**LAPORAN PRAKTIKUM**

**Politeknik Negeri Malang**

**Praktikan**

**ALGORITMA DAN STRUKTUR DATA**

****

2022

2141720183

RIDWAN CAESAR RIZQI KARISMA BIWARNI

TI 1C

Daftar Isi [harus diupdate setiap ada perubahan]

[Searching / Pencarian Menggunakan Agoritma Sequential Search 4](#_Toc99613487)

[Langkah 1 4](#_Toc99613488)

[Langkah 2 4](#_Toc99613489)

[Langkah 3 4](#_Toc99613490)

[Langkah 4 5](#_Toc99613491)

[Langkah 5 5](#_Toc99613492)

[Langkah 6 5](#_Toc99613493)

[Langkah 7 5](#_Toc99613494)

[Langkah 8 6](#_Toc99613495)

[Langkah 9 6](#_Toc99613496)

[Langkah 10 6](#_Toc99613497)

[Langkah 11 6](#_Toc99613498)

[Langkah 12 7](#_Toc99613499)

[Langkah 13 7](#_Toc99613500)

[Langkah 14 7](#_Toc99613501)

[Langkah 15 8](#_Toc99613502)

[Langkah 16 8](#_Toc99613503)

[Langkah 17 8](#_Toc99613504)

[Langkah 18 8](#_Toc99613505)

[Verifikasi Hasil Percobaan 8](#_Toc99613506)

[Pertanyaan 9](#_Toc99613507)

[Jawaban 9](#_Toc99613508)

[Searching / Pencarian Menggunakan Binary Search 11](#_Toc99613509)

[Langkah 1 11](#_Toc99613510)

[Langkah 2 11](#_Toc99613511)

[Langkah 3 11](#_Toc99613512)

[Verifikasi Hasil Percobaan 12](#_Toc99613513)

[Pertanyaan 12](#_Toc99613514)

[Jawaban 13](#_Toc99613515)

[Percobaan Pengayaan Divide and Conquer 15](#_Toc99613516)

[Langkah 1 15](#_Toc99613517)

[Langkah 2 15](#_Toc99613518)

[Langkah 3 15](#_Toc99613519)

[Langkah 4 15](#_Toc99613520)

[Langkah 5 15](#_Toc99613521)

[Langkah 6 16](#_Toc99613522)

[Langkah 7 16](#_Toc99613523)

[Langkah 8 16](#_Toc99613524)

[Langkah 9 17](#_Toc99613525)

[Langkah 10 17](#_Toc99613526)

[Verifikasi Hasil Percobaan 17](#_Toc99613527)

[Latihan Praktikum 18](#_Toc99613528)

[Soal nomor 1 18](#_Toc99613529)

[Jawaban nomor 1 18](#_Toc99613530)

[Modifikasi source code class PencarianMhs 18](#_Toc99613531)

[Modifikasi source code class MahasiswaMain 18](#_Toc99613532)

[Output 19](#_Toc99613533)

[Soal nomor 2 19](#_Toc99613534)

[Jawaban nomor 2 20](#_Toc99613535)

[Modifikasi source code class PencarianMhs 20](#_Toc99613536)

[Modifikasi source code class MahasiswaMain 20](#_Toc99613537)

[Output 21](#_Toc99613538)

[Soal nomor 3 21](#_Toc99613539)

[Jawaban nomor 3 21](#_Toc99613540)

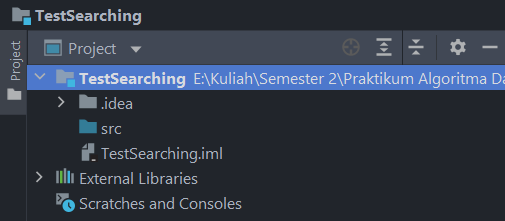
[Source code 21](#_Toc99613541)

[Output 23](#_Toc99613542)

# Searching / Pencarian Menggunakan Agoritma Sequential Search

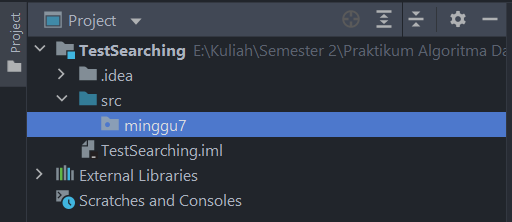
## Langkah 1

Buatlah Project baru pada Netbeans dengan nama TestSearching



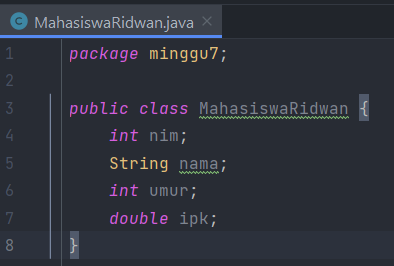
## Langkah 2

Kemuadian buat packages baru dengan nama minggu7.



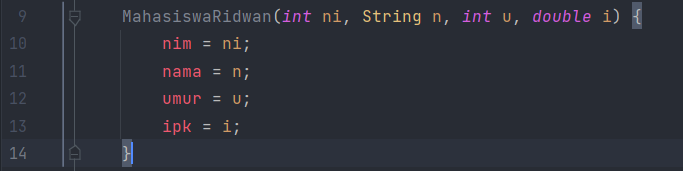
## Langkah 3

Buat class Mahasiswa, kemudian deklarasikan atribut berikut ini:



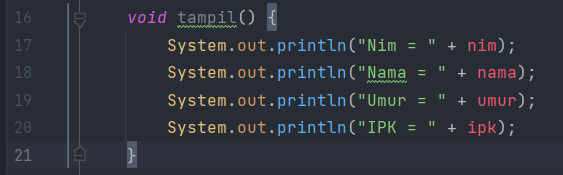
## Langkah 4

Buatlah konstruktor dengan nama Mahasiswa dengan parameter (int ni, String n, int u, double i) kemudian Isi konstruktor tersebut dengan kode berikut!



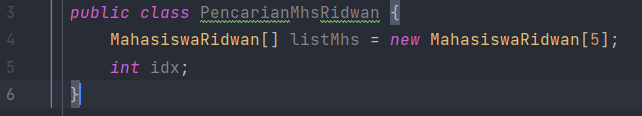
## Langkah 5

Buatlah method tampil bertipe void.

1

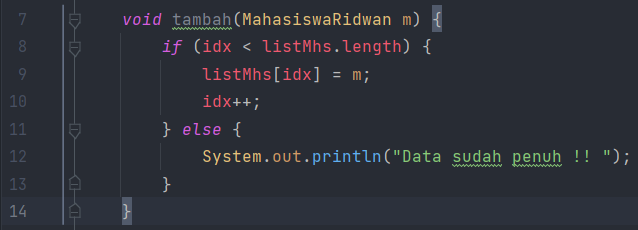
## Langkah 6

Buat class baru dengan nama PencarianMhs seperti di bawah ini!



