**Laporan Hasil Praktikum**

**Dasar Pemrograman**

**Jobsheet 9**



RIFQI HILMI MUBAROK

244107060110

SIB 1-D

**Program Studi Sistem Informasi Bisnis**

**Jurusan Teknologi Informasi**

**POLINEMA**

**2024**

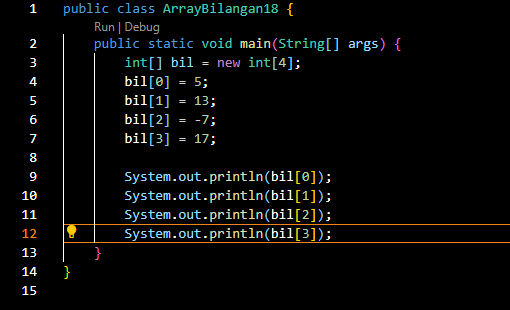
**Jobsheet 9**

**Array 1**

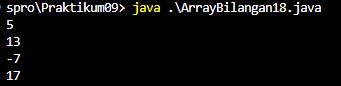
1. **Tujuan**

* Mahasiswa mampu memahami pembuatan Array 1 dimensi dan pengaksesan elemennya di Java
* Mahasiswa mampu membuat program dengan penggunaan Array satu dimensi

1. **Praktikum**
   1. **Percobaan 1: Mengisi Elemen Array**
      1. *Source Code:*

****

* + 1. Hasil

****

* + 1. *Push* dan *commit* kode program ke github

****

* + 1. Pertanyaan
       1. Jika isi masing-masing elemen array bil diubah dengan angka 5.0, 12867, 7.5, 2000000. Apa yang terjadi? Mengapa bisa demikian?
       2. Modifikasi kode program di atas dengan melakukan inisialisasi elemen array sekaligus pada saat deklarasi array.
       3. Ubah statement pada langkah No 4 menjadi seperti berikut

**a**

for (int i = 0; i < 4; i++) {

System.out.println(bil[i]);

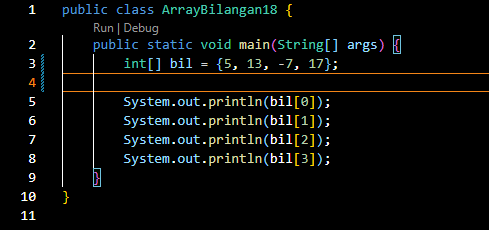
}

**a**

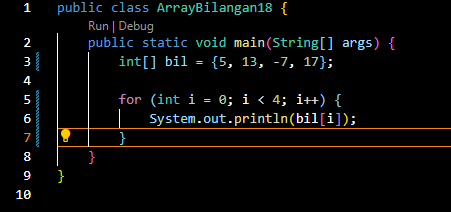
**a**

Apa keluaran dari program? Jelaskan maksud dari statement tersebut.

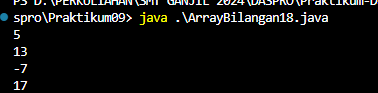
* + - 1. Jika kondisi pada statement for-loop di atas diubah menjadi: i <= 4, apa keluaran dari program? Mengapa demikian?
      2. *Push* dan *commit* kode program ke github
    1. Jawaban
       1. Jika mengubah nilai setiap elemen array bil menjadi 5.0, 12867, 7.5, dan 2000000, akan terjadi error saat kompilasi. Hal ini karena bil dideklarasikan sebagai array bertipe int, sedangkan 5.0 dan 7.5 adalah angka desimal (tipe double) yang tidak dapat langsung dikonversi ke tipe int dalam Java. Hanya angka bulat (seperti 12867 dan 2000000) yang dapat digunakan untuk array int tanpa perlu konversi tipe.
       2. Modifikasi:



* + - 1. Modifikasi:



