**LAPORAN PRAKTIKUM**

**Politeknik Negeri Malang**

**Praktikan**

**ALGORITMA DAN STRUKTUR DATA**

****

2022

2141720183

RIDWAN CAESAR RIZQI KARISMA BIWARNI

TI 1C

Daftar Isi

[Percobaan 1 4](#_Toc106180850)

[Langkah 1 4](#_Toc106180851)

[Langkah 2 4](#_Toc106180852)

[Verifikasi Hasil Percobaan 5](#_Toc106180853)

[Pertanyaan Percobaan 5](#_Toc106180854)

[Jawaban Percobaan 5](#_Toc106180855)

[Percobaan 2 7](#_Toc106180856)

[Langkah 1 7](#_Toc106180857)

[Langkah 2 7](#_Toc106180858)

[Verifikasi Hasil Percobaan 8](#_Toc106180859)

[Pertanyaan percobaan 8](#_Toc106180860)

[Jawaban percobaan 8](#_Toc106180861)

[Percobaan 3 11](#_Toc106180862)

[Langkah 1 11](#_Toc106180863)

[Langkah 2 11](#_Toc106180864)

[Langkah 3 11](#_Toc106180865)

[Langkah 4 12](#_Toc106180866)

[Langkah 5 12](#_Toc106180867)

[Verifikasi Hasil Percobaan 13](#_Toc106180868)

[Pertanyaan Percobaan 13](#_Toc106180869)

[Jawaban Percobaan 13](#_Toc106180870)

[Tugas Praktikum 16](#_Toc106180871)

[Nomor 1 16](#_Toc106180872)

[Jawaban Nomor 1 17](#_Toc106180873)

[Source code Class Film\_26 17](#_Toc106180874)

[Source code Class MainTugas1\_26 18](#_Toc106180875)

[Output Menu Awal dan Tambah Data 19](#_Toc106180876)

[Output Cetak Semua Data Film 19](#_Toc106180877)

[Output Judul Film di Tumpukan Teratas 19](#_Toc106180878)

[Output Hapus Film Teratas 20](#_Toc106180879)

[Nomor 2 21](#_Toc106180880)

[Jawaban Nomor 2 22](#_Toc106180881)

[Source code Class Mahasiswa\_26 22](#_Toc106180882)

[Source code Class Nilai\_26 23](#_Toc106180883)

[Source code Class MataKuliah\_26 23](#_Toc106180884)

[Source code Class MainTugas2\_26 23](#_Toc106180885)

[Output Menu Awal dan Penambahan Data 25](#_Toc106180886)

[Output Tampil Nilai 26](#_Toc106180887)

[Output Pencarian Data Mahasiswa 26](#_Toc106180888)

[Output Pengurutan Data Nilai 26](#_Toc106180889)

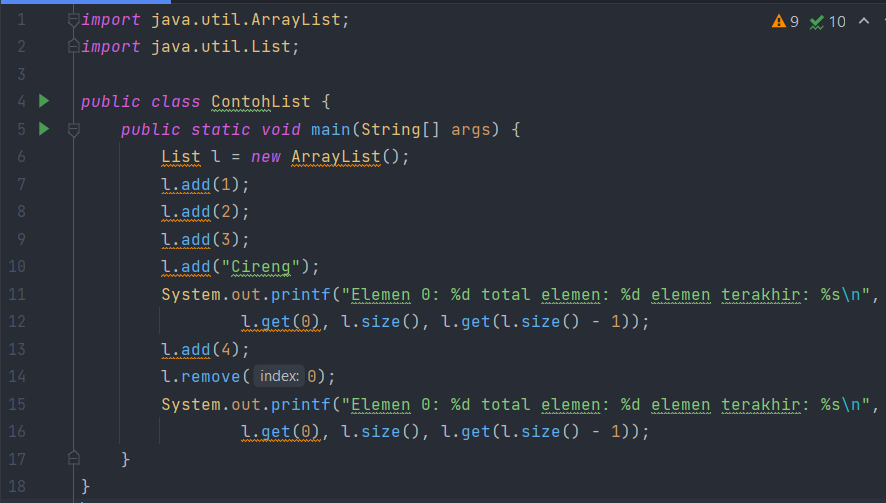
# Percobaan 1

Pada percobaan 1 ini akan dicontohkan penggunaan collection untuk menambahkan sebuah

elemen, mengakses elemen, dan menghapus sebuah elemen

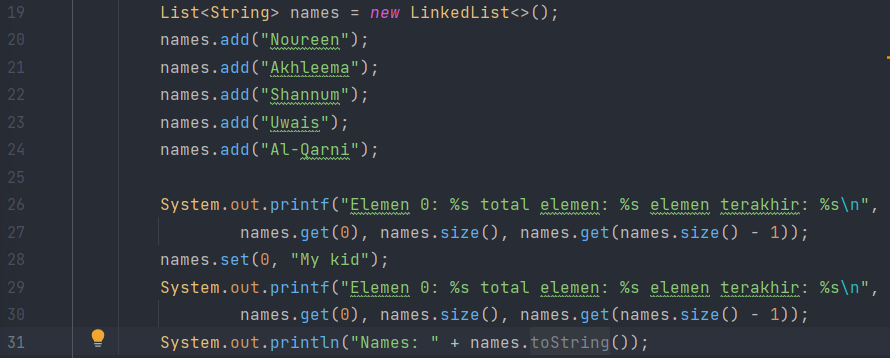
## Langkah 1

Buatlah sebuah class ContohList yang main methode berisi kode program seperti di bawah ini

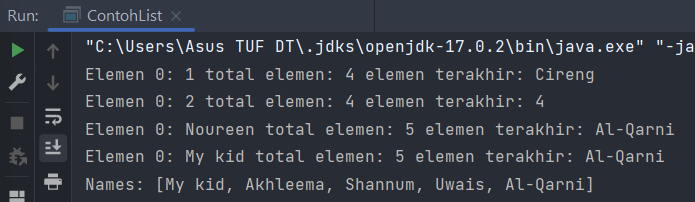


## Langkah 2

Tambahkan kode program untuk menggunakan collection dengan aturan penulisan kode program seperti berikut



## Verifikasi Hasil Percobaan

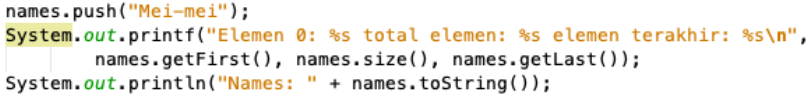


### Pertanyaan Percobaan

1. Perhatikan baris kode 25-36, mengapa semua jenis data bisa ditampung ke dalam sebuah Arraylist?
2. Modifikasi baris kode 25-36 seingga data yang ditampung hanya satu jenis atau spesifik tipe tertentu!
3. Ubah kode pada baris kode 38 menjadi seperti ini



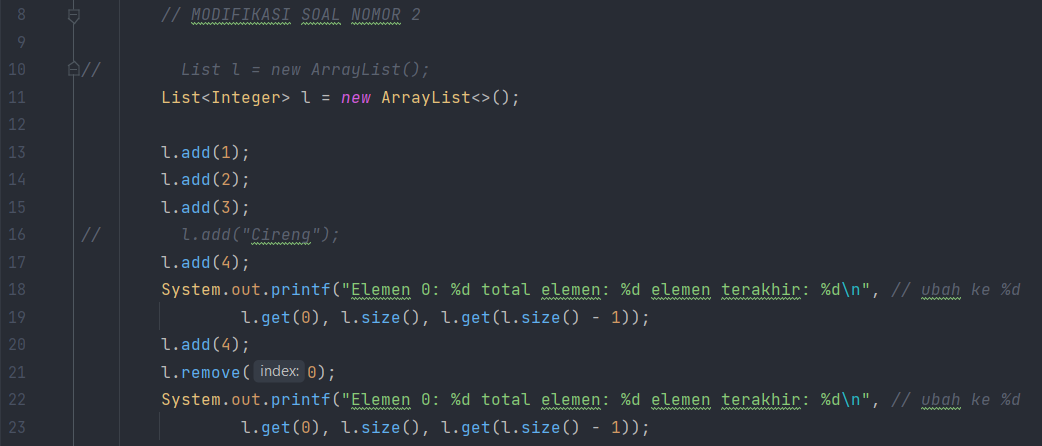
1. Tambahkan juga baris berikut ini, untuk memberikan perbedaan dari tampilan yang sebelumnya.



