JOBSHEET 16

Collection

Satria Wiguna/Ti 1D/ Absen 26

# Tujuan Praktikum

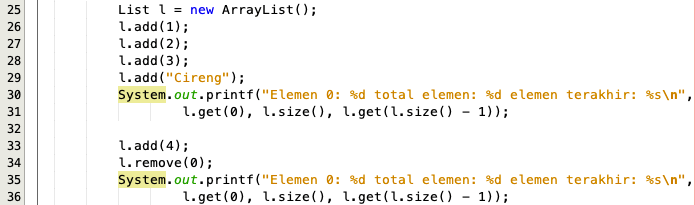
Setelah melakukan praktikum ini, mahasiswa mampu:

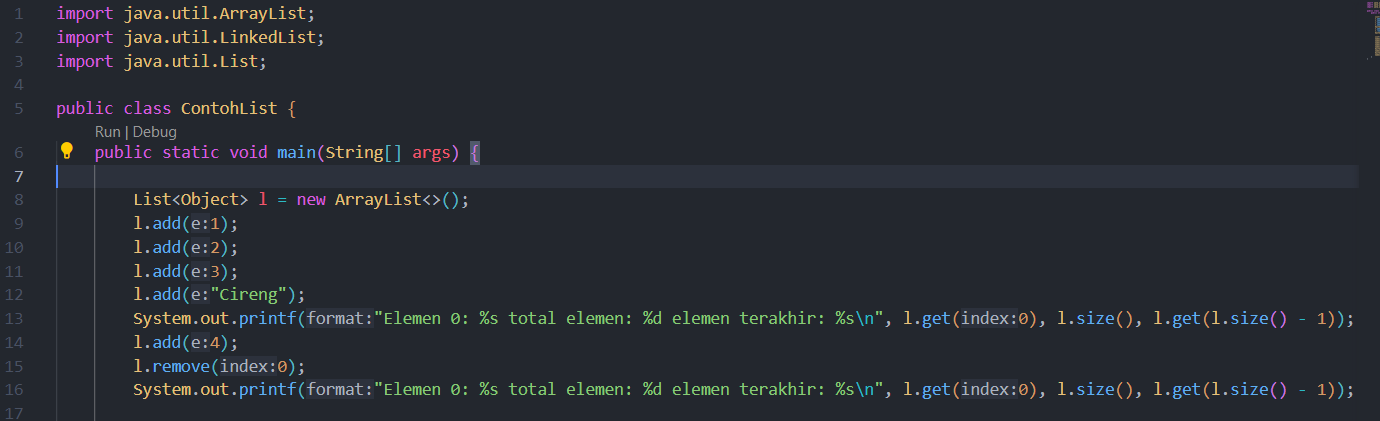
* + 1. memahami bentuk-bentuk collection dan hierarkinya;
    2. menerapkan collection sesuai dengan fungsi dan jenisnya;
    3. menyelesaikan kasus menggunakan collection yang sesuai.