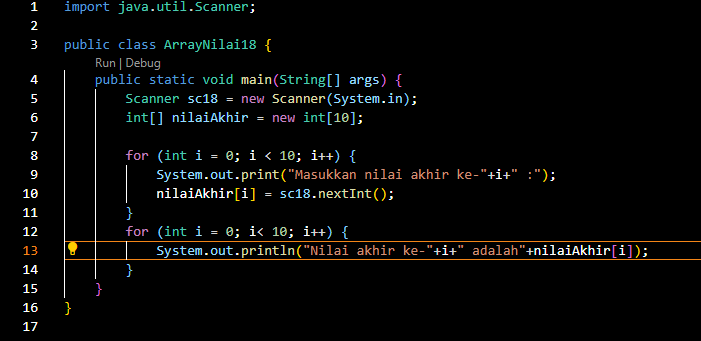
Keluaran dari program:



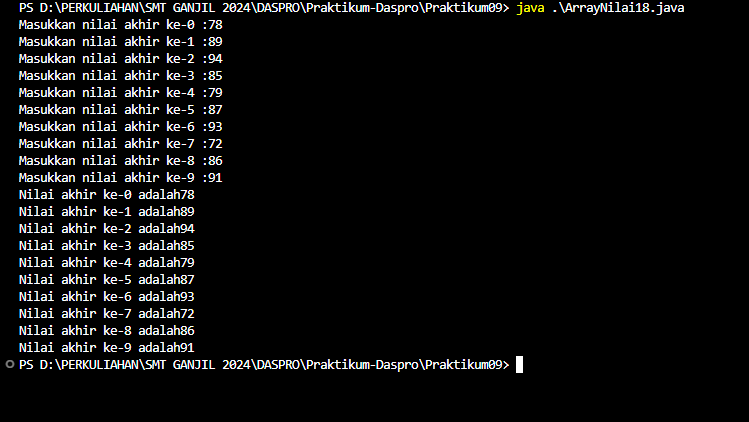
* Perulangan for dimulai dari i = 0 dan akan berjalan selama i < 4. Artinya, perulangan akan berhenti ketika i mencapai nilai 4.
* Setiap iterasi, System.out.println(bil[i]) akan mencetak nilai elemen array bil pada indeks i.
* Karena array bil memiliki empat elemen (dengan indeks 0 hingga 3), perulangan ini akan mencetak nilai bil[0], bil[1], bil[2], dan bil[3] secara berurutan.

Tidak akan terjadi error pada kode ini karena batas perulangan (i < 4) sesuai dengan jumlah elemen dalam array bil, sehingga tidak ada akses ke indeks yang di luar jangkauan array.

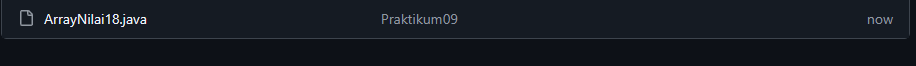
* + - 1. Jika kondisi diubah menjadi seperti pertanyaan tersebut, maka akan terjadi error ArrayIndexOutOfBoundsException. Hal ini karena array bil hanya memiliki 4 elemen dengan indeks dari 0 hingga 3. Saat perulangan mencapai i = 4, program akan mencoba mengakses bil[4], yang tidak ada dalam array tersebut. Oleh karena itu, program akan berhenti dan menampilkan error.
      2. 
  1. **Percobaan 2: Studi Kasus Nilai Mahasiswa di SIAKAD – Meminta Inputan Pengguna untuk Mengisi Elemen Array**
     1. *Source code:*



*Output*:



*Push* dan *commit* kode program ke github



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

i

for (int i = 0; i < nilaiAkhir.length; i++) {

System.out.println(“Masukkan nilai akhir ke-“+i+” : “);

nilaiAkhir[i] = sc18.nextInt();

}

s

Jalankan program. Apakah terjadi perubahan? Mengapa demikian?

* + - 1. Apa yang dimaksud dengan kondisi: i < nilaiAkhir.length?
      2. 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!");

}

}

S

A

Jalankan program dan jelaskan alur program!

* + - 1. 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:

a

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!