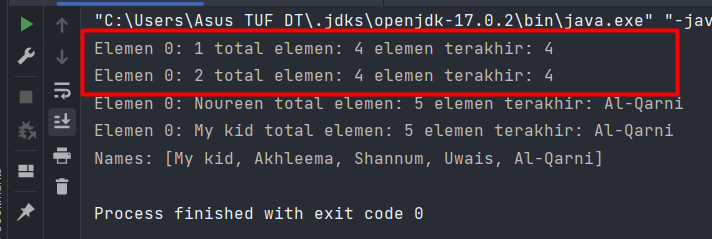
1. Dari penambahan kode tersebut, silakan dijalankan dan apakah yang dapat Anda jelaskan!

### Jawaban Percobaan

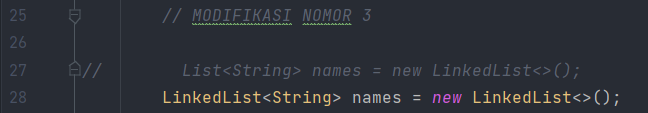
1. Karena array list menggunakan type yang tidak disebutkan secara spesifik maka dapat di isi oleh semua tipe data.
2. Source code



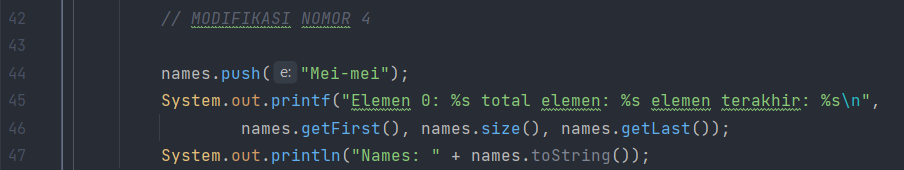
Output



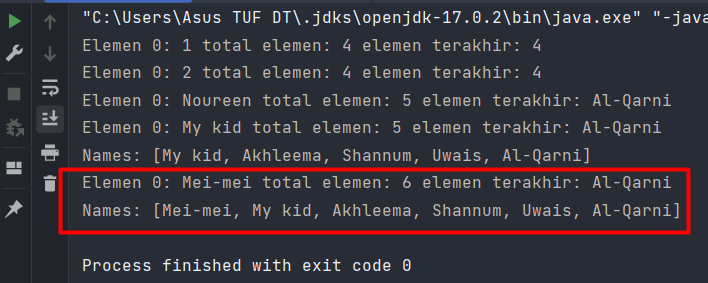
1. Source code modifikasi



1. Source code



1. Output



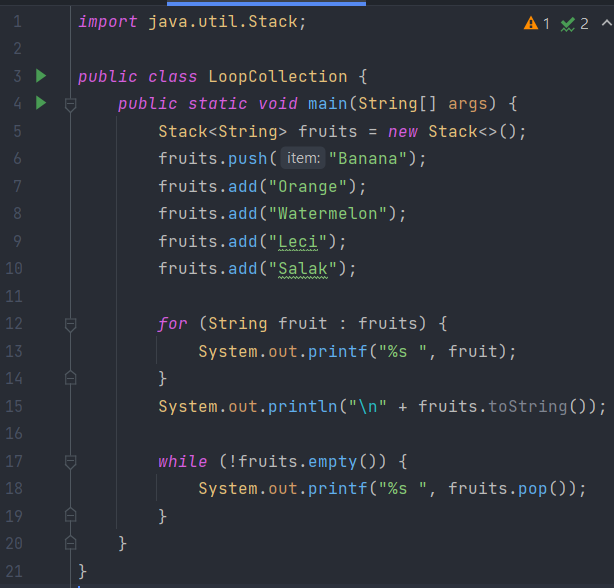
Pada source code tersebut melakukan penambahan data pada depan list dengan method push, kemudian menampilkan elemen pertama pada list menggunakan method getFirst, menampilkan jumlah elemen menggunakan method size, kemudian menampilkan elemen terakhir pada list menggunakan method getLast.

# Percobaan 2

Pada percobaan 2 ini akan dibuat beberapa method untuk menampilkan beberapa cara yang dapat dilakukan untuk mengambil/menampilkan elemen pada sebuah collection. Silakan ikutilah Langkah-langkah di bawah ini

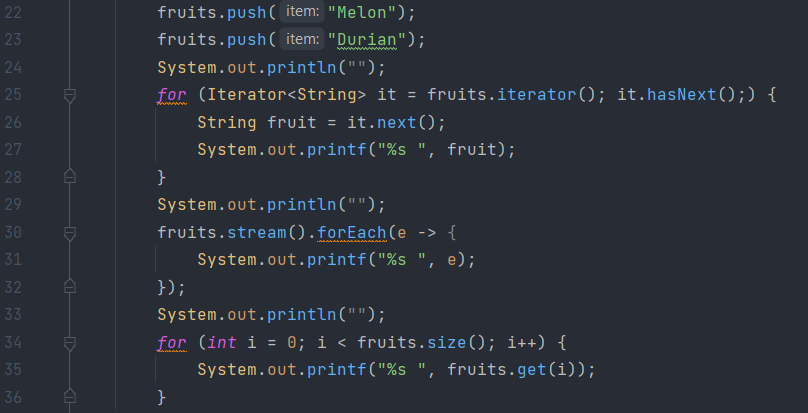
## Langkah 1

Buatlah class dengan nama LoopCollection serta tambahkan method main yang isinya adalah sebagai berikut

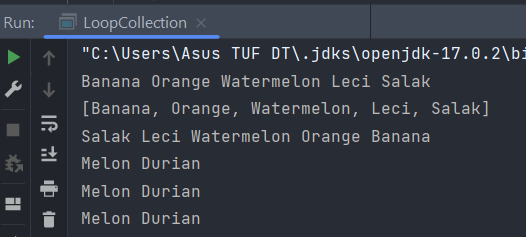


## Langkah 2

Tambahkan potongan kode berikut ini dari yang sebelumnya agar proses menampilkan elemen pada sebuah stack bervariasi.