# Kegiatan Praktikum 1

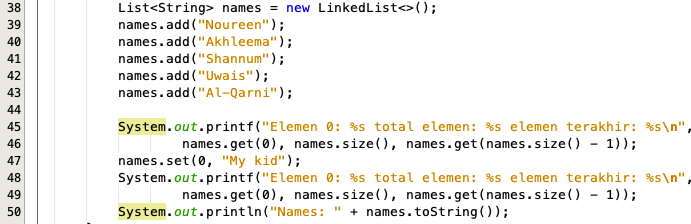
* + 1. **Percobaan 1**

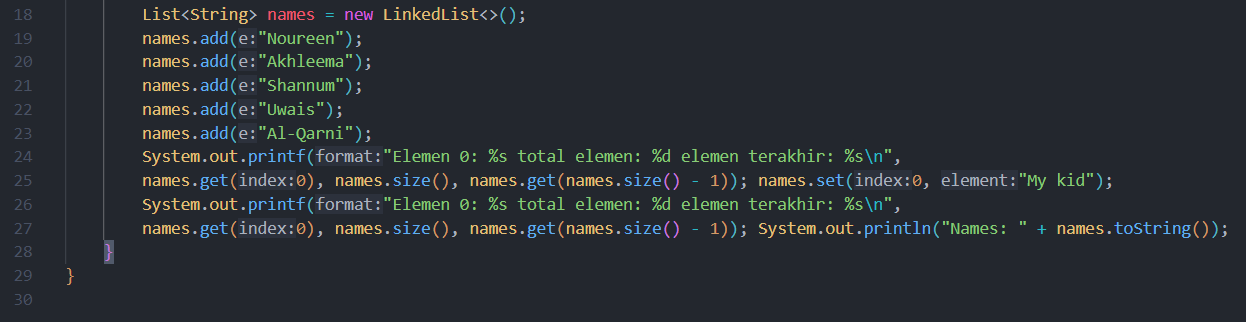
Pada percobaan 1 ini akan dicontohkan penggunaan collection untuk menambahkan sebuah elemen, mengakses elemen, dan menghapus sebuah elemen.

* + - 1. Buatlah sebuah class ContohList yang main methode berisi kode program seperti di bawah ini



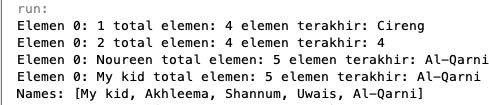
* + - 1. Tambahkan kode program untuk menggunakan collection dengan aturan penulisan kode program seperti berikut

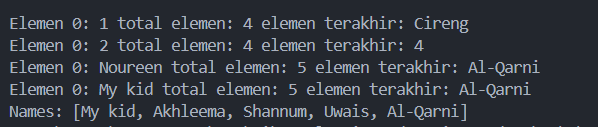




## Verifikasi Hasil Percobaan

Verifikasi hasil kompilasi kode program Anda dengan gambar berikut ini.



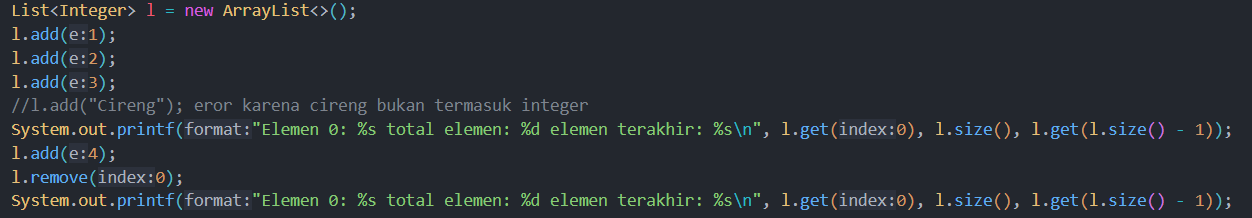


## Pertanyaan Percobaan

1. Perhatikan baris kode 25-36, mengapa semua jenis data bisa ditampung ke dalam sebuah Arraylist?

ArrayList dideklarasikan sebagai List<Object>, yang berarti dapat menampung objek dari tipe data apapun karena Object adalah superclass dari semua tipe data non-primitif di Java

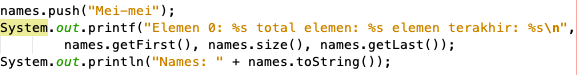
1. Modifikasi baris kode 25-36 seingga data yang ditampung hanya satu jenis atau spesifik tipe tertentu!

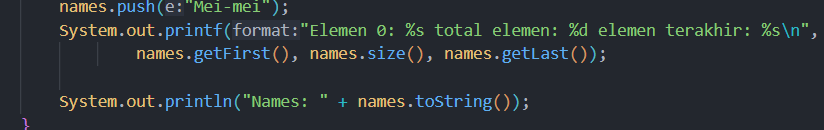


1. Ubah kode pada baris kode 38 menjadi seperti ini



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!

