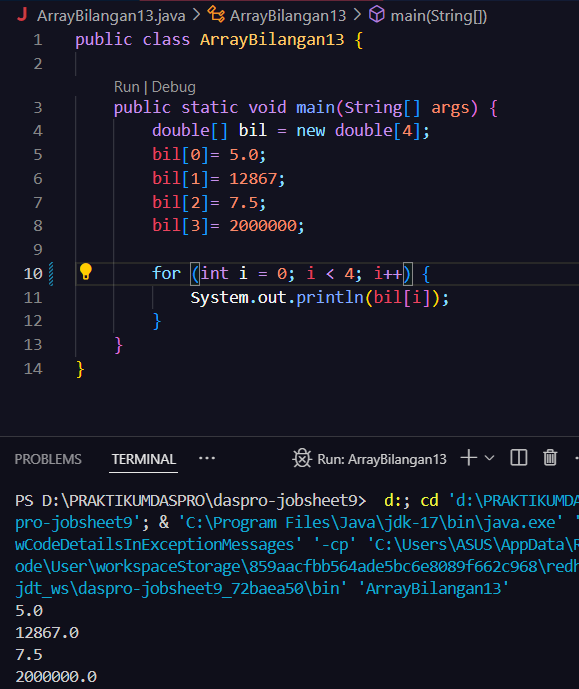
1. Push dan commit kode program ke github

Jawaban:

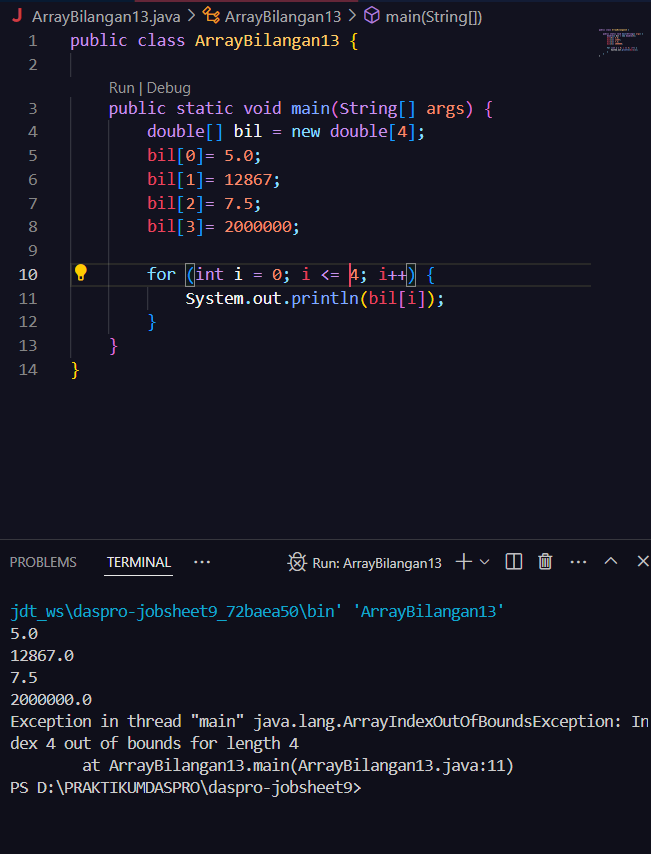
1. Jika elemen-elemen array bil diubah menjadi 5.0, 12867, 7.5, dan 2000000, maka akan terjadi error kompilasi dalam program Java tersebut. Tipe data array bil dideklarasikan sebagai int[] sehingga array ini hanya bisa menyimpan elemen-elemen bertipe bilangan bulat(integer) sehingga pada saat mencoba memasukkan nilai 5.0 dan 7.5 yang merupakan tipe data double, Java tidak dapat mengonversi secara otomatis nilai desimal ke tipe data integer tanpa melakukan konversi.
2. Modifikasi kode program di atas dengan melakukan inisialisasi elemen array sekaligus pada saat deklarasi array.



1. Kode program ini mendeklarasikan array bil bertipe double dengan 4 elemen, diisi dengan nilai-nilai yang berbeda, lalu mencetak nilai-nilai tersebut menggunakan loop for. Loop ini digunakan untuk mencetak seluruh elemen dari array bil satu per satu. Setiap elemen array diakses berdasarkan indeksnya, dimulai dari indeks 0 hingga 3, dan dicetak menggunakan System.out.println(). Loop berhenti karena kondisi i < 4 tidak lagi terpenuhi ketika i mencapai 4.



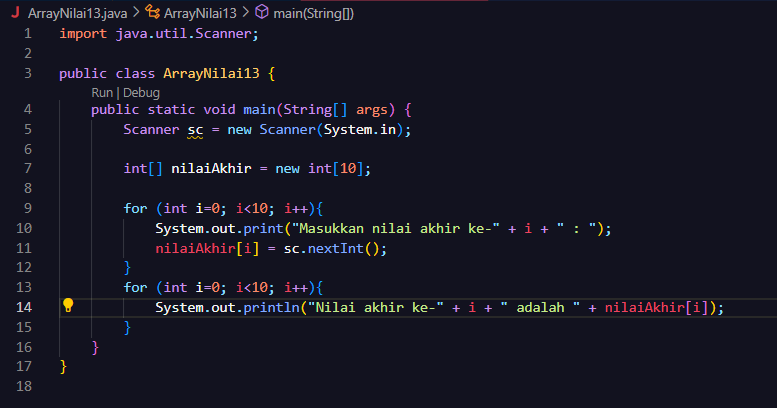
1. Program akan mengalami error pada saat dijalankan jika kondisi pada statement for-loop diubah menjadi i<=4. Error yang muncul adalah **ArrayIndexOutOfBoundsException**, yang berarti bahwa program mencoba mengakses indeks array di luar batas yang ada. Pada iterasi ketika i = 4, program akan mencoba mengakses elemen bil[4], tetapi array bil hanya memiliki elemen dengan indeks 0 hingga 3. Karena tidak ada elemen dengan indeks 4, Java akan mengeluarkan output **ArrayIndexOutOfBoundsException** dan menghentikan eksekusi program.