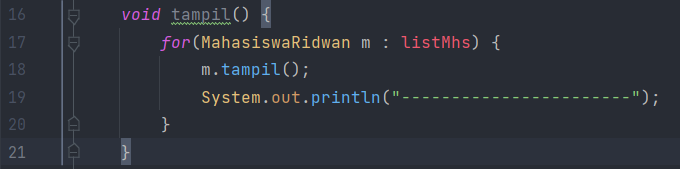
## Langkah 7

Tambahkan method tambah() di dalam class tersebut! Method tambah() digunakan untuk menambahkan objek dari class Mahasiswa ke dalam atribut listMhs.



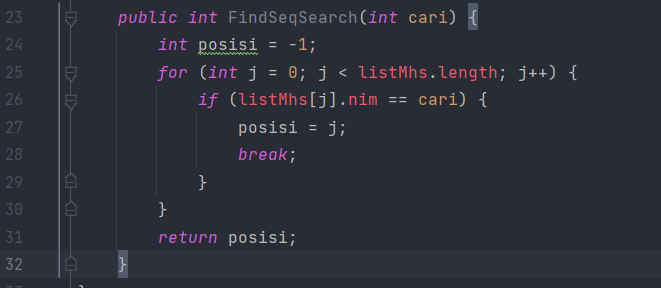
## Langkah 8

Tambahkan method tampil() di dalam class PencarianMhs! Method tampil() digunakan untuk menampilkan semua data mahasiswa-mahasiswa yang ada di dalam class tersebut! Perhatikan penggunaan sintaks for yang agak berbeda dengan for yang telah dipelajari sebelumnya, meskipun secara konsep sebenarnya mirip.



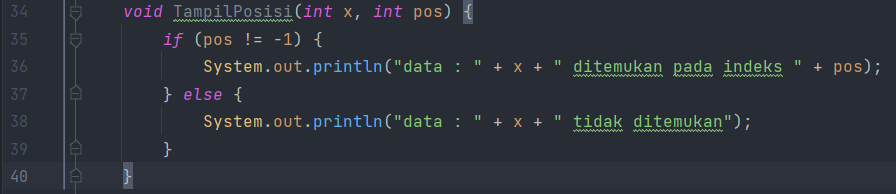
## Langkah 9

Tambahkan method FindSeqSearch bertipe integer dengan parameter cari bertipe integer. Kemudian Deklarasikan isi method FindSeqSearch dengan algoritma pencarian data menggunakan teknik sequential searching.



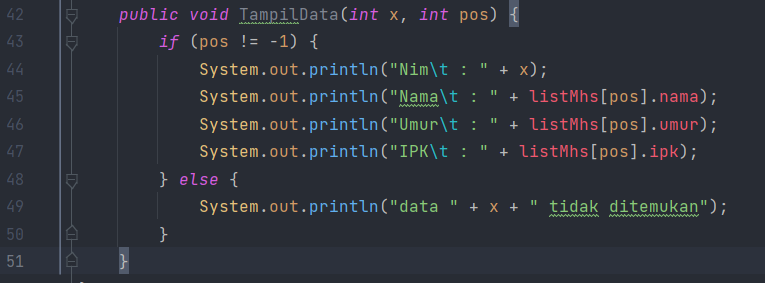
## Langkah 10

Buatlah method Tampilpoisisi bertipe void dan Deklarasikan isi dari method Tampilpoisisi



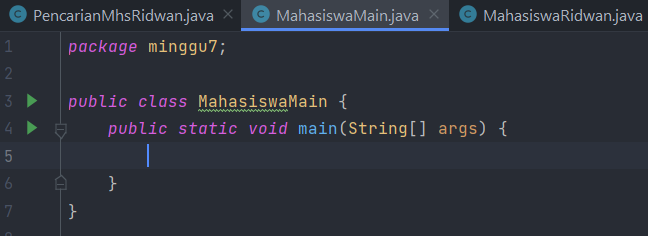
## Langkah 11

Buatlah method TampilData bertipe void dan Deklarasikan isi dari method TampilData.



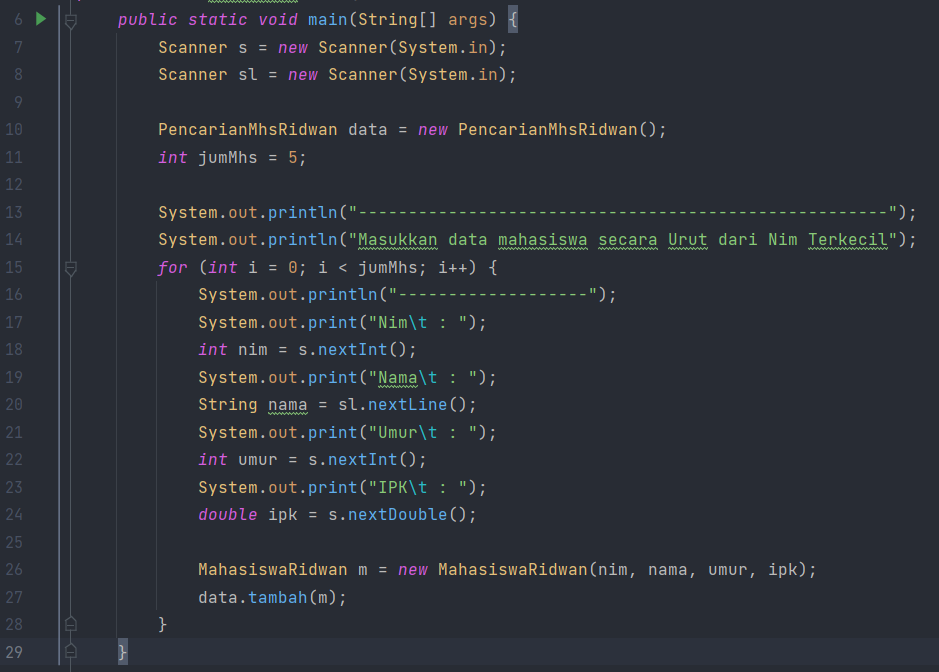
## Langkah 12

Buatlah class baru dengan nama MahasiswaMain tambahkan method main seperti pada gambar berikut!



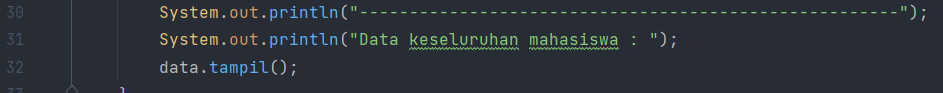
## Langkah 13

Di dalam method main(), buatlah sebuah objek PencarianMhs dan buatlah 5 objek mahasiswa kemudian tambahkan semua objek mahasiswa tersebut dengan memanggil fungsi tambah pada objek PencarianMhs.