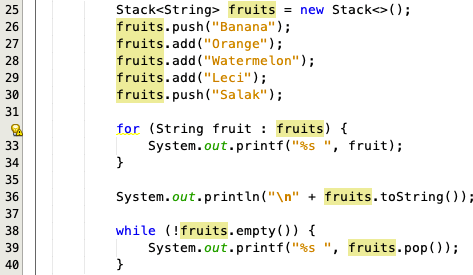
Kode di atas menunjukkan penggunaan ArrayList untuk menyimpan bilangan bulat dan LinkedList untuk menyimpan nama-nama. Menggunakan List<Integer> memastikan hanya bilangan bulat yang diterima, sementara LinkedList<String> memungkinkan penggunaan metode tambahan seperti push() untuk menambahkan elemen baru di depan. Outputnya mencakup cetakan nilai elemen pertama, jumlah elemen, dan elemen terakhir dari setiap struktur data.

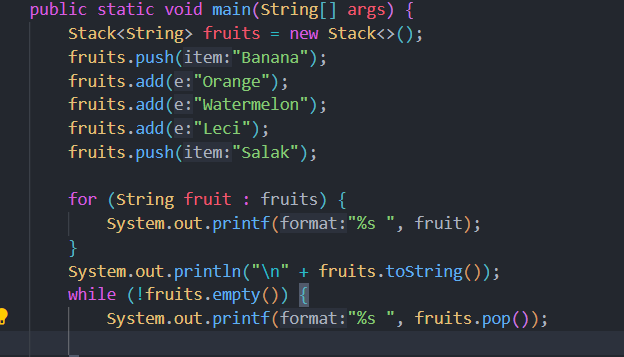
# Kegiatan Praktikum 2

## Tahapan Percobaan

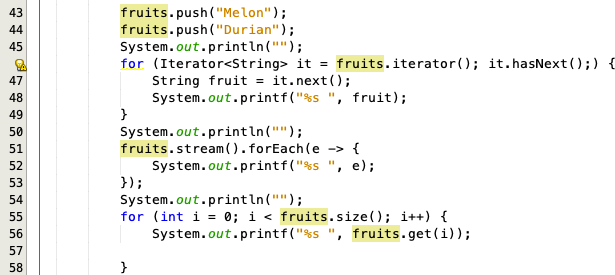
Pada praktikum 2 ini akan dibuat beberapa method untuk menampilkan beberapa cara yang dapat dilakukan untuk mengambil/menampilkan elemen pada sebuah collection. Silakan ikutilah Langkah-langkah di bawah ini

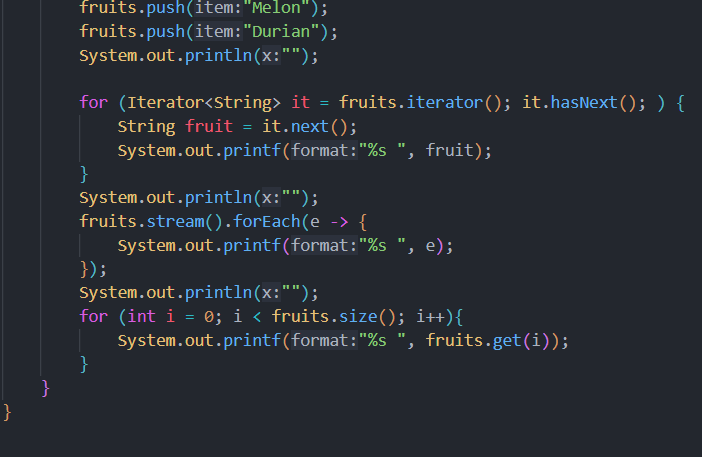
* + - 1. Buatlah class dengan nama LoopCollection serta tambahkan method main yang isinya adalah sebagai berikut**.**





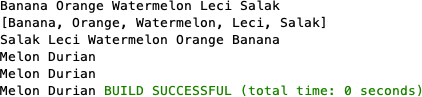
* + - 1. Tambahkan potongan kode berikut ini dari yang sebelumnya agar proses menampilkan elemen pada sebuah stack bervariasi.