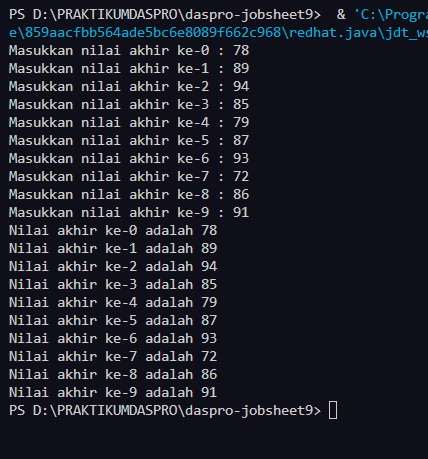
1. Push dan commit kode program ke github.

Percobaan 2

1. Buka text editor, buat file Java kemudian simpan dengan nama ArrayNilaiXX.java. (XX=nomor absen)
2. Buatlah struktur dasar java (membuat class dan method main).
3. Tambahkan import library Scanner.
4. Buat deklarasi array bertipe integer dengan nama nilaiAkhir dan berkapasitas 10 elemen seperti di bawah ini :
5. Buatlah struktur perulangan untuk menerima input dan mengisi elemen array nilaiAkhir, seperti berikut:
6. Menggunakan struktur perulangan, tampilkan semua isi elemen dari array nilaiAkhir, seperti berikut



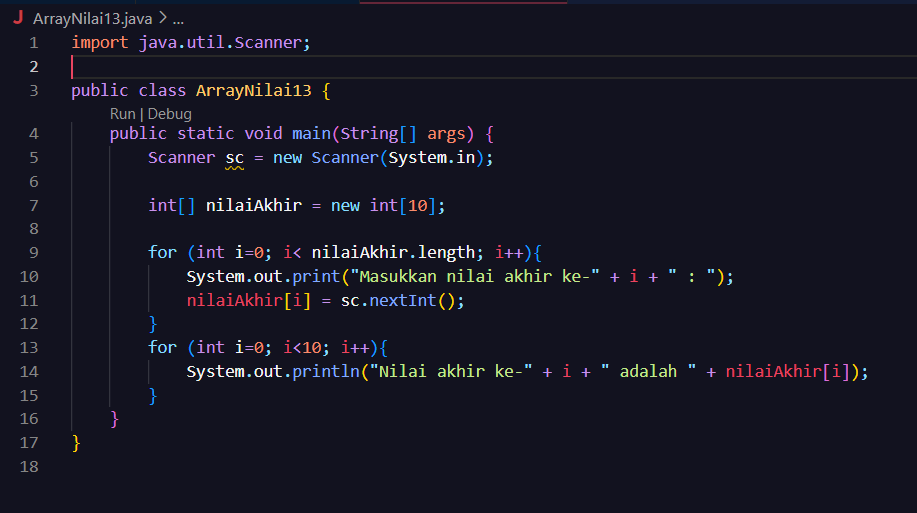
1. Jalankan program. Amati dan cocokkan dengan output berikut:

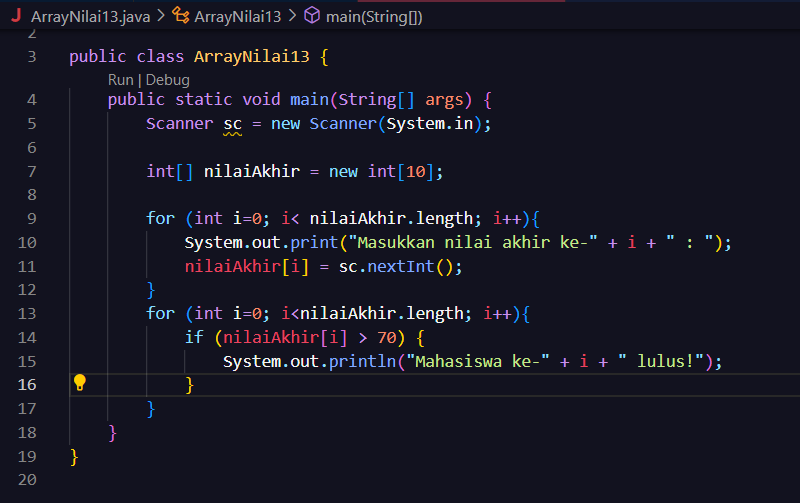


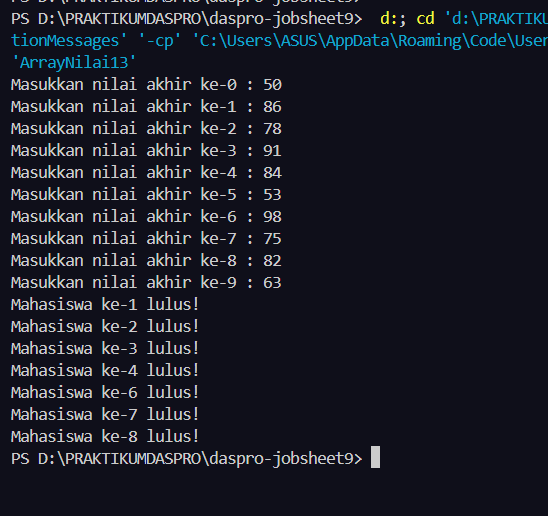
1. Push dan commit kode program ke github.