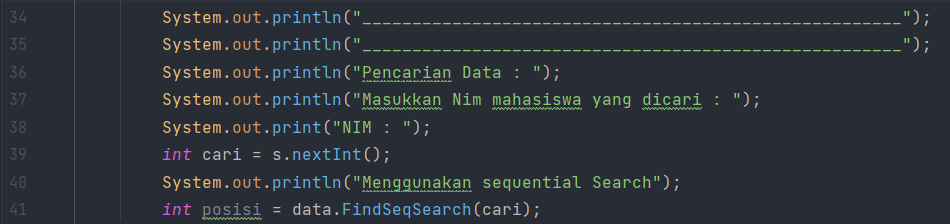
## Langkah 14

Panggil method tampil() untuk melihat semua data yang telah dimasukan.



## Langkah 15

Untuk melakukan pencarian berdasarkan NIM mahasiswa. Buatlah variable cari yang dapat menampung masukan dari keyboard lalu panggil method FindSeqSearch dengan isi parameternya adalah variable cari.



## Langkah 16

Lakukan pemanggilan method Tampilposisi dari class PencarianMhs.



## Langkah 17

Lakukan pemanggilan method TampilData dari class PencarianMhs.

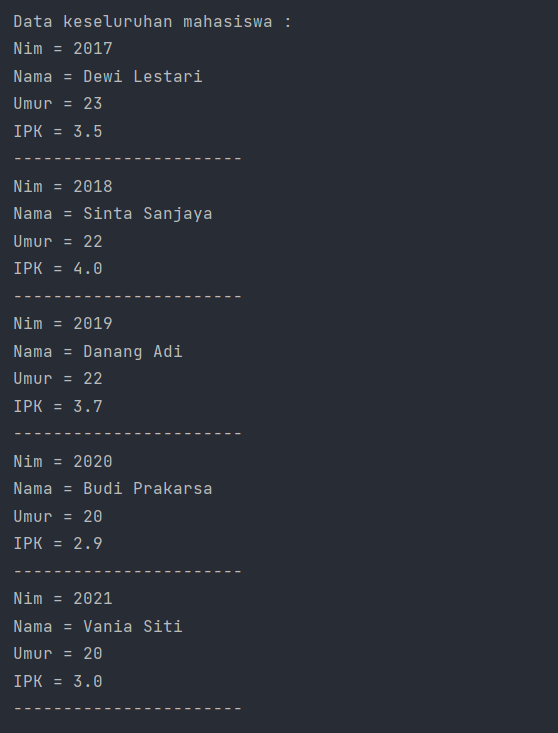
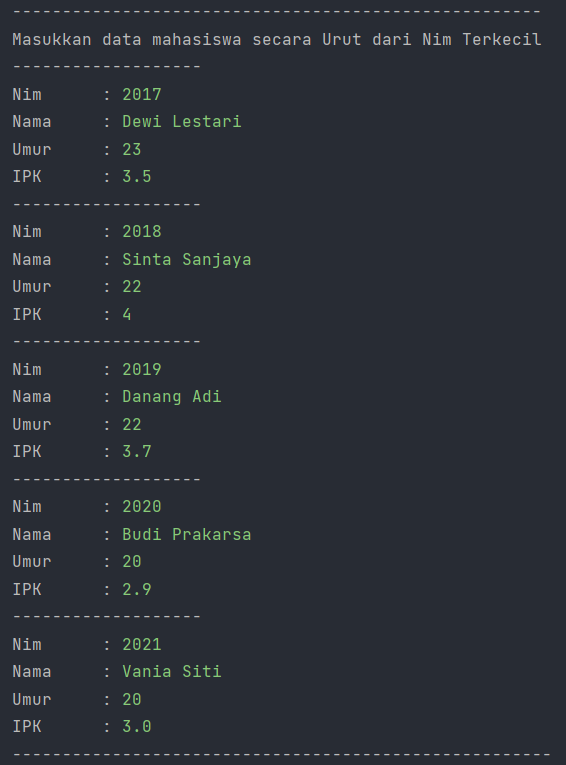


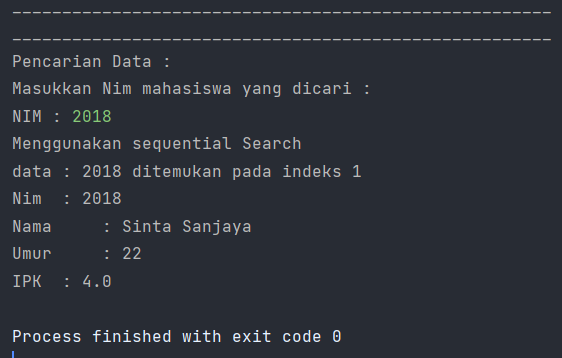
## Langkah 18

Jalankan dan amati hasilnya.

## Verifikasi Hasil Percobaan

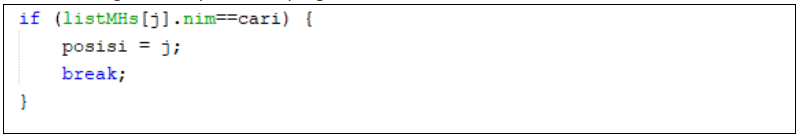
Cocokkan hasil kode program anda dengan gambar berikut ini.





### Pertanyaan

1. Jelaskan perbedaan metod TampilData dan Tampilposisi pada class PencarianMhs
2. Jelaskan fungsi break pada kode program dibawah ini!



1. Jika Data Nim yang dimasukkan tidak terurut dari kecil ke besar. Apakah program masih dapat berjalan? Apakah hasil yang dikeluarkan benar? Mengapa demikian!

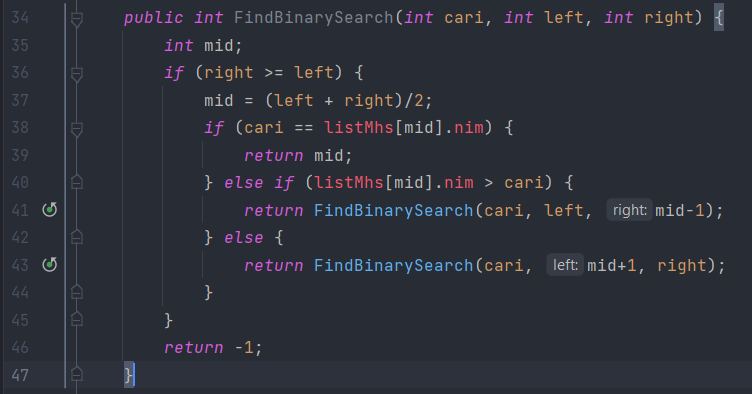
### Jawaban

1. Untuk method tampilData jika keyword ditemukan akan menampilkan isi dari data yang dicari sedangkan untuk method tampilPosisi akan menampilkan posisi indeks dari keyword yang dicari.
2. Fungsi break pada program tersebut adalah jika nilai listMhs[j].nim sama dengan nilai keyword yang dicari maka akan keluar dari perulangan.
3. Masih bisa berjalan karena ini menggunakan algoritme pencarian sequential yang tidak mempedulikan urutan data.