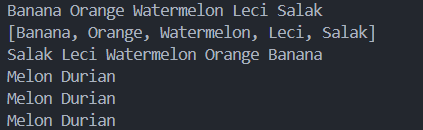




## Verifikasi Hasil Percobaan

Verifikasi hasil kompilasi kode program Anda dengan gambar berikut ini.





## Pertanyaan Percobaan

* + - 1. Apakah perbedaan fungsi push() dan add() pada objek *fruits*?

push() adalah metode dari kelas Stack yang digunakan untuk menambahkan elemen ke atas stack.

add() adalah metode dari antarmuka Collection (yang diimplementasikan oleh Stack) yang digunakan untuk menambahkan elemen ke dalam koleksi

* + - 1. Silakan hilangkan baris 43 dan 44, apakah yang akan terjadi? Mengapa bisa demikian?

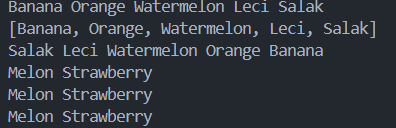
Jika baris 43 (System.out.println("")) dan baris 44 (for (Iterator<String> it = fruits.iterator(); it.hasNext(); ) {) dihapus, maka output program tidak akan mencetak elemen-elemen dari fruits setelah di-push kembali dengan "Melon" dan "Durian". Ini terjadi karena bagian kode yang mengiterasi dan mencetak elemen setelah push akan hilang.

* + - 1. Jelaskan fungsi dari baris 46-49?
  + Baris 46-49 menggunakan iterasi dengan Iterator untuk mencetak setiap elemen yang tersisa dalam fruits setelah elemen-elemen sebelumnya di-pop.
  + fruits.iterator() mengambil iterator dari stack fruits.
  + it.hasNext() memeriksa apakah masih ada elemen yang tersisa dalam stack.
  + it.next() mengambil dan mencetak elemen saat ini dari iterator.
    - 1. Silakan ganti baris kode 25, *Stack<String>* menjadi *List<String>* dan apakah yang terjadi? Mengapa bisa demikian?

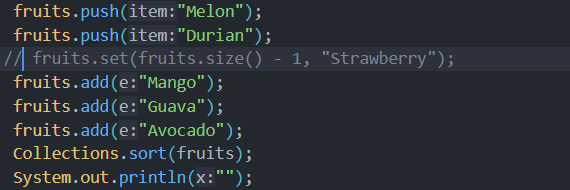
atidak akan bisa langsung menggunakan metode push() karena List tidak memiliki metode push(). Sebagai gantinya, Anda akan menggunakan metode add() untuk menambahkan elemen ke dalam List

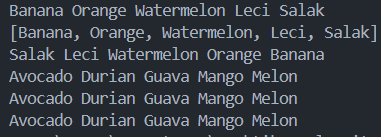
* + - 1. Ganti elemen terakhir dari dari objek fruits menjadi “Strawberry”!





* + - 1. Tambahkan 3 buah seperti “Mango”,”guava”, dan “avocado” kemudian dilakukan sorting!



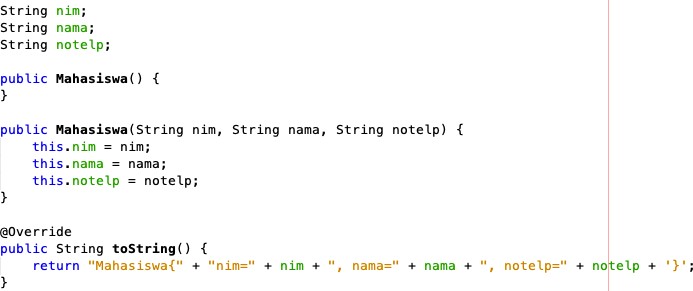


# Kegiatan Praktikum 3

## Tahapan Percobaan

Pada praktikum 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.