Jawaban:

1. Tidak terjadi perubahan namun dengan menggunakan nilaiAkhir.length membuat program lebih dinamis dan fleksibel karena tidak bergantung pada jumlah elemen yang ditetapkan secara eksplisit. Jika ukuran array berubah (misalnya menjadi new int[20]), loop akan otomatis menyesuaikan tanpa perlu mengubah batas manual pada loop sehingga maintenance lebih mudah.



1. Kondisi i<nilaiAkhir,length berarti batas loop bergantung pada panjang array nilaiAkhir yang memiliki 10 elemen karena array nilaiAkhir dideklarasikan sebagai int[] nilaiAkhir = new int[10];. Penggunaan kondisi i<nilaiAkhir,length akan membuat program lebih fleksibel karena tidak bergantung pada jumlah yang telah ditetapkan.
2. 



Alur program:

* Program mendeklarasikan array nilaiAkhir berukuran 10 untuk menyimpan nilai akhir dari 10 mahasiswa.
* Menggunakan loop for sehingga program akan meminta pengguna untuk memasukkan nilai akhir bagi setiap mahasiswa sebanyak 10 kali.
* Nilai tersebut dimasukkan dan disimpan dalam array nilaiAkhir pada indeks yang sesuai dari nilaiAkhir[0] sampai nilaiAkhir[9].
* Penggunaan loop for kedua untuk memeriksa setiap nilai dalam array nilaiAkhir. Jika nilai lebih dari 70 (nilaiAkhir[i] > 70), maka program menampilkan bahwa mahasiswa ke-i lulus. Mahasiswa yang memiliki nilai 70 atau kurang tidak akan ditampilkan apa pun (tidak lulus).

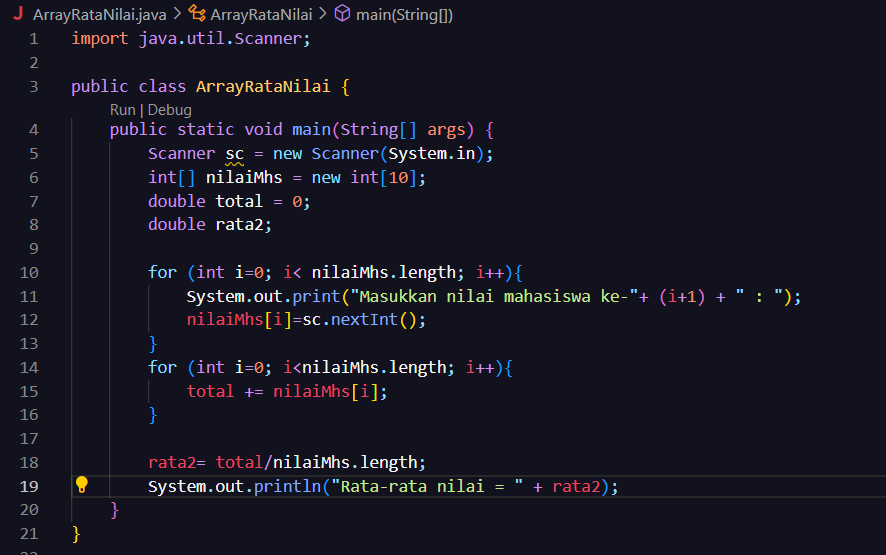
1. 

Pada modifikasi program ini, setiap mahasiswa akan dicek apakah nilainya lebih dari 70. Jika nilai lebih dari 70, maka ditampilkan bahwa mahasiswa tersebut lulus. Jika nilai 70 atau kurang, maka ditampilkan bahwa mahasiswa tersebut tidak lulus.

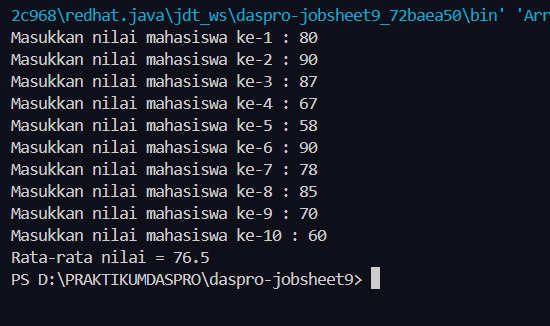
1. Push dan commit kode program ke github.

Percobaan 3

1. Buka text editor, buat file Java, kemudian simpan dengan nama ArrayRataNilaiXX.java. (XX = nomor absen).
2. Buatlah struktur dasar java (membuat class dan method main).
3. Import dan deklarasikan Scanner untuk keperluan input.
4. Buat array nilaiMhs bertipe integer dengan kapasitas 10. Kemudian deklarasikan variable total dan rata2 seperti gambar berikut ini:
5. Isi array nilaiMhs dengan nilai dari input pengguna, sebagai berikut:
6. Gunakan perulangan untuk menghitung jumlah keseluruhan nilai dalam array nilaiMhs, sebagai berikut:
7. Kemudian hitung nilai rata-rata dengan cara nilai total dibagi jumlah elemen dari array nilaiMhs:



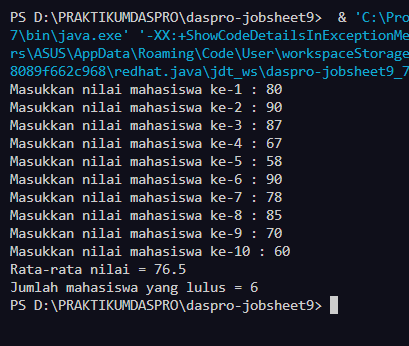
1. Amati hasilnya sebagai berikut:



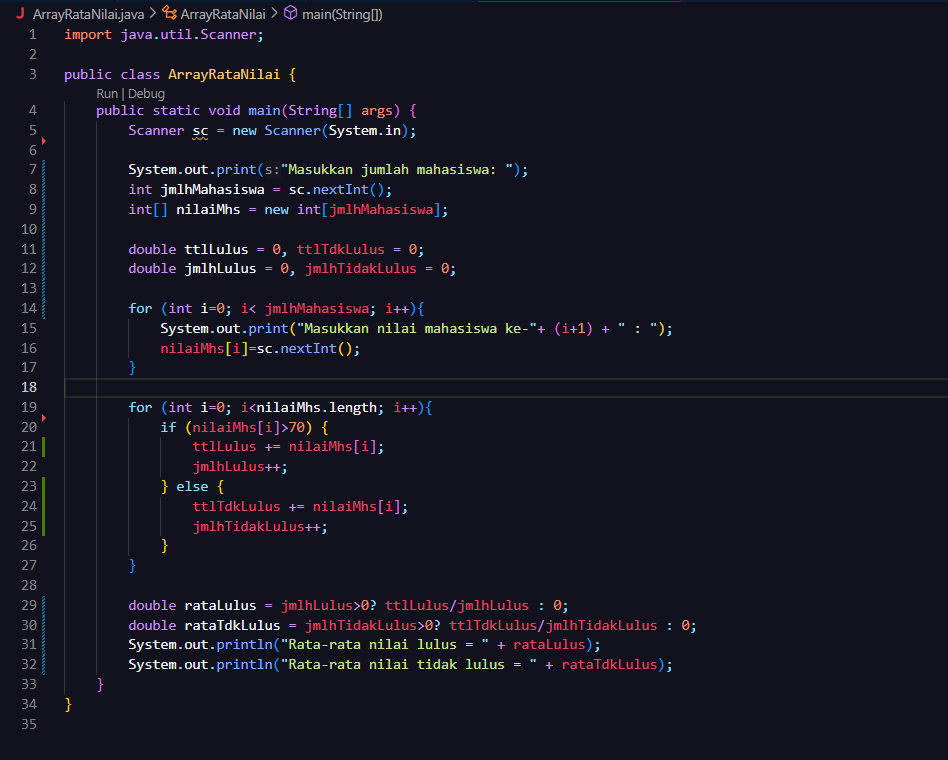
1. Push dan commit kode program ke github

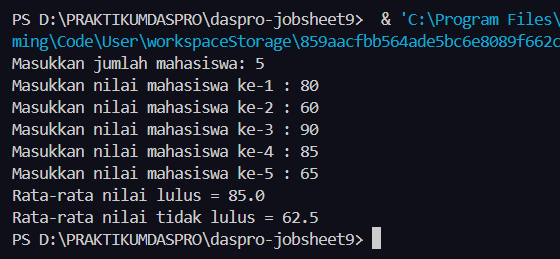
Jawaban:

1. 



Program ini dimodifikasi dengan menambahkan variabel counter yang akan menghitung jumlah mahasiswa yang nilainya lebih dari 70. Lalu pada loop kedua, ketika menjumlahkan total nilai, program sekaligus mengecek apakah nilai tersebut lebih besar dari 70 dan menambah nilai counter jika kondisi terpenuhi.

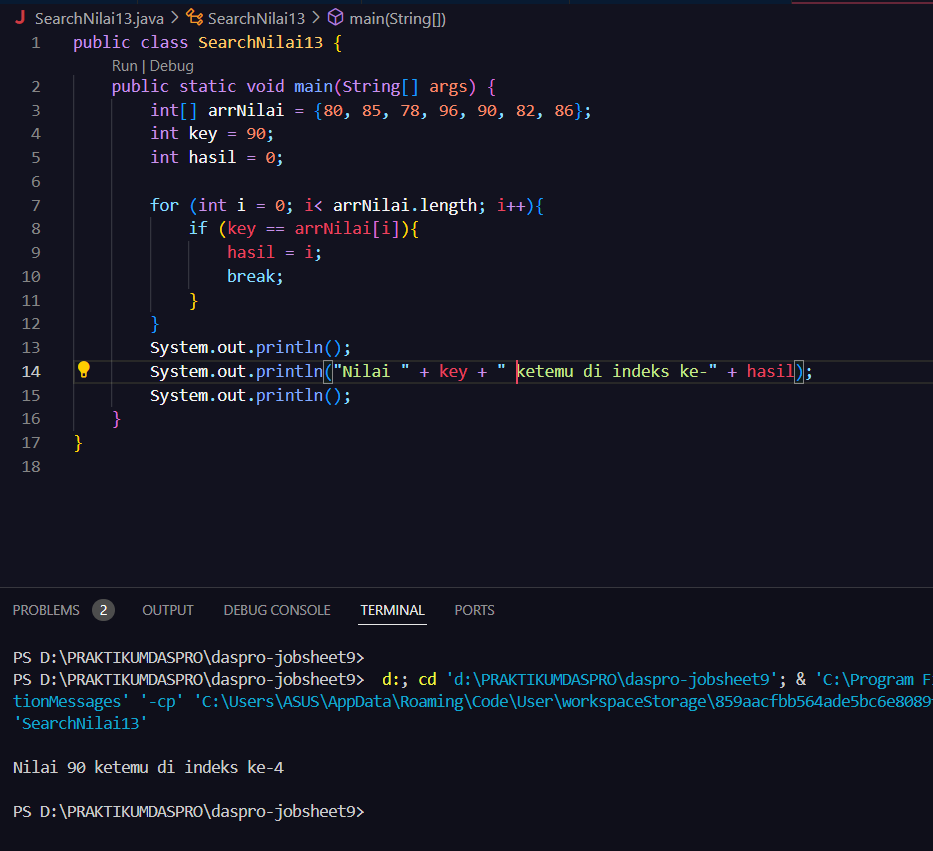
1. 