# Searching / Pencarian Menggunakan Binary Search

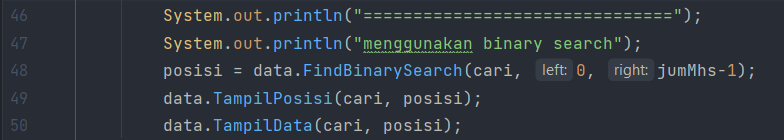
## Langkah 1

Pada percobaan 6.2.1 (sequential search) tambahkan method FindBinarySearch bertipe integer pada class PencarianMhs. Kemudian Deklarasikan isi method FindBinarySearch dengan algoritma pencarian data menggunakan teknik binary searching.



## Langkah 2

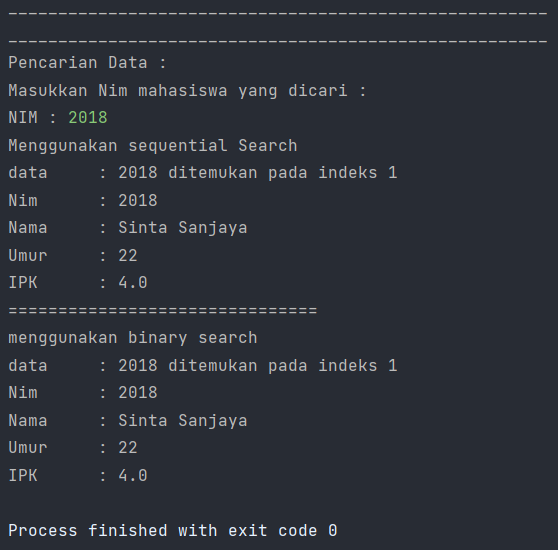
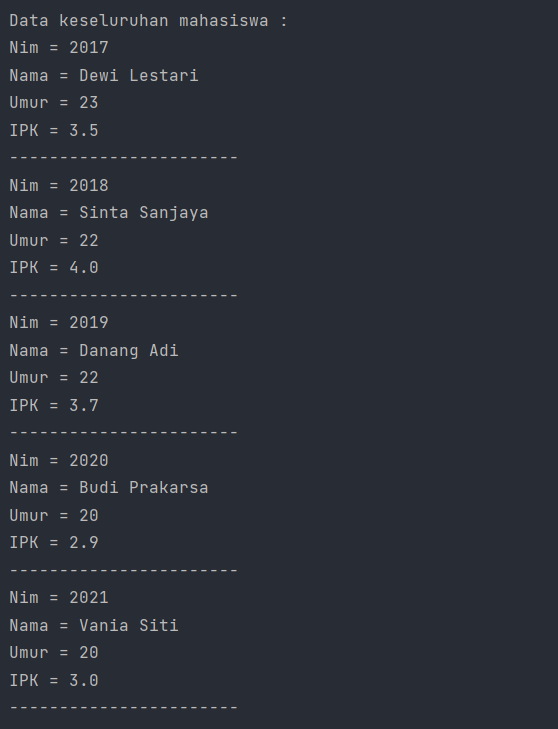
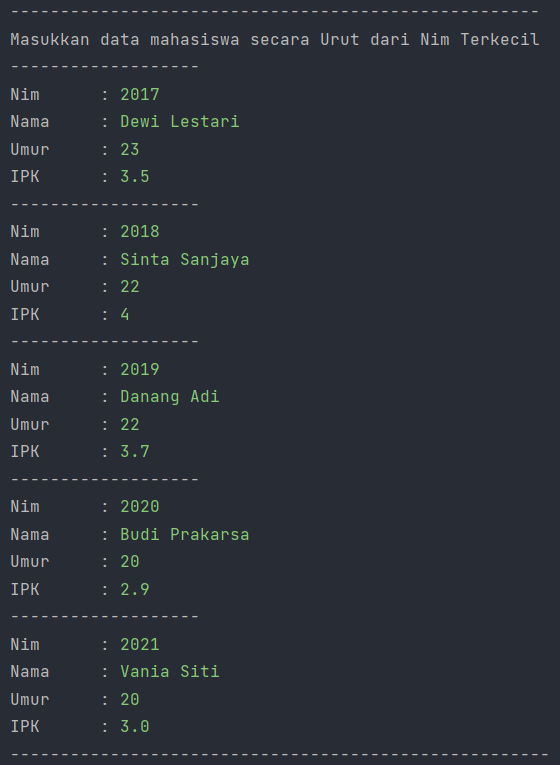
Panggil method FindBinarySearch terdapat pada class PencarianMhs di kelas Mahasiswamain. Kemudia panggil method tampilposisi dan tampilData



## Langkah 3

Jalankan dan amati hasilnya.

## Verifikasi Hasil Percobaan

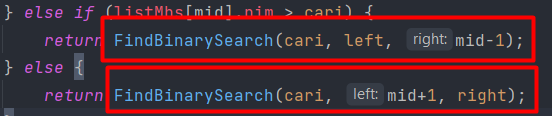


### Pertanyaan

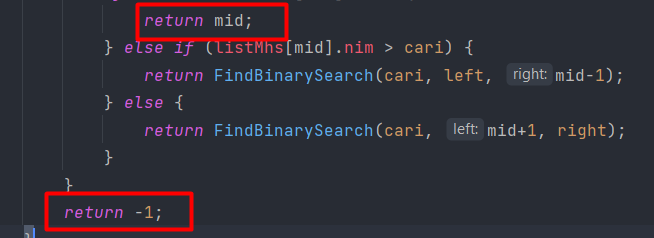
1. Tunjukkan pada kode program yang mana proses divide dijalankan!
2. Tunjukkan pada kode program yang mana proses conquer dijalankan!
3. Jika data Nim yang dimasukkan tidak urut. Apakah program masih dapat berjalan? Mengapa demikian!
4. Jika Nim yang dimasukkan dari NIM terbesar ke terkecil (missal : 20215, 20214, 20212, 20211, 20210) dan elemen yang dicari adalah 20210. Bagaimana hasil dari binary search? Apakah sesuai? Jika tidak sesuai maka ubahlah kode program binary seach agar hasilnya sesuai
5. Modifikasilah program diatas yang mana jumlah mahasiswa yang di inputkan sesuai dengan masukan dari keyboard.