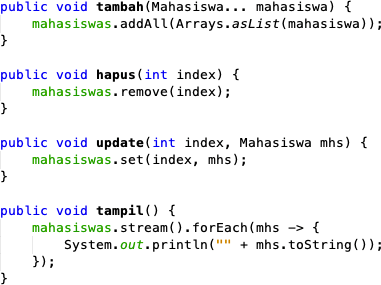
* + - 1. Buatlah sebuah class Mahasiswa dengan attribute, kontruktor, dan fungsi sebagai berikut.





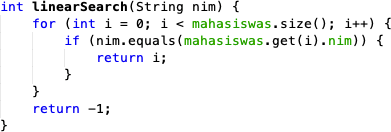
* + - 1. Selanjutnya, buatlah sebuah class ListMahasiswa yang memiliki attribute seperti di bawah ini

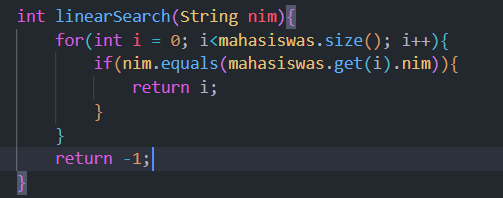


* + - 1. Method **tambah(), hapus(), update(), dan tampil()** secara berurut dibuat agar bisa melakukan operasi-operasi seperti yang telah disebutkan.

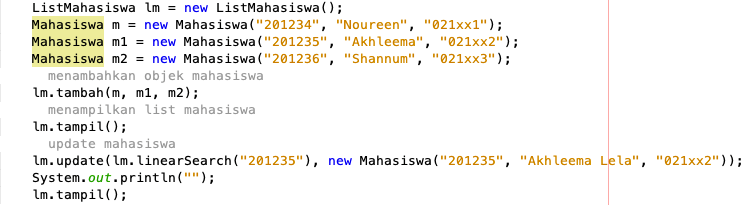


* + - 1. Untuk proses hapus, update membutuhkan fungsi pencarian terlebih dahulu yang potongan kode programnya adalah sebagai berikut





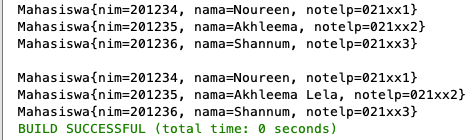
* + - 1. Pada class yang sama, tambahkan main method seperti potongan program berikut dan amati hasilnya!

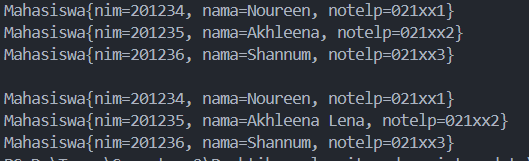




## Verifikasi Hasil Percobaan

Verifikasi hasil kompilasi kode program Anda dengan gambar berikut ini.



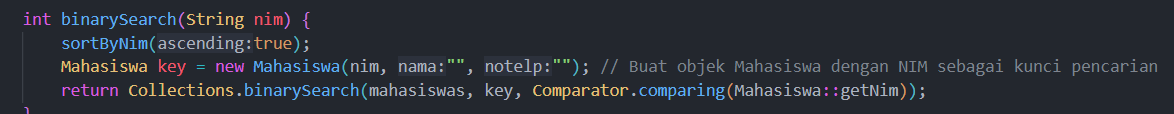


## Pertanyaan Percobaan

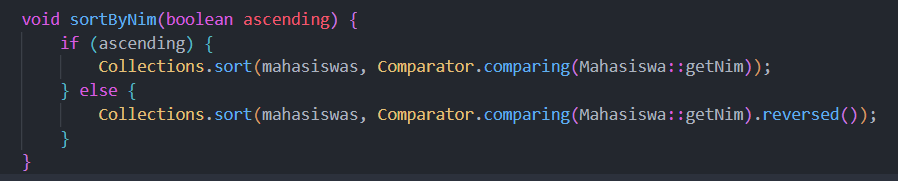
* + - 1. Pada fungsi tambah() yang menggunakan unlimited argument itu menggunakan konsep apa? Dan kelebihannya apa?