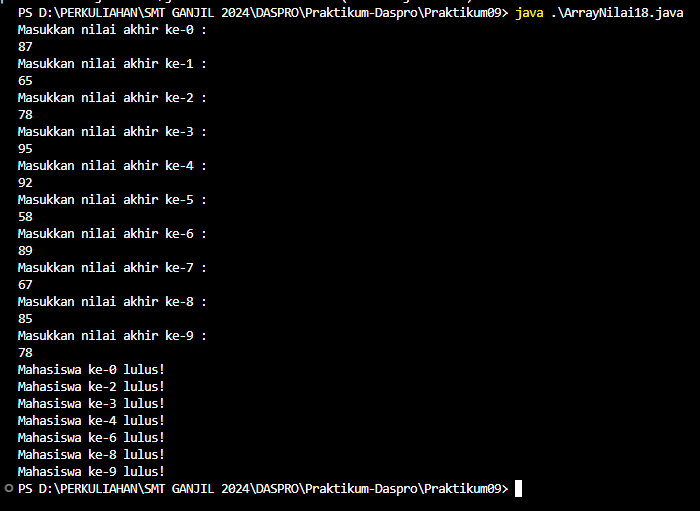
Mahasiswa ke-6 lulus!

Mahasiswa ke-7 tidak lulus!

Mahasiswa ke-8 lulus!

Mahasiswa ke-9 lulus!

* + - 1. *Push* dan *commit* kode program ke github.
    1. Jawaban
       1. Tidak terjadi perubahan setelah mengubah statement pada langkah nomor 5. Dengan menggunakan i < nilaiAkhir.length, program akan tetap berfungsi dengan baik, meskipun panjang array nilaiAkhir berubah. Jika array diubah menjadi ukuran lain, misalnya new int[20], perulangan otomatis menyesuaikan dengan panjang baru. Namun terjadi perubahan pada kolom baru, dari yang sebelumnya hanya print menjadi println.
       2. Kondisi i < nilaiAkhir.length dalam perulangan for berarti perulangan akan berjalan selama nilai i kurang dari panjang array nilaiAkhir. Karena array nilaiAkhir berukuran 10, maka nilaiAkhir.length adalah 10, dan perulangan akan berjalan dari i = 0 hingga i = 9. Kondisi ini memastikan bahwa setiap elemen dalam array diakses tanpa melebihi batas.
       3. Hasil program:



Alur program:

* Program akan meminta pengguna untuk memasukkan nilai akhir untuk 10 mahasiswa.
* Setelah itu, program akan memeriksa setiap nilai dalam array nilaiAkhir.
* Jika nilai lebih besar dari 70, program akan mencetak bahwa mahasiswa tersebut lulus.
* Jika nilai tidak lebih dari 70, tidak ada output untuk mahasiswa tersebut.
  + - 1. Agar program dapat menampilkan status kelulusan semua mahasiswa (baik yang lulus maupun tidak lulus), Anda bisa memodifikasi for loop kedua menjadi seperti ini:

s

for (int i = 0; i < nilaiAkhir.length; i++) {

if (nilaiAkhir[i] > 70) {

System.out.println("Mahasiswa ke-" + i + " lulus!");

} else {

System.out.println("Mahasiswa ke-" + i + " tidak lulus!");