### Jawaban

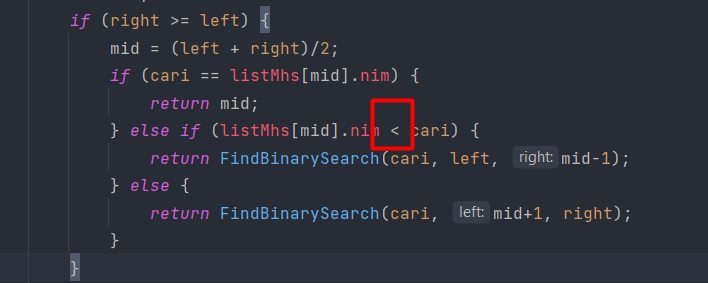
1. Berikut adalah kode program yang melakukan proses divide



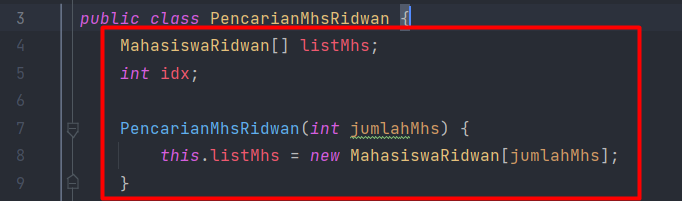
1. Berikut adalah kode program yang melakukan proses conquer



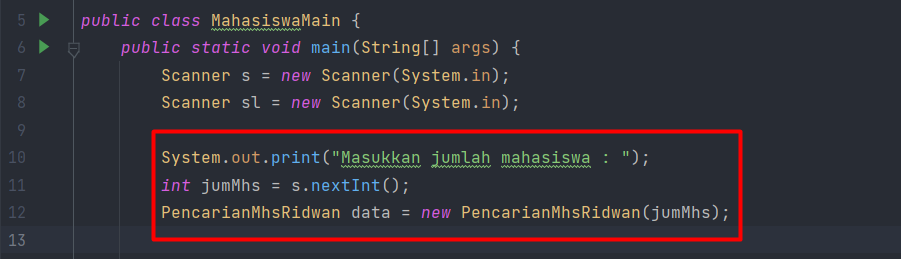
1. Tidak bisa karena jika menggunakan algoritma binary search data harus di urutkan terlebih dahulu
2. Hasilnya tidak sesuai maka kita harus mengubah sedikit algoritma nya dengan mengubah tanda lebih dari menjadi kurang dari maka hasilnya akan sesuai.



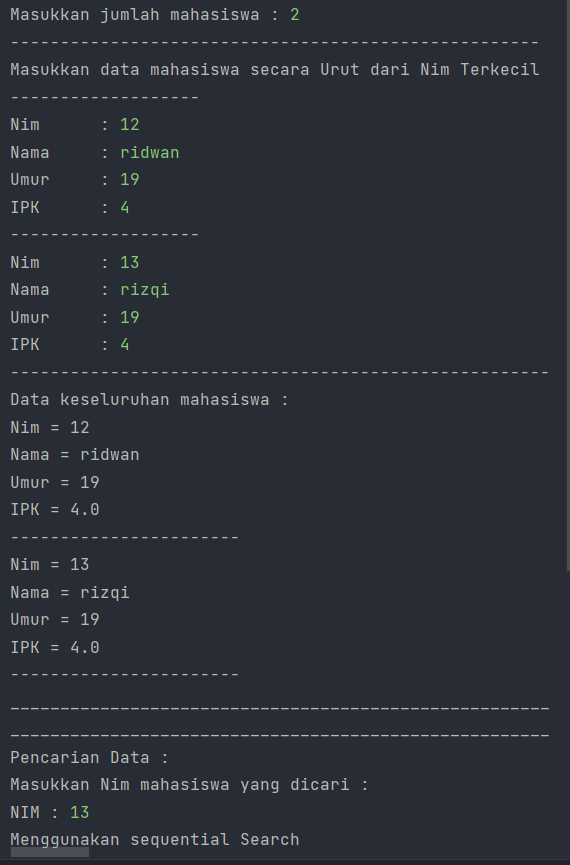
1. Pada class PencarianMhs



Pada class MahasiswaMain



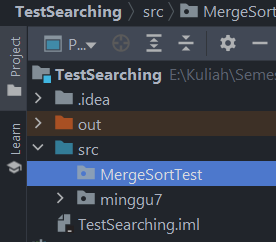
Output



# Percobaan Pengayaan Divide and Conquer

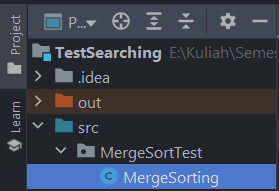
## Langkah 1

Buatlah Package baru pada NetBeans dengan nama MergeSortTest



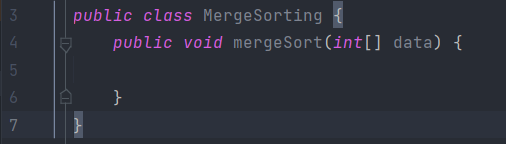
## Langkah 2

Tambahkan class MergeSorting pada package tersebut



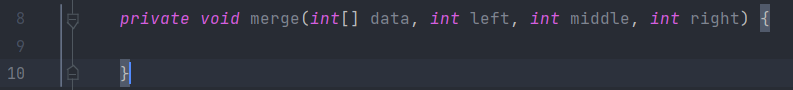
## Langkah 3

Pada class MergeSorting buatlah method mergeSort yang menerima parameter data array yang akan diurutkan



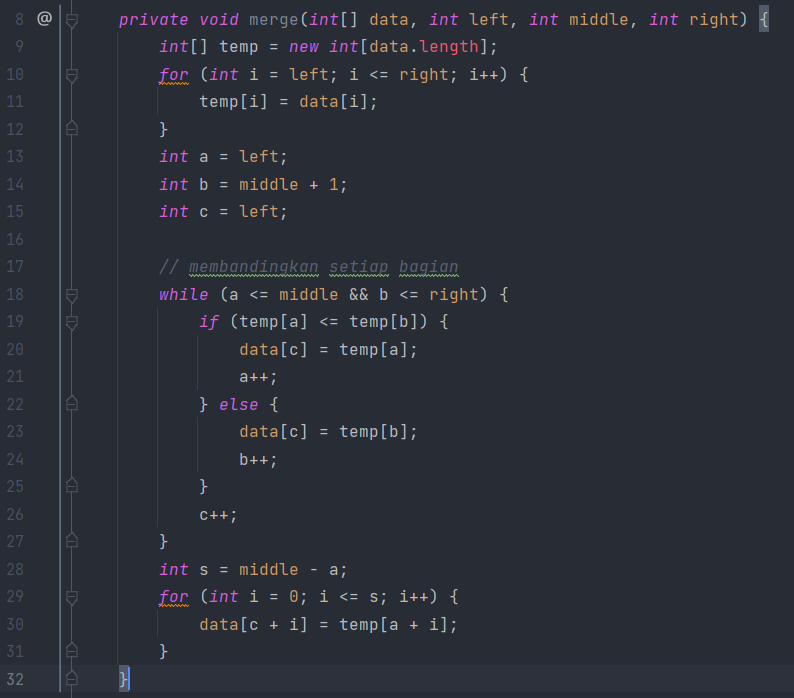
## Langkah 4

Buatlah method merge untuk melakukan proses penggabungan data dari bagian kiri dan kanan.