Program ini dimodifikasi agar meminta pengguna memasukkan jumlah mahasiswa yang digunakan untuk menentukan ukuran array nilaiMhs. Array nilaiMhs sekarang memiliki ukuran yang sesuai dengan jumlah mahasiswa yang dimasukkan oleh pengguna (int[] nilaiMhs = new int[jumlahMahasiswa];) sehingga program bisa menangani jumlah mahasiswa yang dinamis sesuai dengan input pengguna. Program juga akan menghitung rata-rata nilai dan jumlah mahasiswa yang lulus berdasarkan jumlah mahasiswa yang ditentukan pengguna pada awal program.

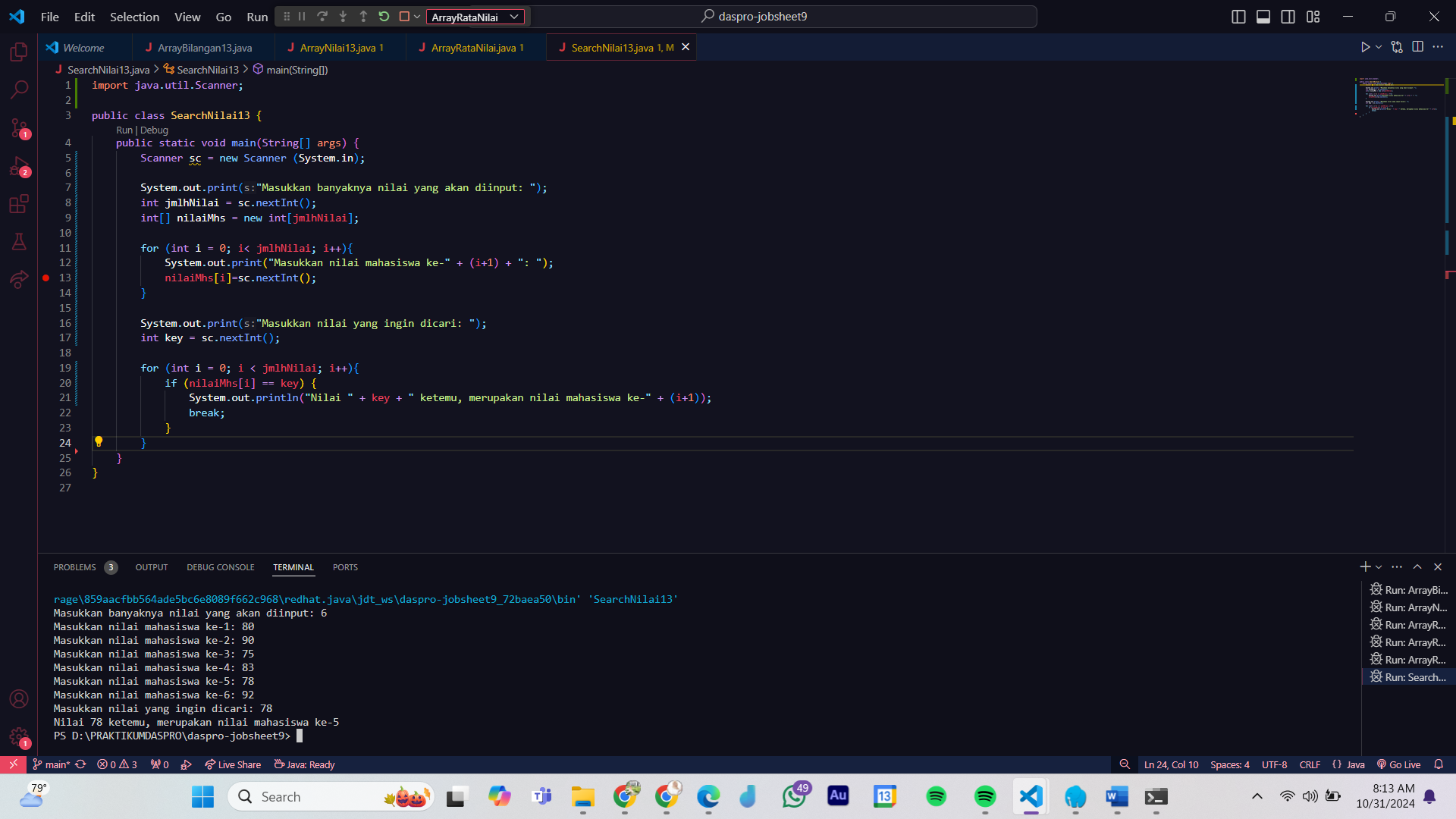
1. Push dan commit kode program ke github