}

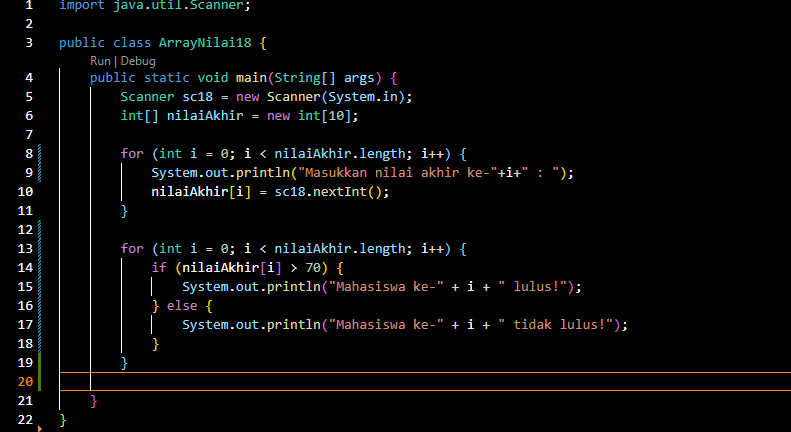
}

S

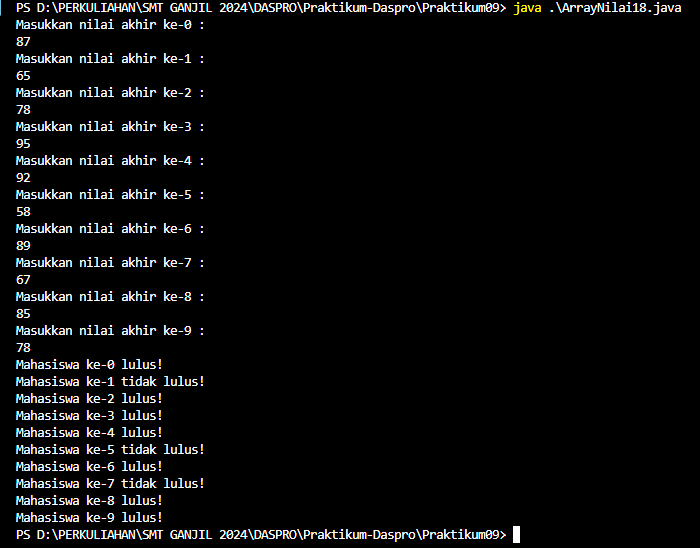
S

S

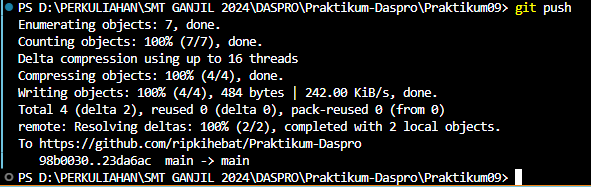
Modifikasi:



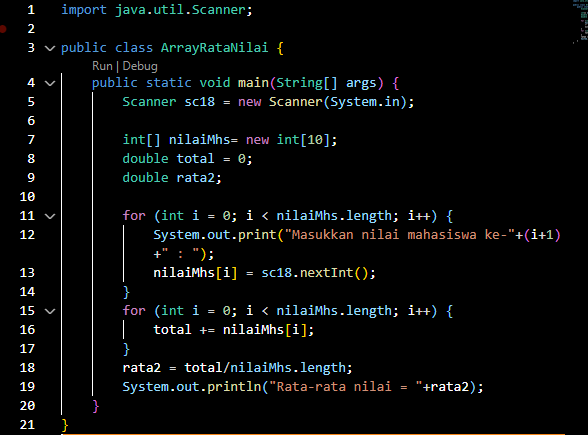
Hasil keluaran program:



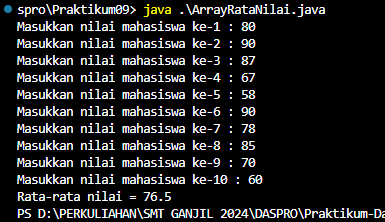
* + - 1. *Push* dan *commit* ke github



* 1. **Percobaan 3: Studi Kasus Nilai Mahasiswa di SIAKAD – Melakukan Operasi Aritmatika terhadap Elemen Array**
     1. *Source code:*



* + 1. Hasil keluaran program:



* + 1. Pertanyaan
       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:

d

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

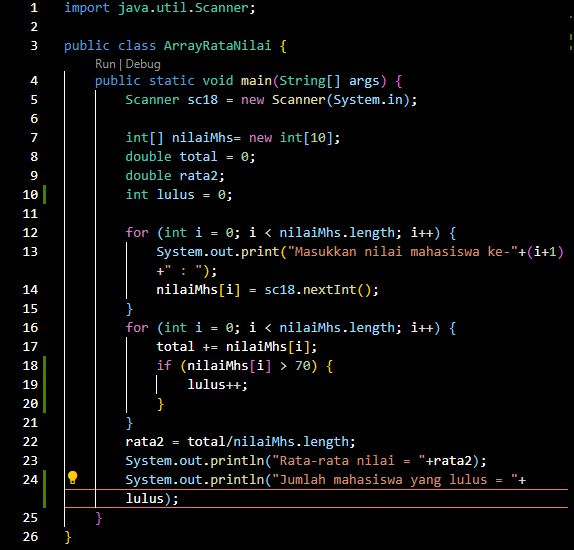
Masukkan nilai mahasiswa ke-5 : 65

Rata-rata nilai lulus = 85.0

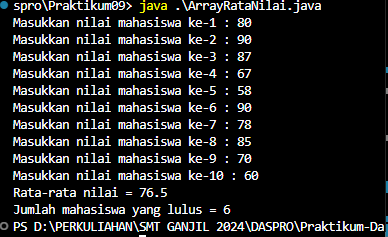
Rata-rata nilai tidak lulus = 62.5

S

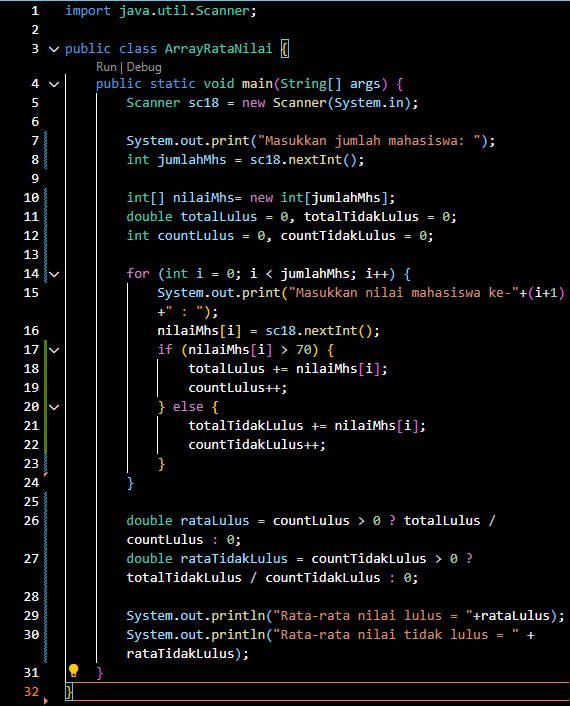
* + - 1. Modifikasi kode program untuk menampilkan nilai tertinggi dan terendah
      2. Push dan commit kode program ke github.
    1. Jawaban
       1. Modifikasi pada kode program



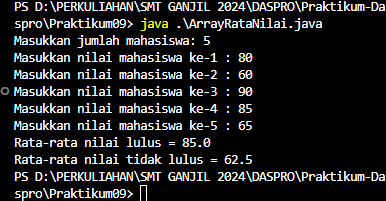
Keluaran dari modifikasi pada program



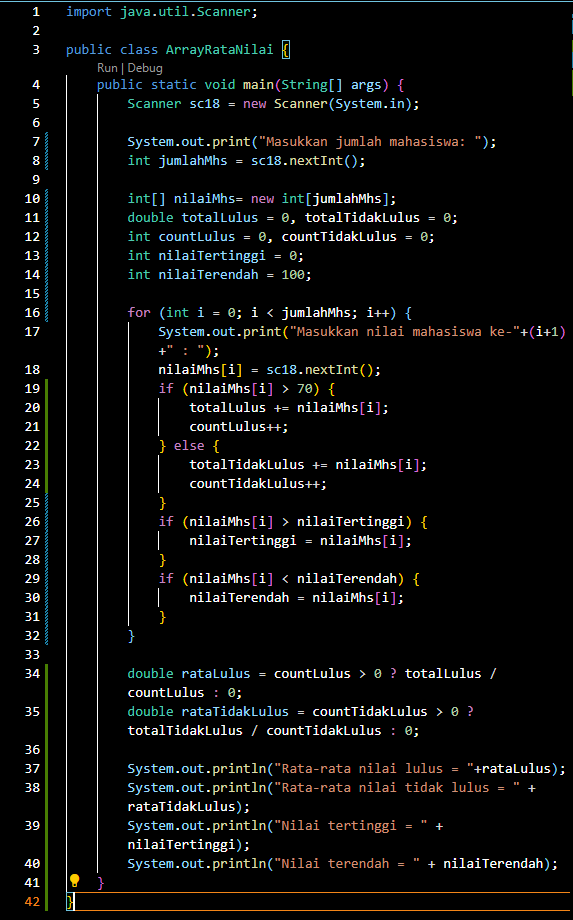
* + - 1. Modifikasi pada kode program



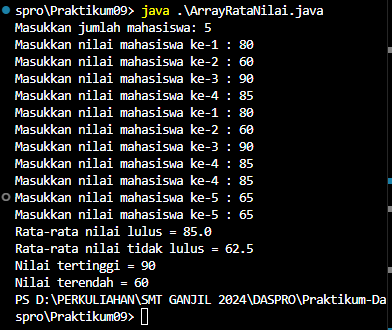
Keluaran dari modifikasi pada kode program