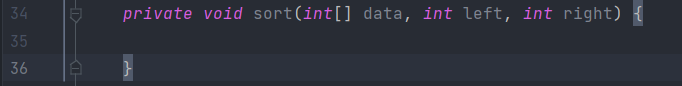
## Langkah 5

Implementasikan proses merge sebagai berikut.



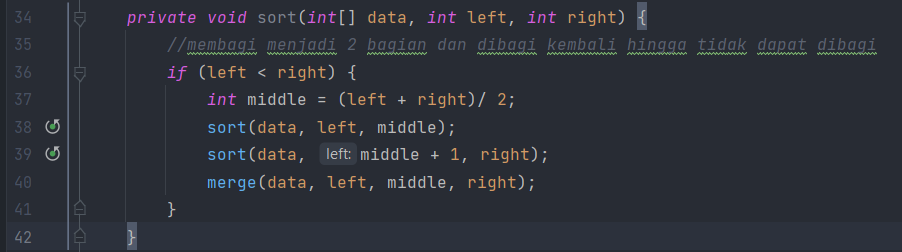
## Langkah 6

Buatlah method sort



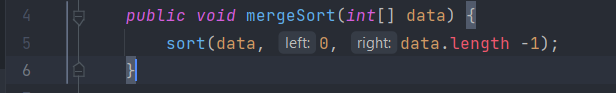
## Langkah 7

Implementasikan kode berikut pada method sort



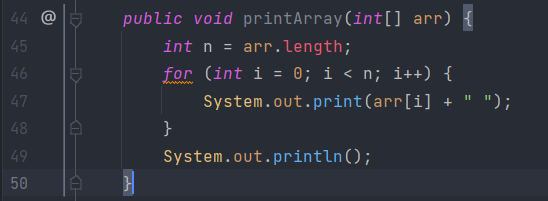
## Langkah 8

Pada method mergeSort, panggil method sort dengan parameter data yang ingin diurutkan serta range data awal sampai dengan akhir.



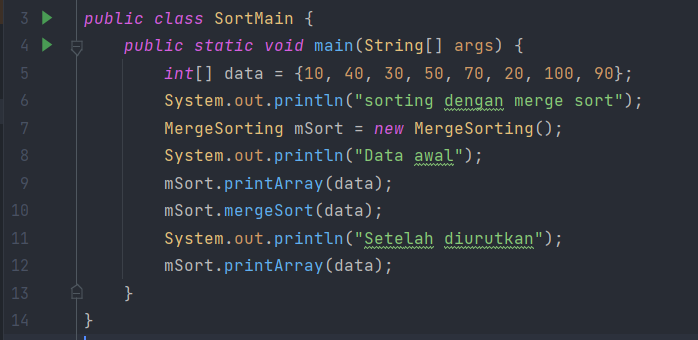
## Langkah 9

Tambahkan method printArray

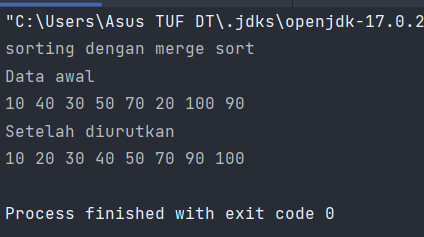


## Langkah 10

Sebagai langkah terakhir, deklarasikan data yang akan diurutkan kemudian panggil proses sorting pada class SortMain



## Verifikasi Hasil Percobaan



# Latihan Praktikum

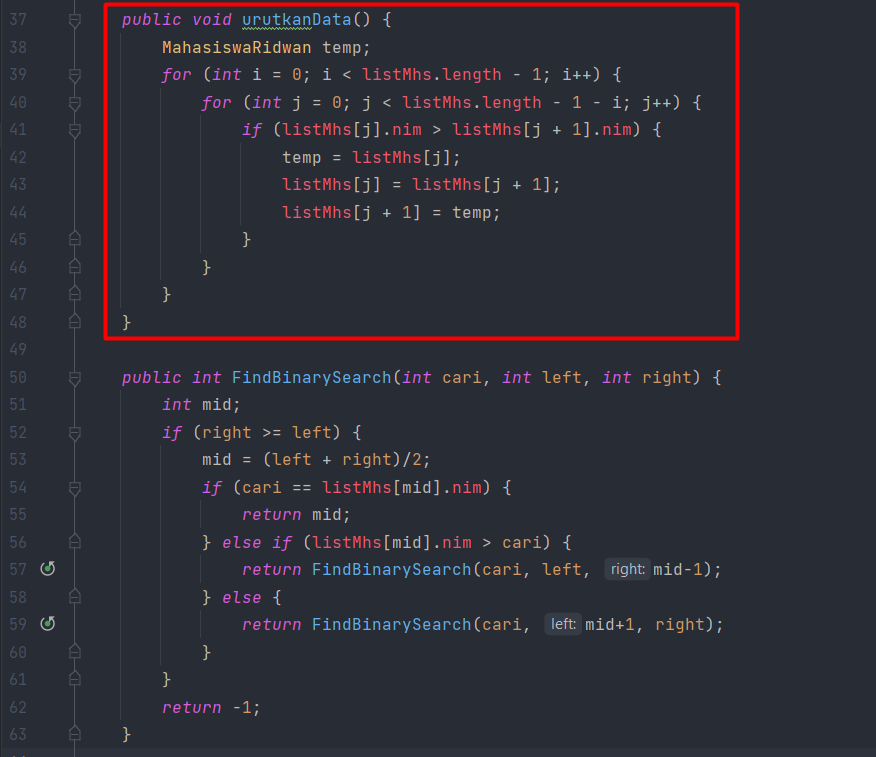
## Soal nomor 1

Modifikasi percobaan searching diatas dengan ketentuan berikut ini

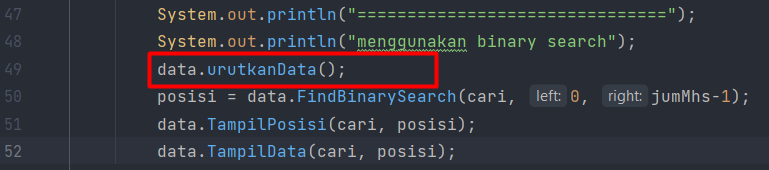
* Sebelum dilakukan searching dengan binary search data harus dilakukan pengurutan dengan menggunakan algoritma sorting (pilih salah satu algoritma sorting dari pertemuan sebelumnya)