Percobaan 4

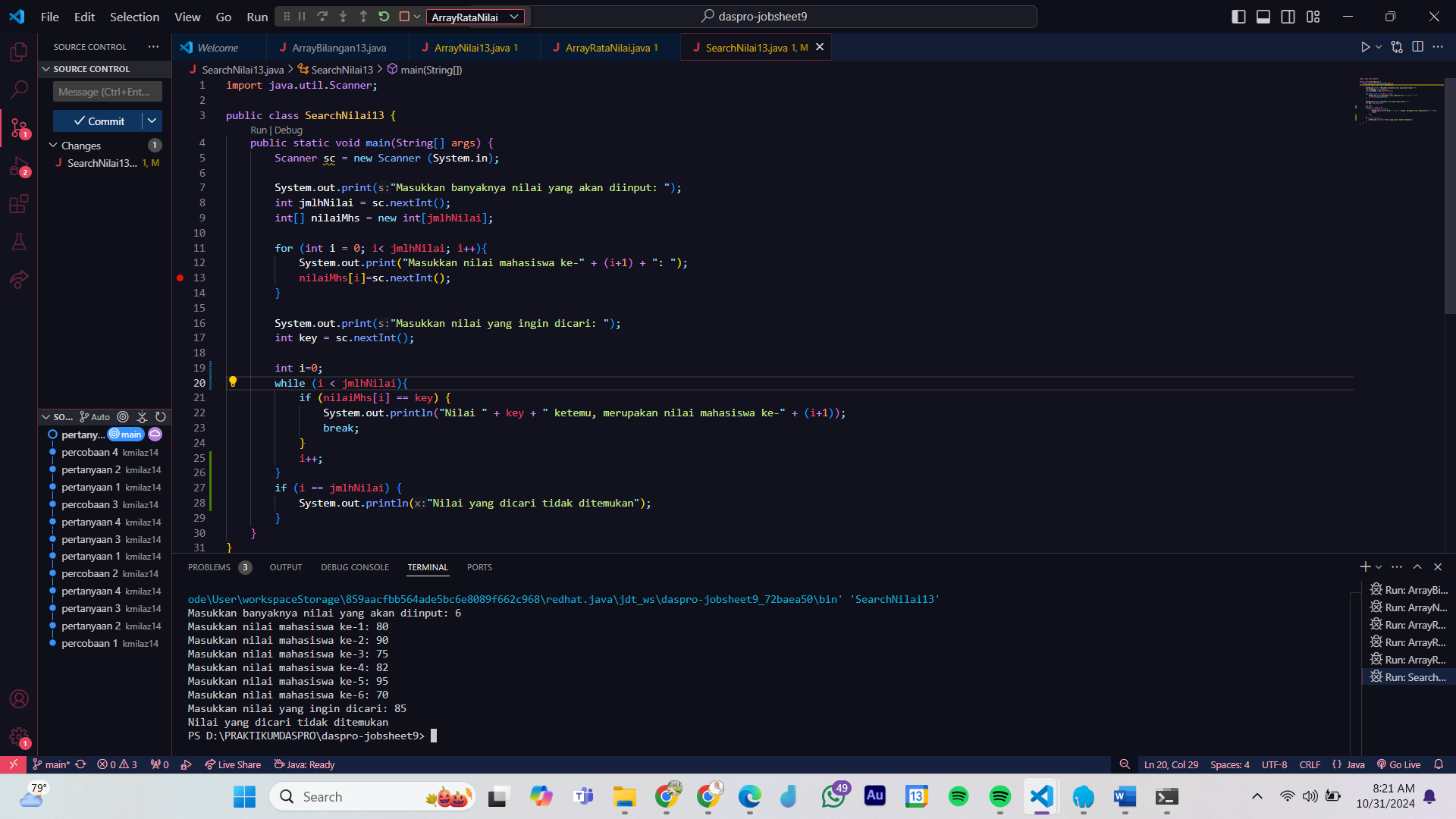
1. Buka text editor, buat file Java, kemudian simpan dengan nama SearchNilaiXX.java. (XX = nomor absen).
2. Tambahkan kode berikut ini:
3. Jalankan program tersebut. Amati apa yang terjadi. Cocokkan dengan output berikut:
4. Push dan commit kode program ke github



Jawaban:

1. Fungsi break digunakan untuk menghentikan atau keluar dari loop sebelum loop selesai berjalan. Pada statement break; di baris ke-10 itu ketika nilai yang telah diset dalam key itu ditemukan pada array, maka break; akan menghentikan looping. Program akan menampilkan pesan dan keluar dari loop, sehingga tidak perlu memeriksa elemen-elemen berikutnya.
2. 

Program ini dimodifikasi dengan mencari nilai (key) dalam array nilaiMhs. Jika nilai ditemukan, program akan menampilkan posisi nilai tersebut (indeks + 1) dan langsung berhenti setelah menemukannya. Jika nilai tidak ditemukan, program tidak menampilkan pesan apa pun.

1. 

Program pencarian nilai dalam array dengan menggunakan metode linear search. Program ini menerima input sejumlah nilai mahasiswa, kemudian mencari nilai tertentu (yang disebut key) yang ingin dicari oleh pengguna. Jika nilai tersebut ditemukan, program akan menampilkan posisi nilai dalam array; jika tidak ditemukan, program akan menampilkan bahwa nilai tersebut tidak ada.

1. Push dan commit kode program ke github

Tugas

1. Anda diminta untuk membuat program yang dapat menyimpan dan mengelola nilai mahasiswa. Nilai berupa bilangan bulat. Program harus menyediakan fitur untuk:

- memasukkan banyaknya nilai mahasiswa yang akan diinput,

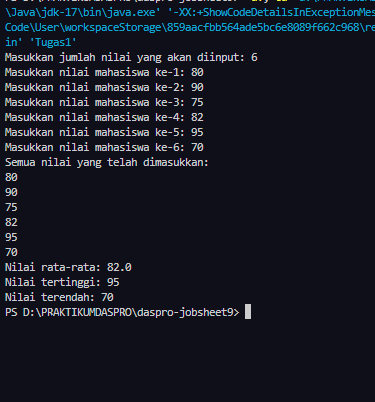
- memasukkan setiap nilai mahasiswa,

- menghitung nilai rata-rata,

- menampilkan nilai tertinggi dan nilai terendah, serta

- menampilkan semua nilai yang telah dimasukkan.