## Verifikasi Hasil Percobaan

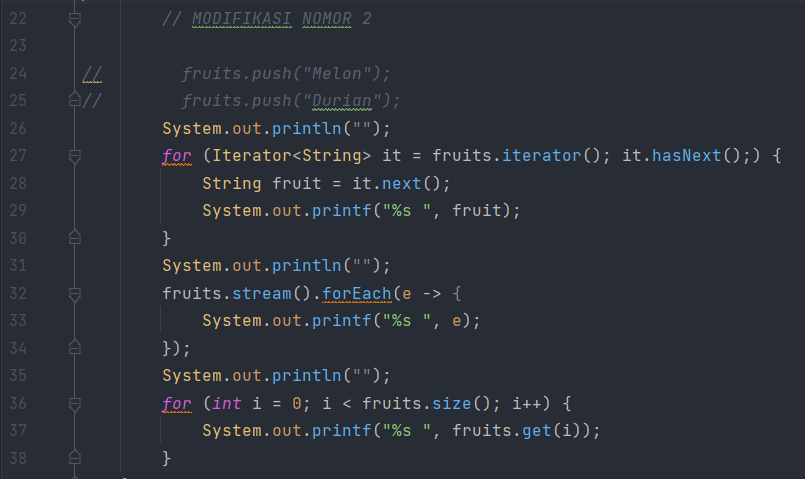


### Pertanyaan percobaan

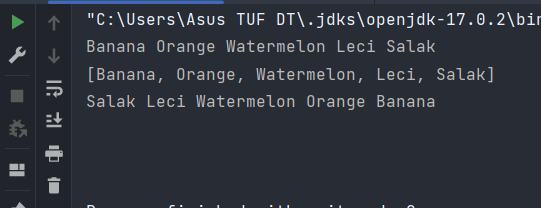
1. Apakah perbedaan fungsi push() dan add() pada objek fruits?
2. Silakan hilangkan baris 43 dan 44, apakah yang akan terjadi? Mengapa bisa demikian?
3. Jelaskan fungsi dari baris 46-49?
4. Silakan ganti baris kode 25, Stack menjadi List dan apakah yang terjadi? Mengapa bisa demikian?
5. Ganti elemen terakhir dari dari objek fruits menjadi “Strawberry”!
6. Tambahkan 3 buah seperti “Mango”,”guava”, dan “avocado” kemudian dilakukan sorting!

### Jawaban percobaan

1. Push berfungsi untuk memasukkan data ke top of stack, sedangkan add merupakan method inherited dari class java.util.Vector yang berfungsi untuk memasukkan data ke akhir Vector atau disini stack.
2. Source code

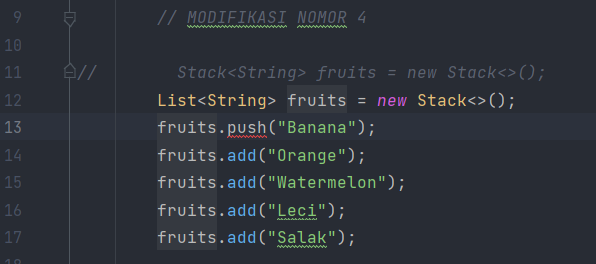


Output

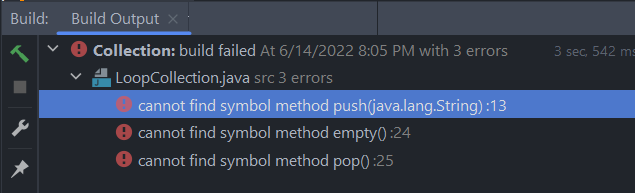


Data melon dan durian tidak ditampilkan karena data tersebut tidak pernah dimasukkan.

1. Baris kode tersebut digunakan untuk melakukan perulangan menggunakan Interface iterator dan menggunakan hasNext yang akan mengembalikan nilai true jika iteration masih mempunyai elemen.
2. Source code



Output

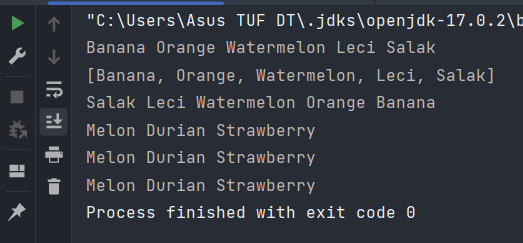


Hasilnya adalah error karena pada interface List tidak memiliki method push, empty dan pop.

1. Source code



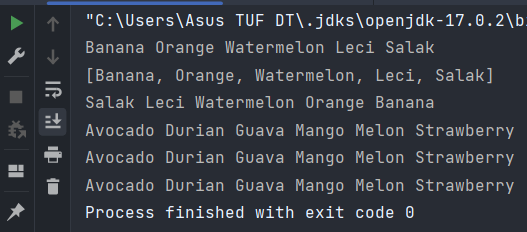
Output



1. Source code



Output



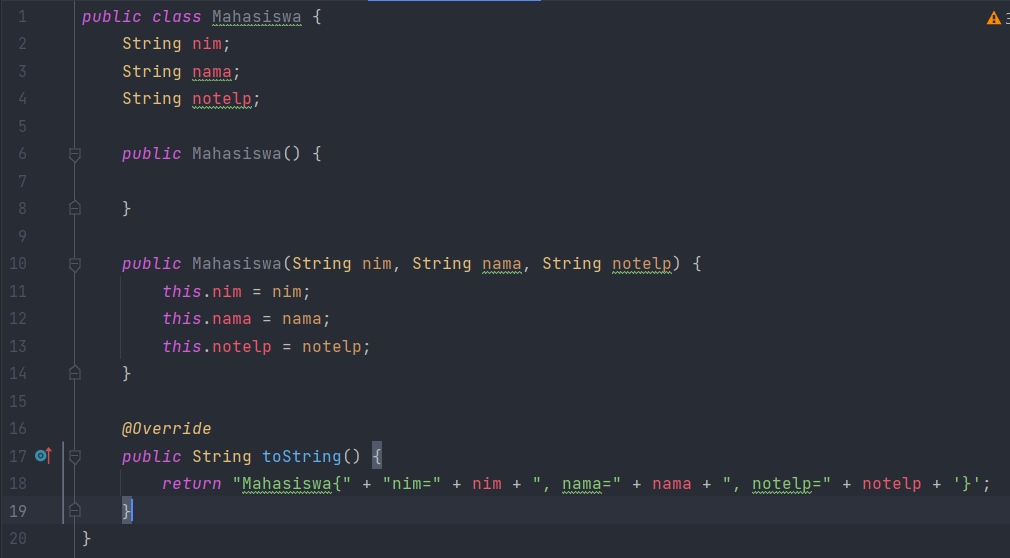
Data sudah diurutkan secara ascending.

# Percobaan 3

Pada percobaan 3 ini dilakukan uji coba untuk mengimplementasikan sebuah collection untuk menampung objek yang dibuat sesuai kebutuhan. Objek tersebut adalah sebuah objek mahasiswa dengan fungsi-fungsi umum seperti menambahkan, menghapus, mengubah, dan mencari.

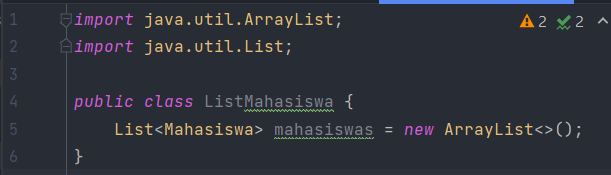
## Langkah 1

Buatlah sebuah class Mahasiswa dengan attribute, kontruktor, dan fungsi sebagai berikut



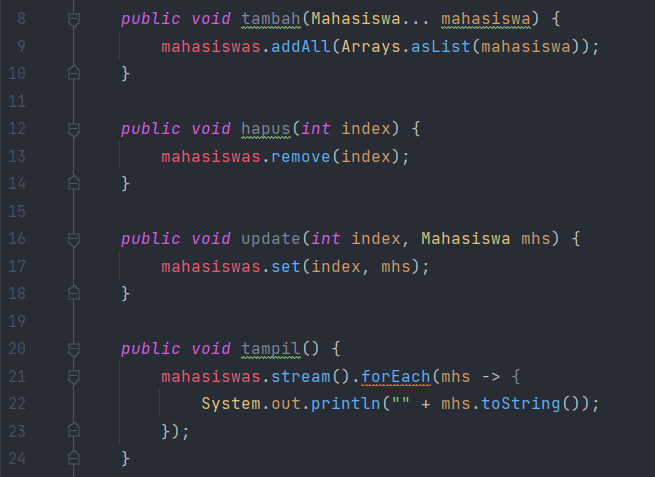
## Langkah 2

Selanjutnya, buatlah sebuah class ListMahasiswa yang memiliki attribute seperti di bawah ini



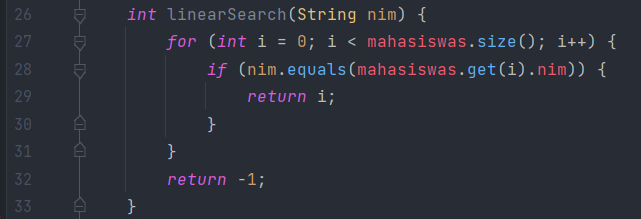
## Langkah 3

Method tambah(), hapus(), update(), dan tampil() secara berurut dibuat agar bisa melakukan operasi-operasi seperti yang telah disebutkan.