## Jawaban nomor 1

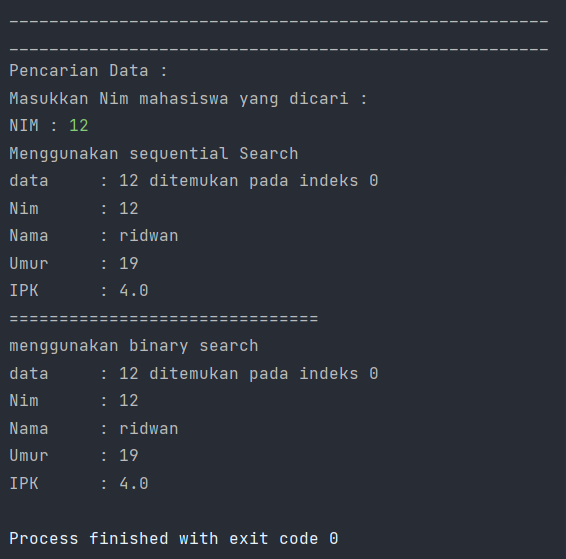
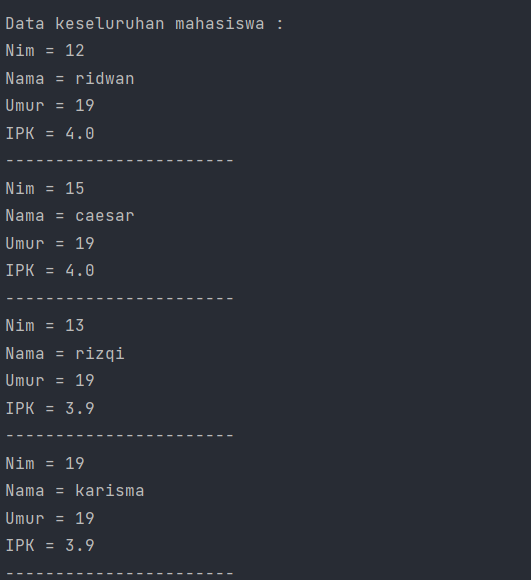
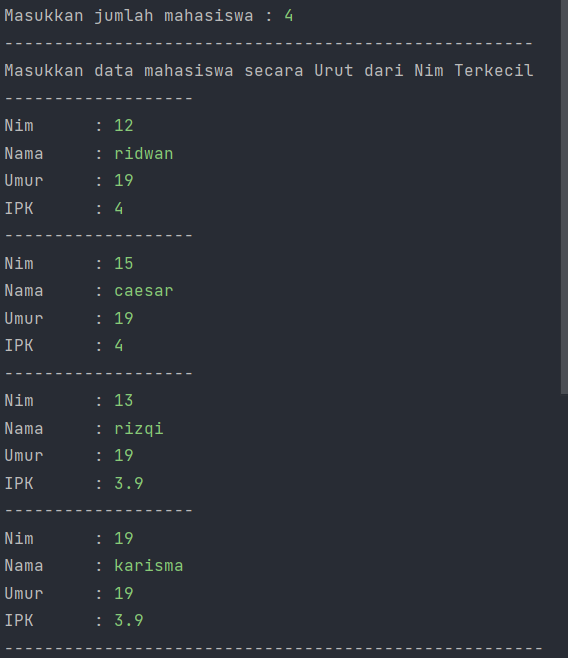
### Modifikasi source code class PencarianMhs



### Modifikasi source code class MahasiswaMain



### Output



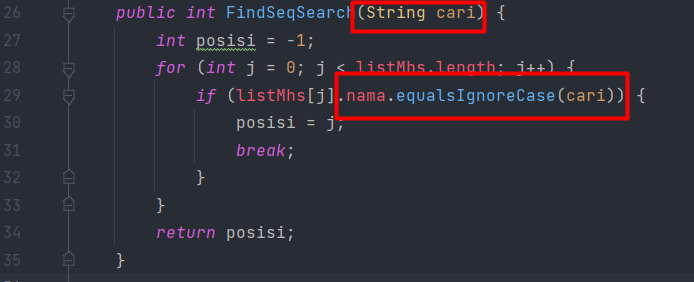
## Soal nomor 2

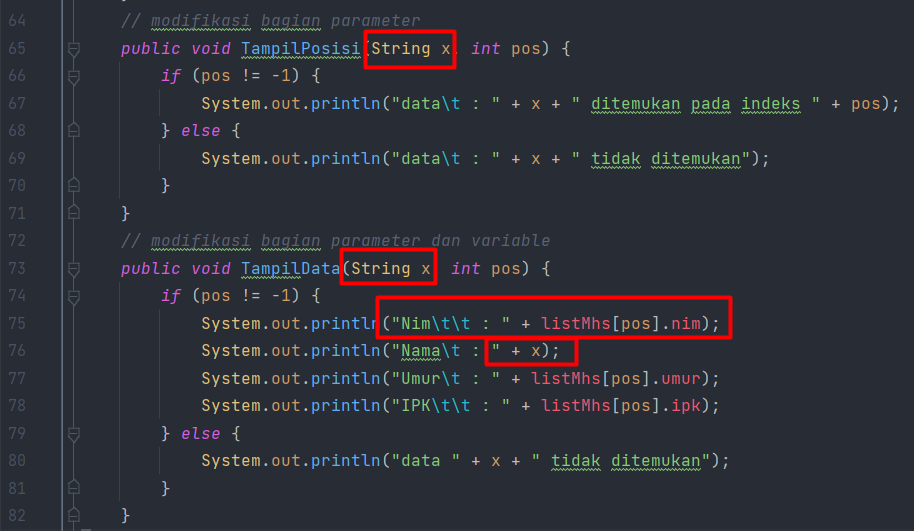
Modifikasi percobaan searching diatas dengan ketentuan berikut ini

* Pencarian dilakukan berdasarkan Nama Mahasiswa (gunakan Algoritma Sequential Search)
* Jika terdapat nama yang sama? Bagaimana keluaran dari kode program ketika pencarian dilakukan pada nama yang sama!