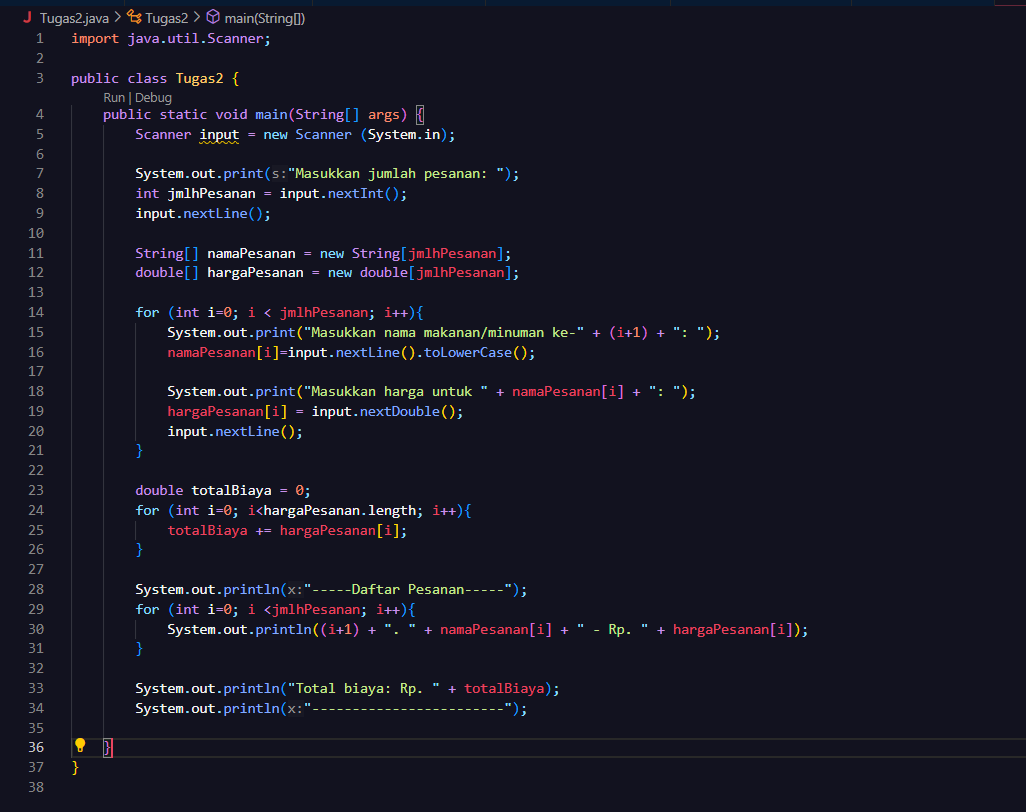
Program ini dimulai dengan meminta pengguna memasukkan jumlah nilai yang akan diinput, lalu membuat array dengan ukuran sesuai jumlah tersebut. Program kemudian menginisialisasi variabel total, nilaiTertinggi, dan nilaiTerendah untuk menghitung total dan menemukan nilai tertinggi serta terendah. Selanjutnya, program mengulangi proses input nilai untuk setiap mahasiswa, menambahkan nilai ke total, serta memperbarui nilaiTertinggi dan nilaiTerendah jika nilai yang baru lebih tinggi atau lebih rendah dari nilai sebelumnya. Setelah semua nilai dimasukkan, program menghitung rata-rata dengan membagi total dengan jumlah nilai. Lalu program akan menampilkan semua nilai yang telah diinput, rata-rata, nilai tertinggi, dan nilai terendah.

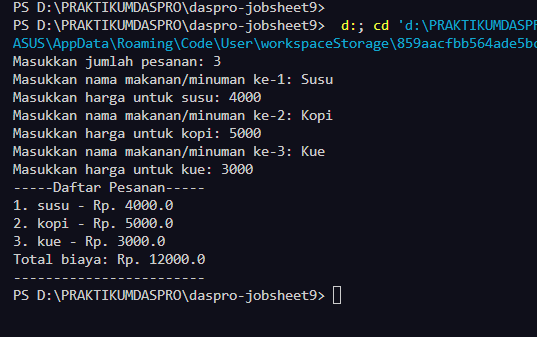
1. Buat program yang dapat mengelola pemesanan makanan dan minuman di sebuah kafe. Program akan memungkinkan pengguna untuk memasukkan pesanan, menghitung total biaya pesanan, dan menampilkan daftar pesanan yang telah dibuat.

- Input: o jumlah pesanan (input dari pengguna). o nama makanan/minuman dan harga untuk masing-masing pesanan (input dari pengguna)

- Proses: o simpan data pesanan dalam array satu dimensi untuk nama pesanan; dan array satu dimensi terpisah untuk harga. o hitung total biaya dari semua pesanan yang dimasukkan. o tampilkan daftar pesanan yang telah dimasukkan bersama dengan total biaya.

- Output: o daftar pesanan dan total biaya dari semua pesanan.