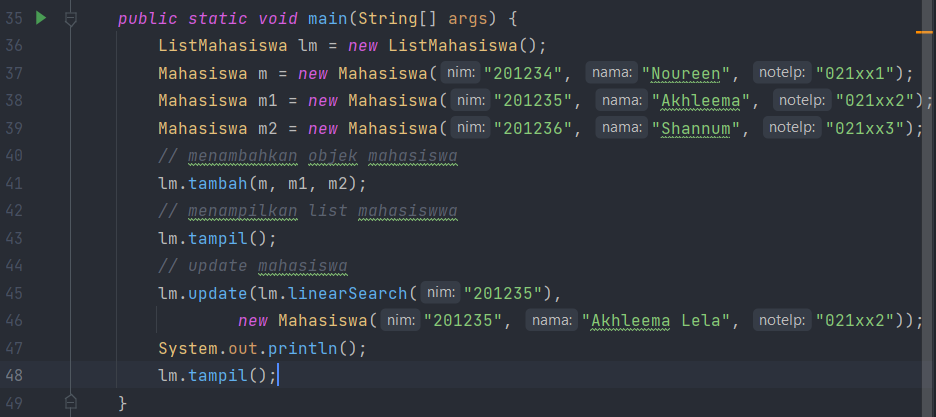
## Langkah 4

Untuk proses hapus, update membutuhkan fungsi pencarian terlebih dahulu yang potongan kode programnya adalah sebagai berikut

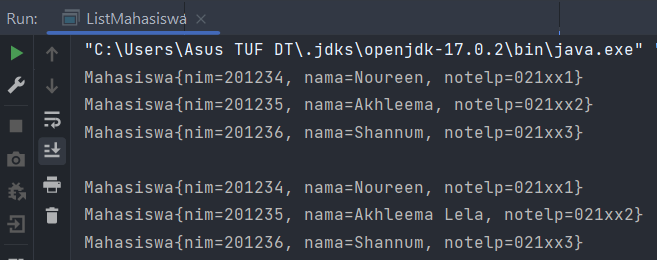


## Langkah 5

Pada class yang sama, tambahkan main method seperti potongan program berikut dan amati hasilnya!



## Verifikasi Hasil Percobaan



### Pertanyaan Percobaan

1. Pada fungsi tambah() yang menggunakan unlimited argument itu menggunakan konsep apa? Dan kelebihannya apa?
2. Pada fungsi linearSearch() di atas, silakan diganti dengan fungsi binarySearch() dari collection!
3. Tambahkan fungsi sorting baik secara ascending ataupun descending pada class tersebut!

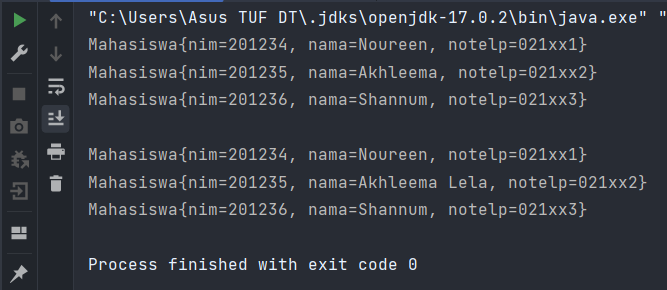
### Jawaban Percobaan

1. Menggunakan konsep varargs, kelebihan dari fitur ini merupakan dapat menerima banyak argumen maka menjadi mudah untuk membuat metode yang perlu mengambil sejumlah variabel argumen.
2. Source code

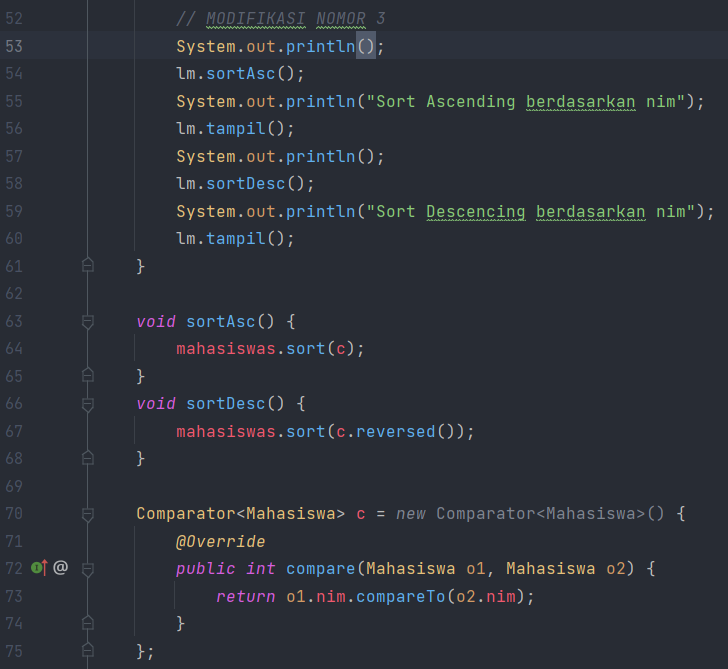


Disini saya membuat comparator untuk membandingkan nim pada objek mahasiswa.

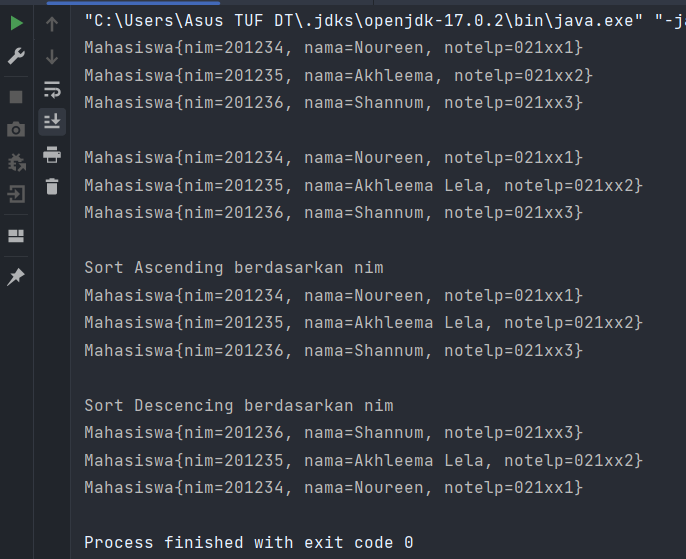
Output



1. Source code



Output



# Tugas Praktikum

## Nomor 1

Implementasikan stack menggunakan collection dengan contoh kasus tumpukan daftar film

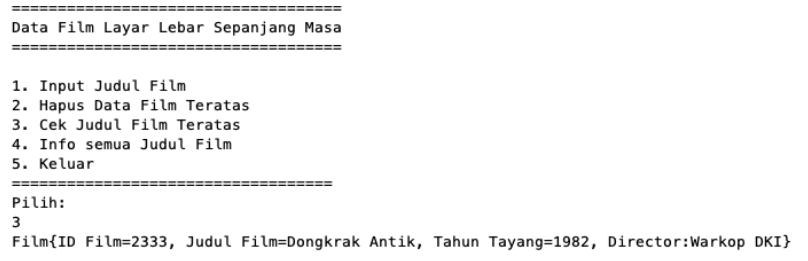
sesuai dengan fitur-fitur yang ditunjukkan pada gambar di bawah ini!Ilustrasi ProgramMenu Awal dan Tambah Data (Push)