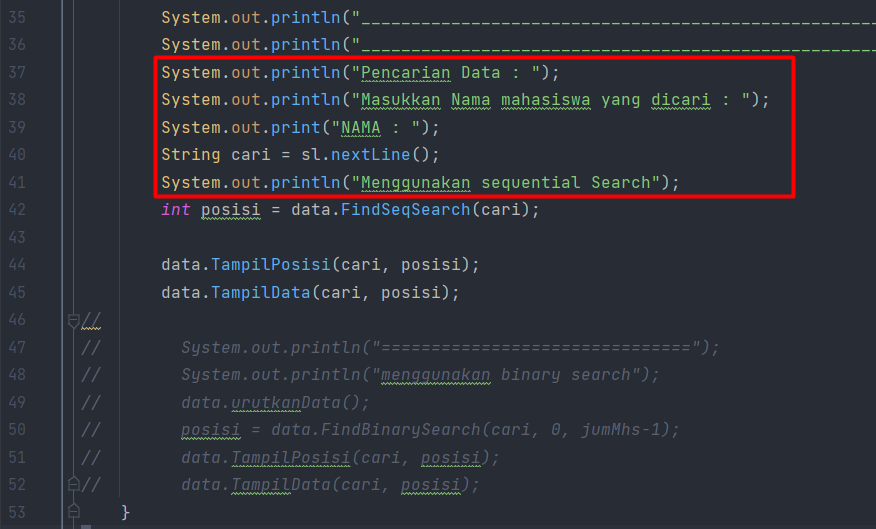
## Jawaban nomor 2

### Modifikasi source code class PencarianMhs

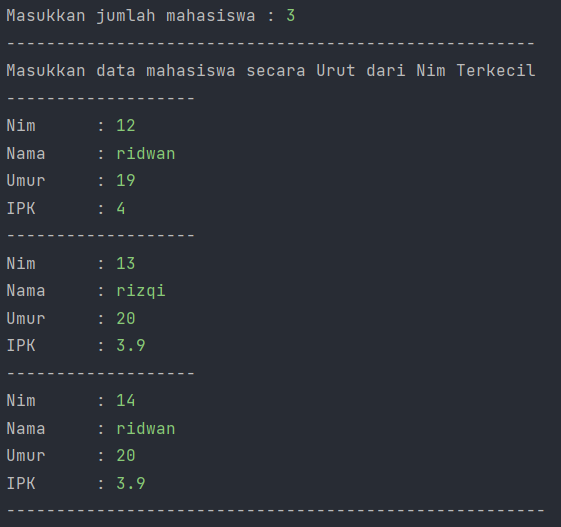
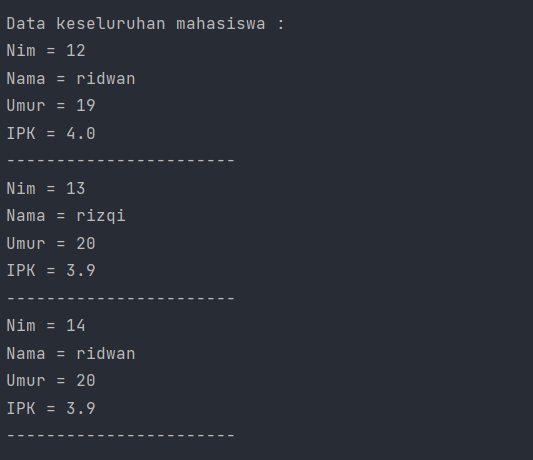


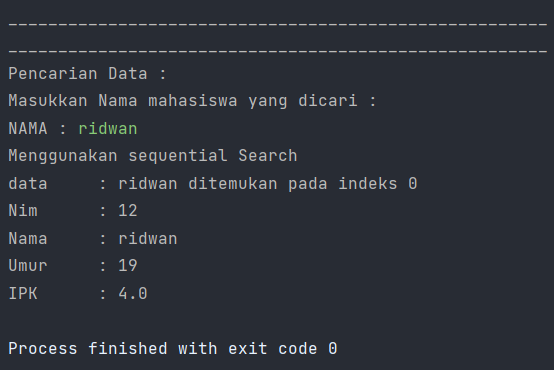


### Modifikasi source code class MahasiswaMain



### Output

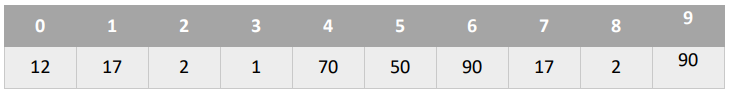
 



Jika terdapat nama yang sama maka pada program diatas akan ditampilkan nama yang pertama kali ditemukan.

## Soal nomor 3

Terdapat sebuah data array 1 dimensi sebagai berikut

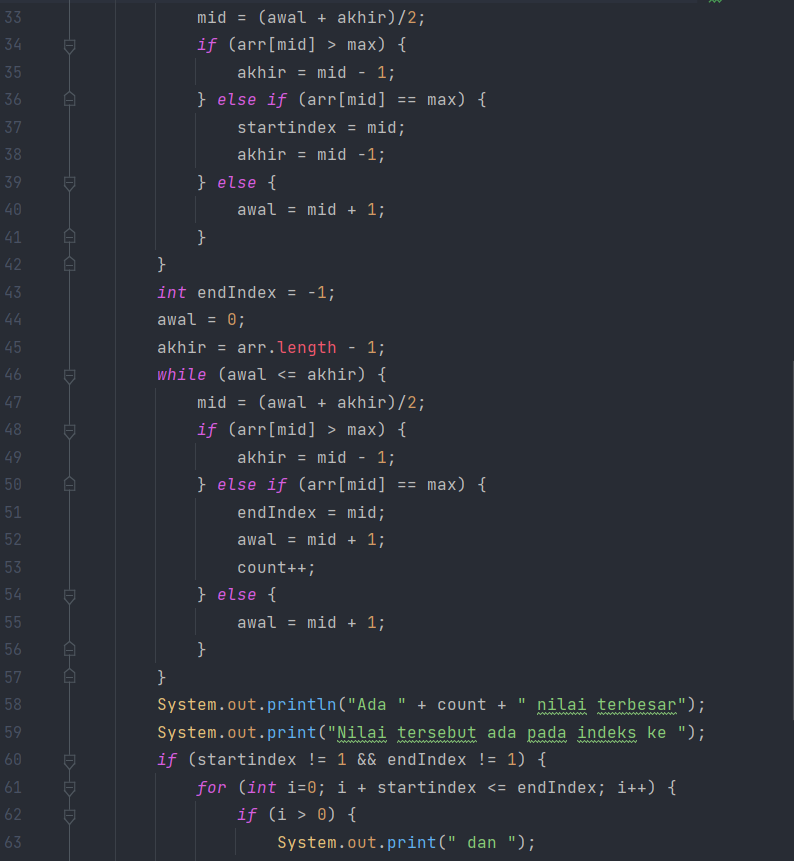


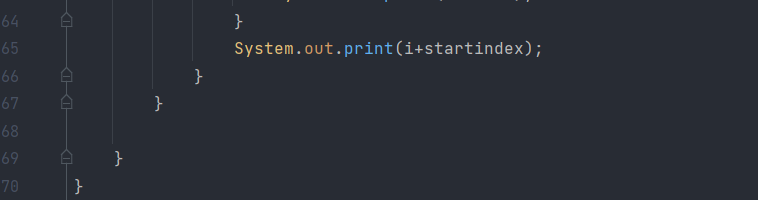
Buatlah program untuk mengurutkan array tersebut (boleh memilih metode pengurutan) selanjutnya lakukan pencarian dan mencetak isi array yang nilainya terbesar, dan mencetak ada berapa buah nilai terbesar tersebut serta berada dilokasi mana saja nilai terbesar tersebut! (menggunakan binary search)

## Jawaban nomor 3

### Source code







### Output