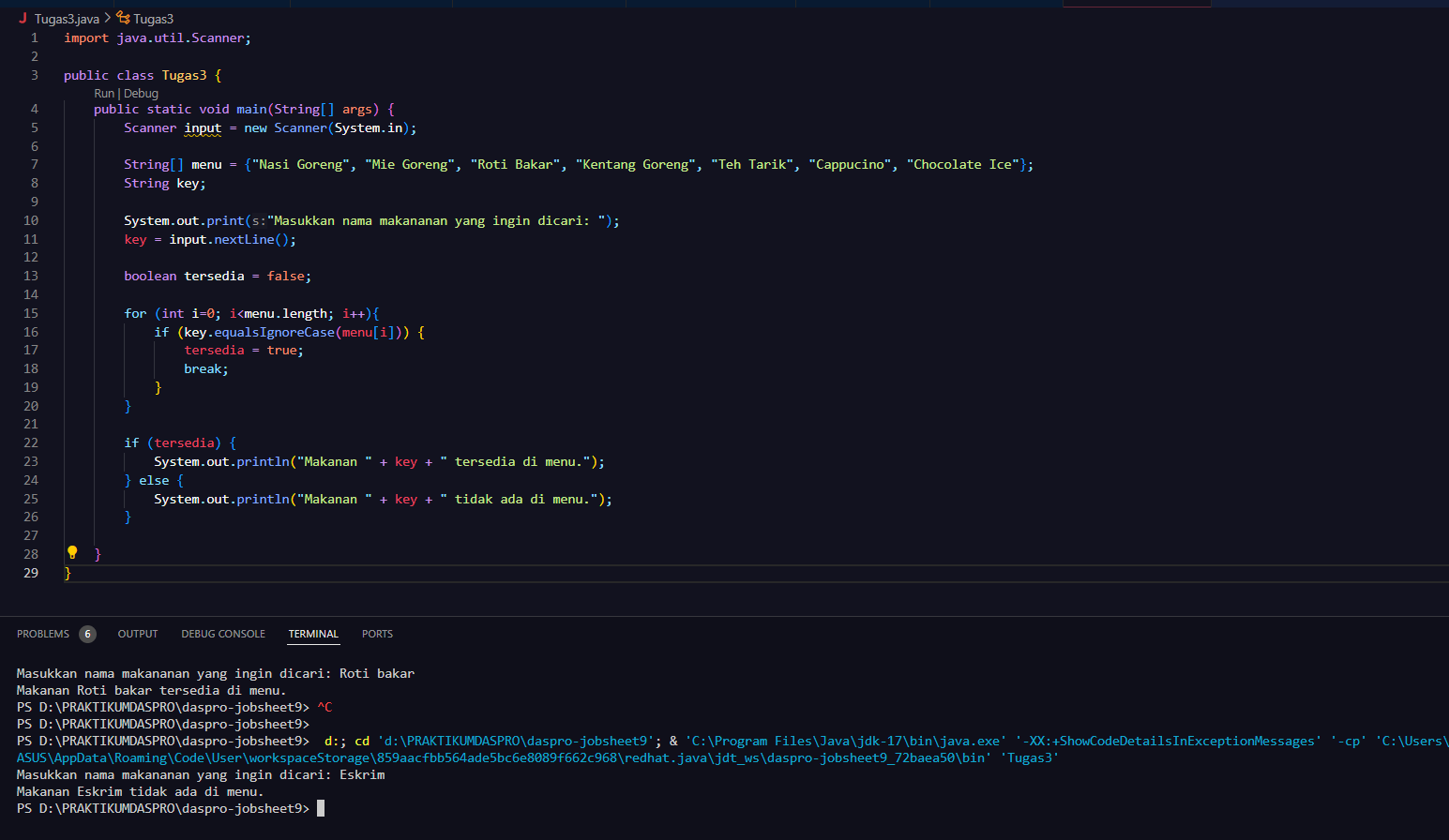
Program ini dimulai dengan meminta pengguna untuk memasukkan jumlah pesanan, lalu membuat array namaPesanan dan hargaPesanan sesuai jumlah tersebut. Pengguna kemudian diminta untuk memasukkan nama dan harga setiap pesanan, yang masing-masing disimpan dalam array. Program kemudian menghitung total biaya semua pesanan dengan menjumlahkan elemen di hargaPesanan. Setelah itu, program menampilkan daftar semua pesanan beserta harganya dan total biaya dari seluruh pesanan.

1. Masih menggunakan kasus pada pemesanan makanan di kafe, buatlah program yang memungkinkan pengguna untuk memesan makanan dari menu yang tersedia di kafe. Program harus menyimpan daftar nama makanan dalam sebuah array dan memberikan opsi untuk mencari makanan yang diinginkan menggunakan metode linear search.

- Input: o daftar menu makanan yang telah ditentukan sebelumnya dalam bentuk array. Nama-nama makanan telah di-inisialisasi saat deklarasi array. Misal: String[] menu = {"Nasi Goreng", "Mie Goreng", "Roti Bakar", "Kentang Goreng", "Teh Tarik", "Cappucino", "Chocolate Ice"}; o nama makanan yang ingin dicari (input dari pengguna).

-Proses: o program mencari nama makanan yang dimasukkan pengguna menggunakan algoritma linear search. o jika makanan ditemukan, program akan menginformasikan pengguna bahwa makanan tersebut tersedia. Jika tidak ditemukan, program akan memberi tahu pengguna bahwa makanan yang dicari tidak ada di menu.

- Output: o Tampilkan hasil pencarian kepada pengguna.



Program ini meminta pengguna memasukkan nama makanan yang ingin dicari, lalu menyimpan input tersebut dalam variabel key. Program menggunakan loop untuk membandingkan key dengan setiap item pada array menu. Jika ada kecocokan, variabel tersedia diubah menjadi true, dan loop dihentikan. Setelah loop selesai, program menampilkan pesan apakah makanan tersedia di menu atau tidak.