Cetak Semua Data Film



Cek Judul Film di tumpukan teratas

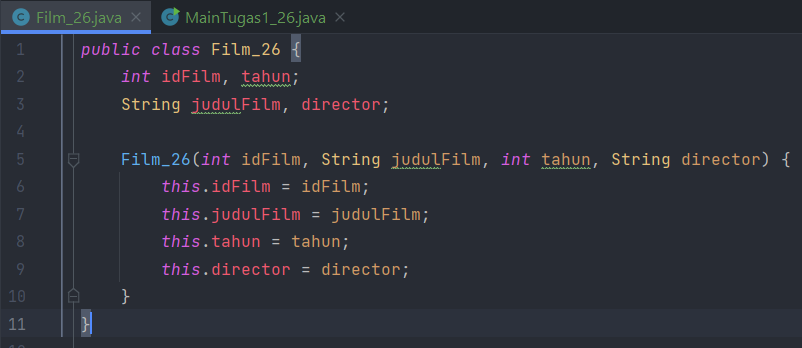


Hapus Film Teratas

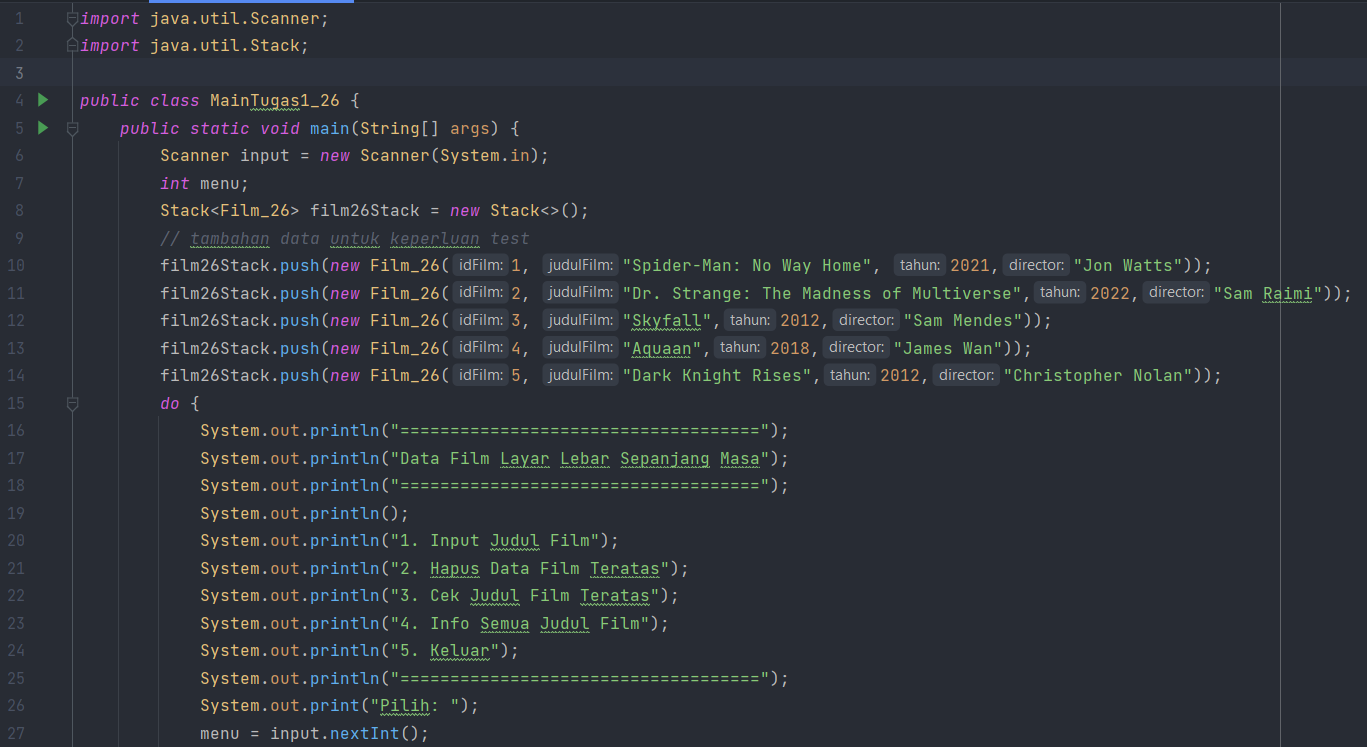


## Jawaban Nomor 1

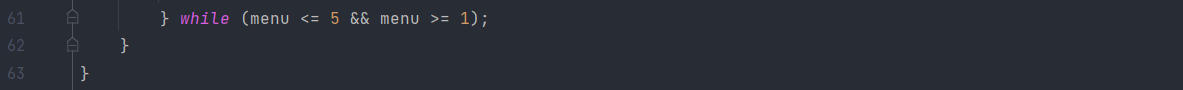
### Source code Class Film\_26



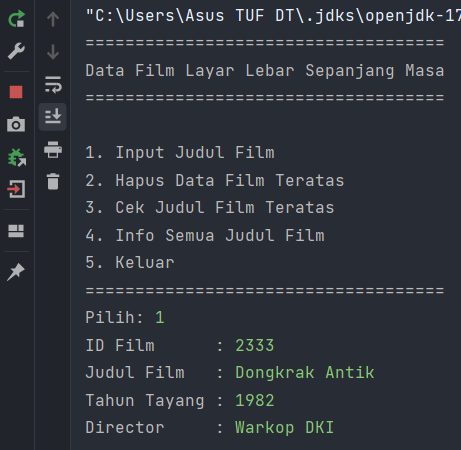
### Source code Class MainTugas1\_26



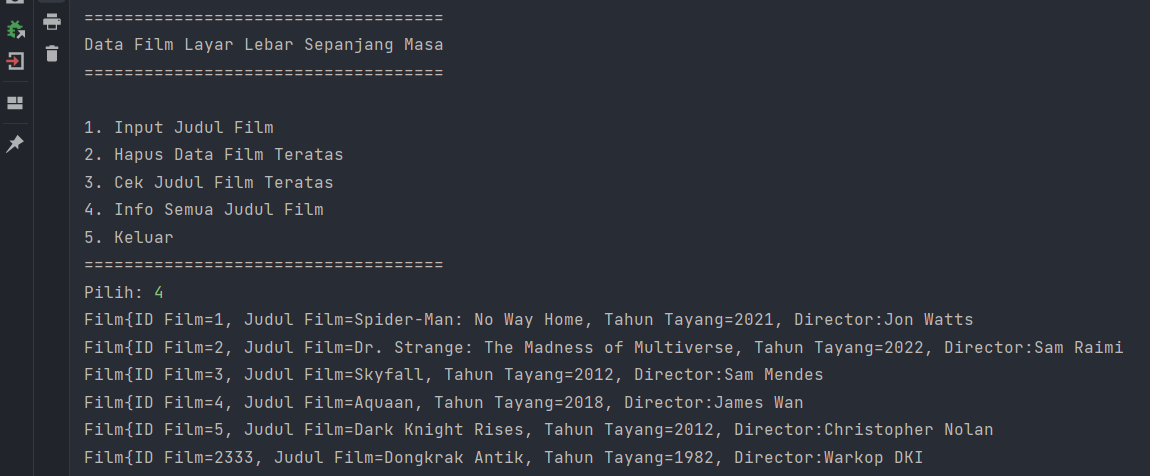




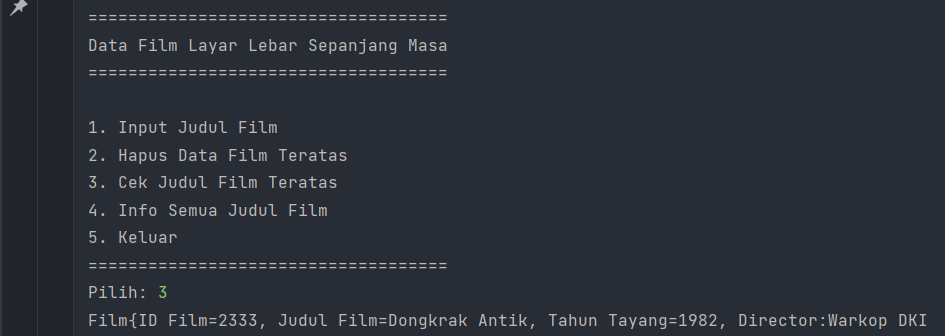
### Output Menu Awal dan Tambah Data



### Output Cetak Semua Data Film



### Output Judul Film di Tumpukan Teratas



### Output Hapus Film Teratas



## Nomor 2

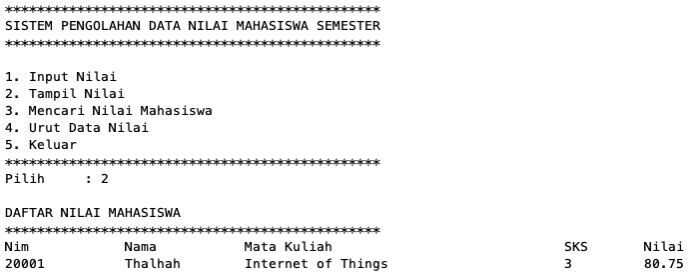
Buatlah implementasi program daftar nilai mahasiswa semester, minimal memiliki 3 class yaitu Mahasiswa, Nilai, dan Mata Kuliah. Khusus untuk data Mahasiswa dan Mata Kuliah harus sudah Page 7 of 18 diinisialisasi, sehingga ketika memasukkan data nilai data mahasiswa dan mata kuliah cukup menginputkan nim ataupun kode mata kuliah. Yang paling penting adalah antara objek mahasiswa, matakuliah, dan nilai harus saling terhubung.