* + - 1. Modifikasi pada kode program



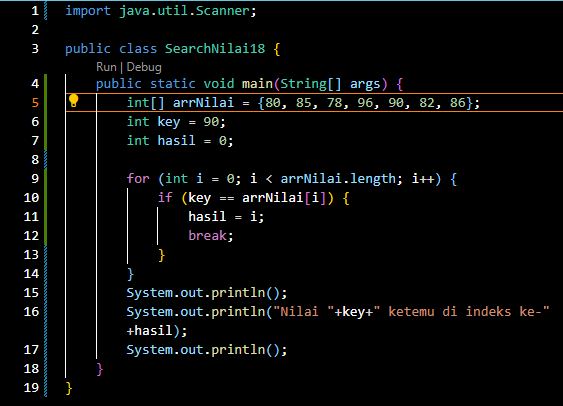
Keluaran dari modifikasi pada kode program



* + - 1. *Push* dan *commit* ke github



* 1. **Percobaan 4: Linear Searching**
     1. *Source code*



Keluaran kode program



*Push* dan *commit* ke github



* + 1. Pertanyaan
       1. Jelaskan maksud dari statement **break;** pada baris ke-10 pada kode program percobaan 4 di atas.
       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:

s

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

Nilai 78 ketemu, merupakan nilai mahasiswa ke-5

S

* + - 1. 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 : 83

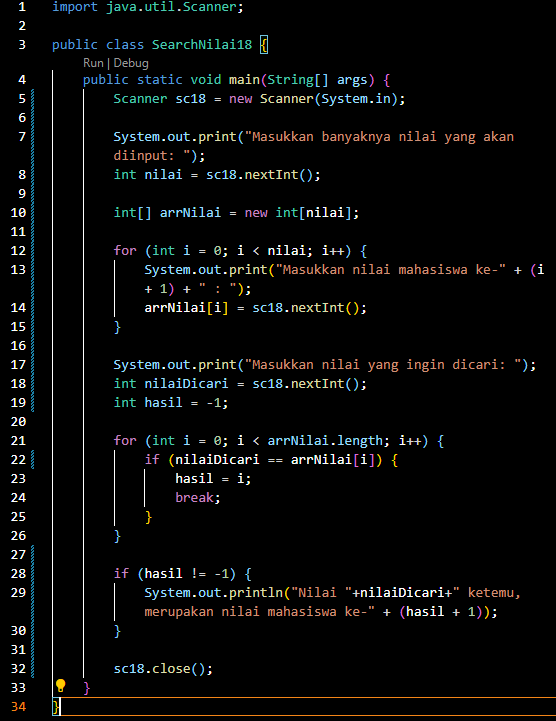
Masukkan nilai mahasiswa ke-5 : 78

Masukkan nilai mahasiswa ke-6 : 92

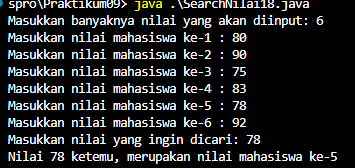
Masukkan nilai yang ingin dicari: 85

Nilai yang dicari tidak ditemukan

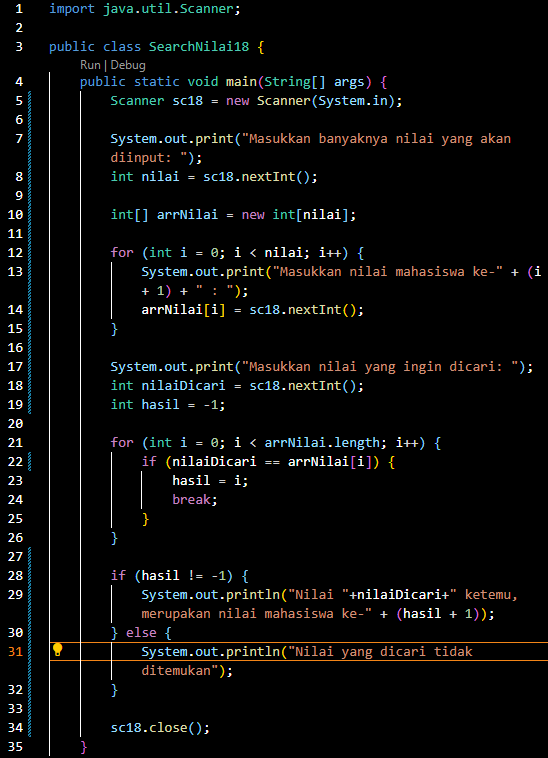
* + - 1. *Push* dan *commit* kode program ke github