Ilustrasi Program

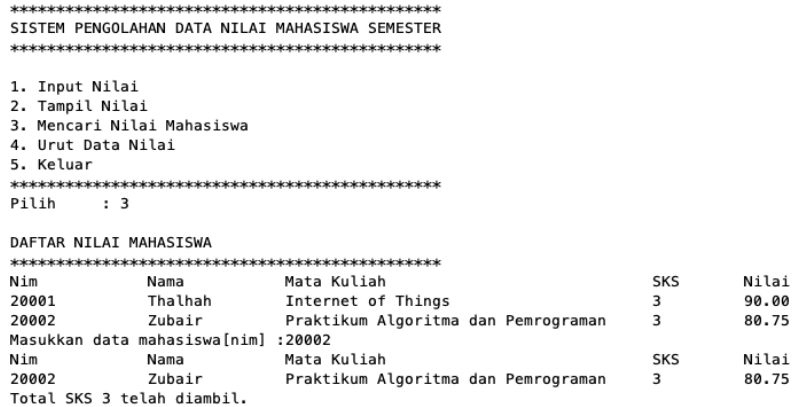
Menu Awal dan Penambahan Data



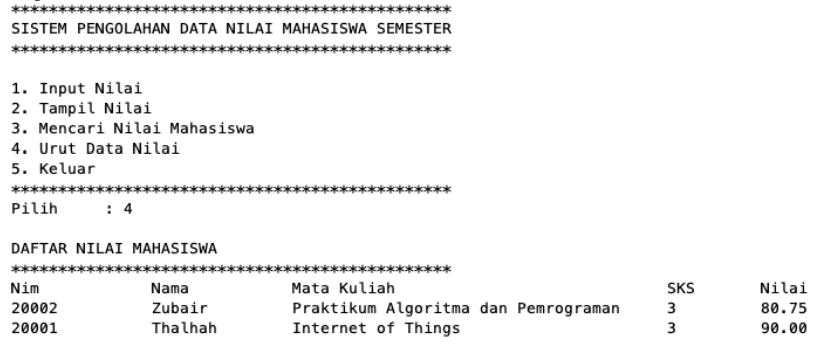
Tampil Nilai



Pencarian Data Mahasiswa

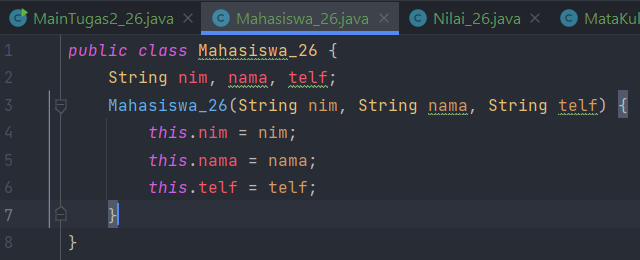


Pengurutan Data Nilai

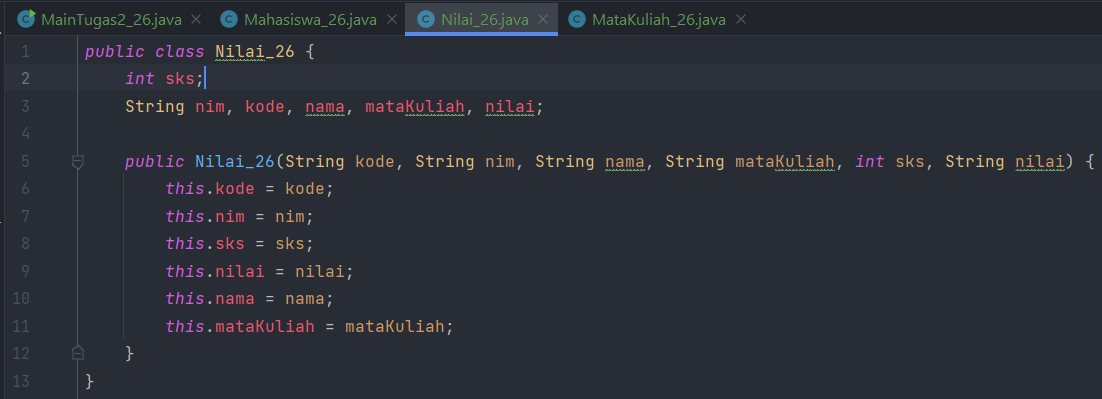


## Jawaban Nomor 2

### Source code Class Mahasiswa\_26



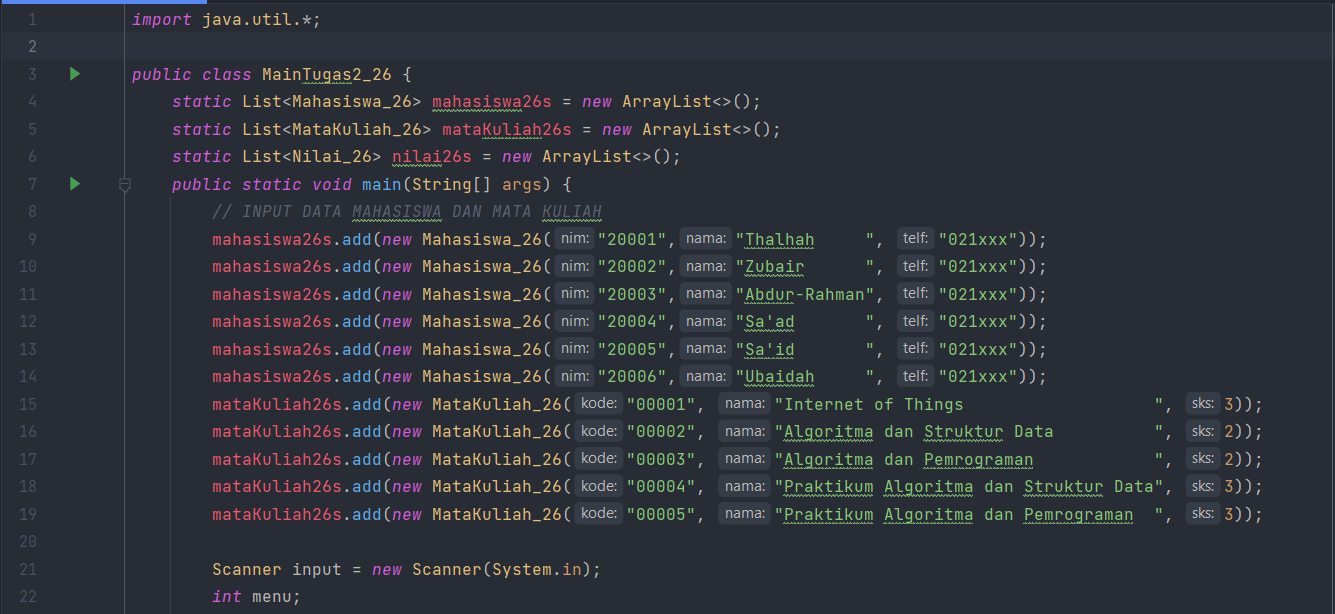
### Source code Class Nilai\_26

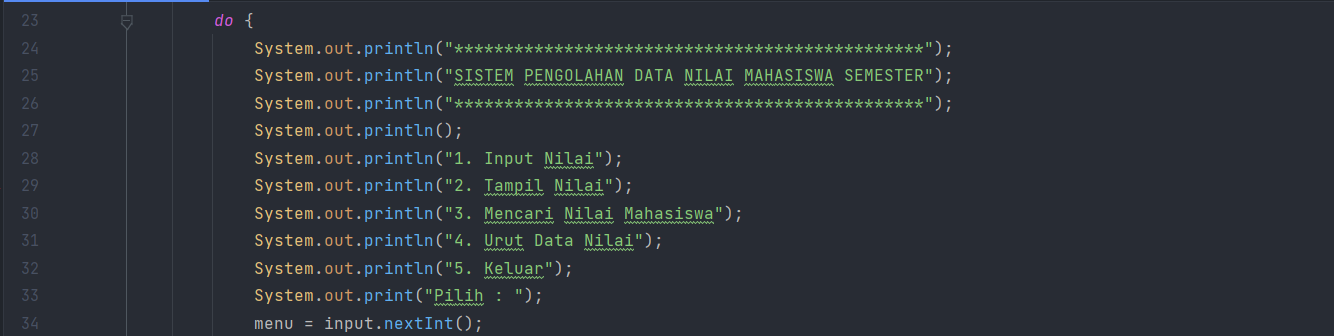


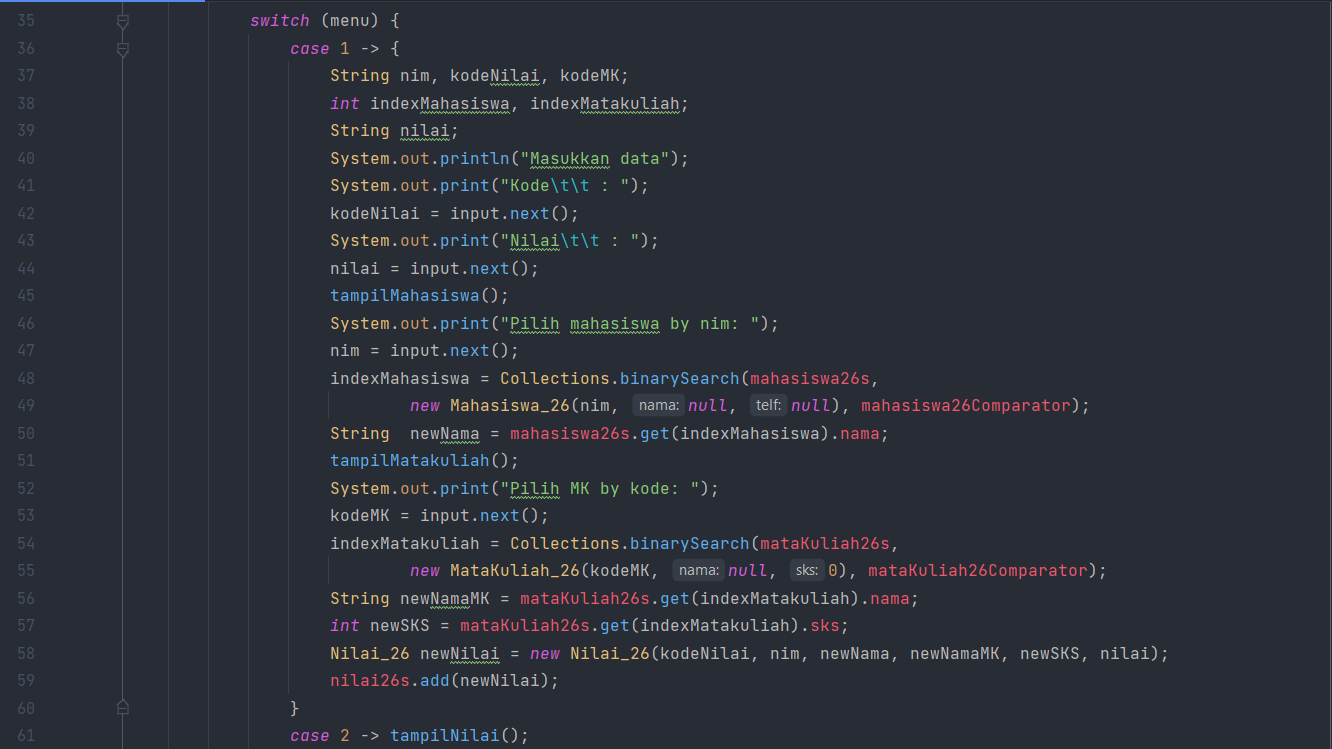
### Source code Class MataKuliah\_26

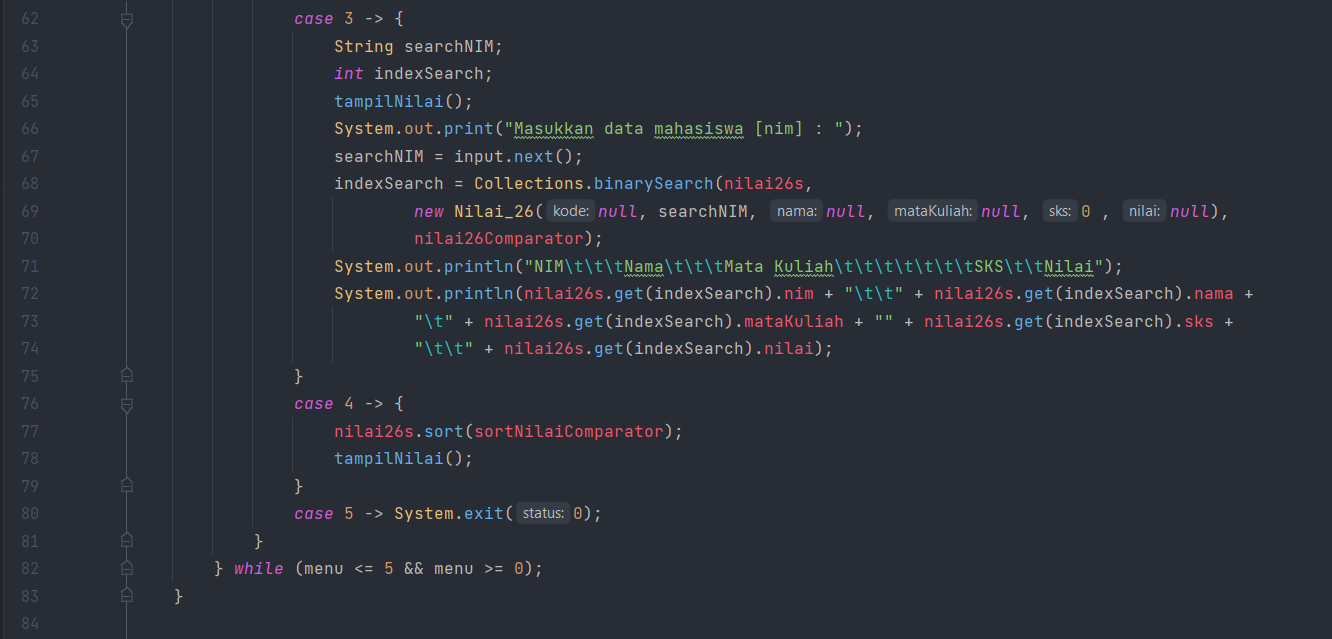


### Source code Class MainTugas2\_26

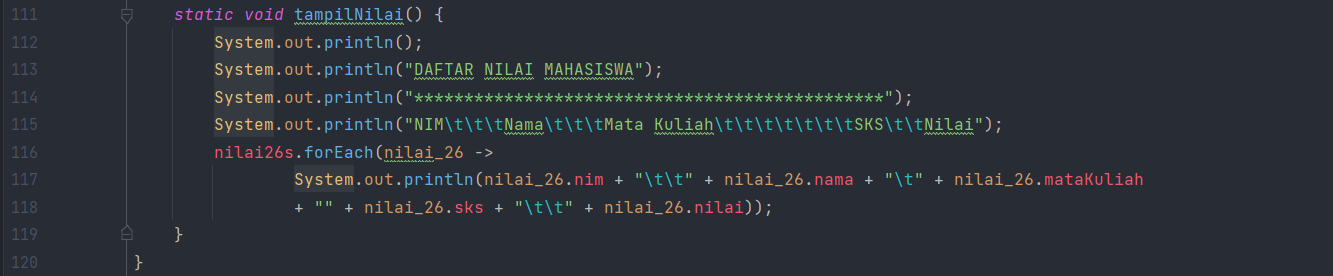




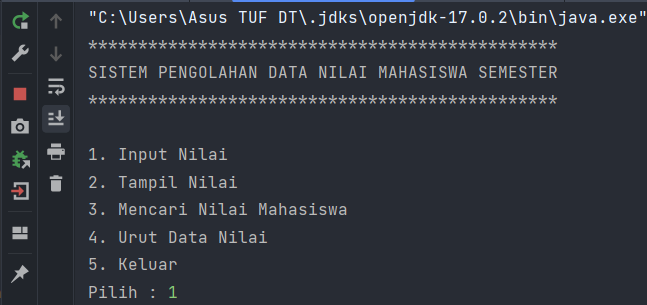