    1. Jawaban
       1. Pada kode yang diberikan, perintah **break;** berfungsi untuk menghentikan eksekusi loop for segera setelah nilai yang dicari ditemukan (nilai yang sesuai dengan key). Ketika kondisi if (key == arrNilai[i]) terpenuhi, hasil = i; akan menyimpan indeks di mana nilai ditemukan, dan kemudian perintah **break;** akan menghentikan pencarian lebih lanjut di dalam array, karena kita sudah menemukan elemen yang dicari. Ini mengoptimalkan program dengan menghentikan pencarian lebih lanjut setelah nilai ditemukan.
       2. Modifikasi pada kode program



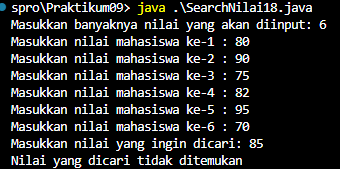
Keluaran dari modifikasi pada kode program



* + - 1. Modifikasi pada kode program



Keluaran dari modifikasi pada kode program

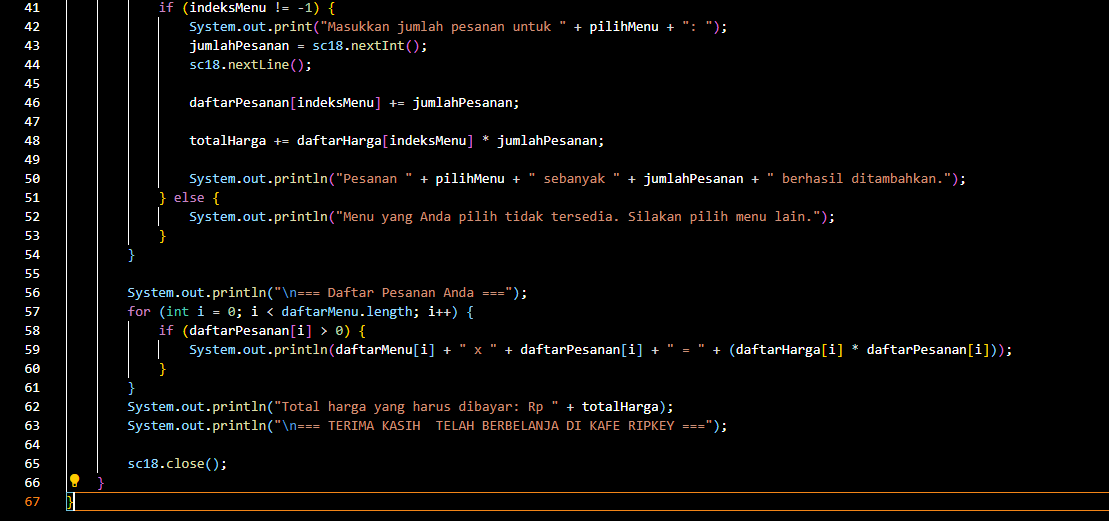


* + - 1. *Push* dan *commit* ke github



1. **Tugas: Membuat Program yang dapat Mengelola Pemesanan Makanan dan Minuman di Sebuah Kafe**
   1. *Source code*





* 1. *Output* Kode Program