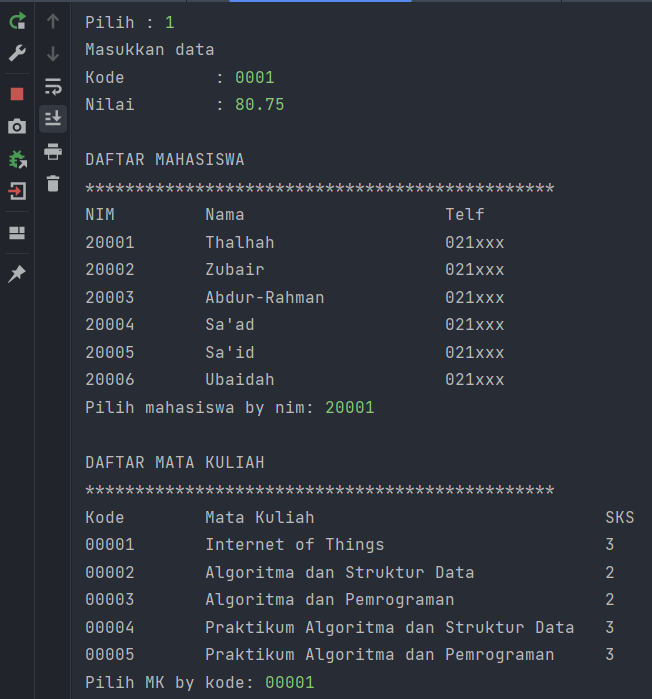




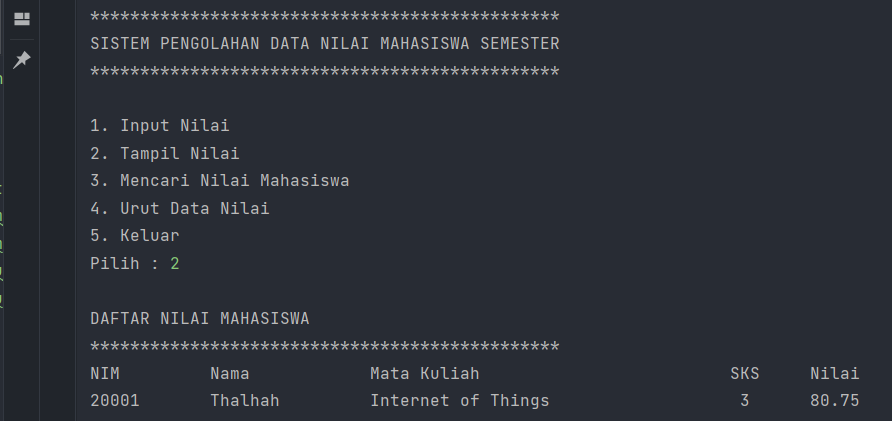


### Output Menu Awal dan Penambahan Data

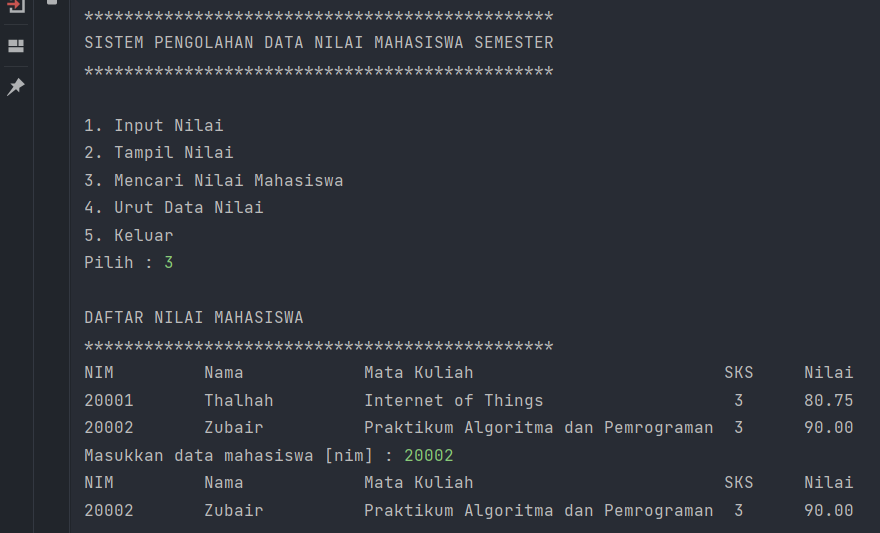




### Output Tampil Nilai



### Output Pencarian Data Mahasiswa



### Output Pengurutan Data Nilai

Sebelumnya sudah saya tambah data agar pengurutan berdasarkan nilai terkecil makin terlihat.