Konsep pada tambah() dengan unlimited argument: Menggunakan konsep varargs. Kelebihannya adalah memberikan fleksibilitas dalam menerima jumlah argumen yang bervariasi saat memanggil metode, tanpa perlu mendefinisikan array secara eksplisit.

* + - 1. Pada fungsi linearSearch() di atas, silakan diganti dengan fungsi binarySearch() dari collection!



* + - 1. Tambahkan fungsi sorting baik secara ascending ataupun descending pada class tersebut!

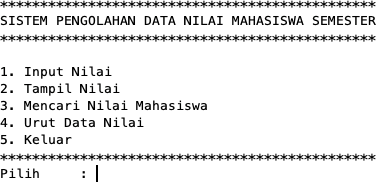


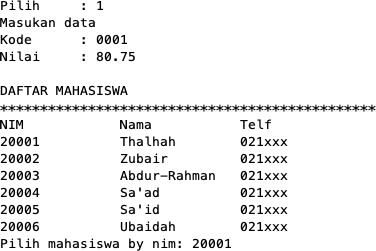
# Tugas Praktikum

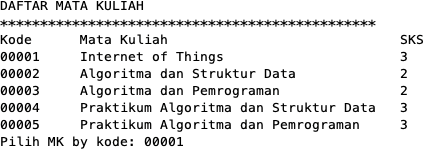
* + 1. Buatlah implementasi program daftar nilai mahasiswa semester, minimal memiliki 3 class yaitu Mahasiswa, Nilai, dan Mata Kuliah. Data Mahasiswa dan Mata Kuliah perlu melalui penginputan data terlebih dahulu.

## Ilustrasi Program

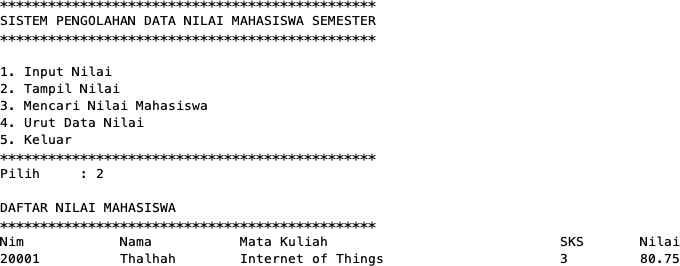
*Menu Awal dan Penambahan Data*

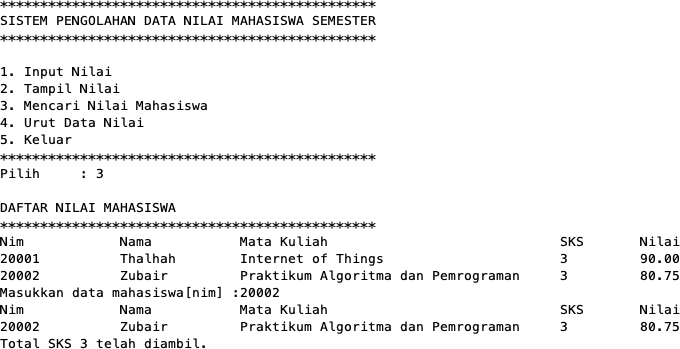


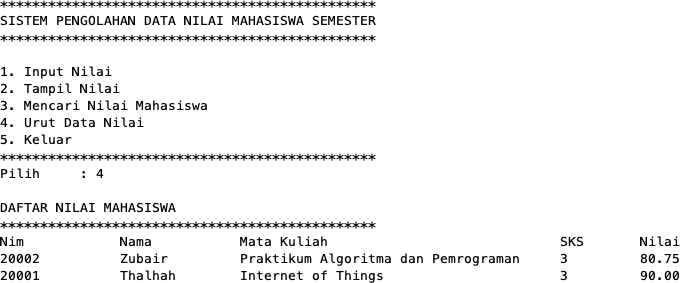




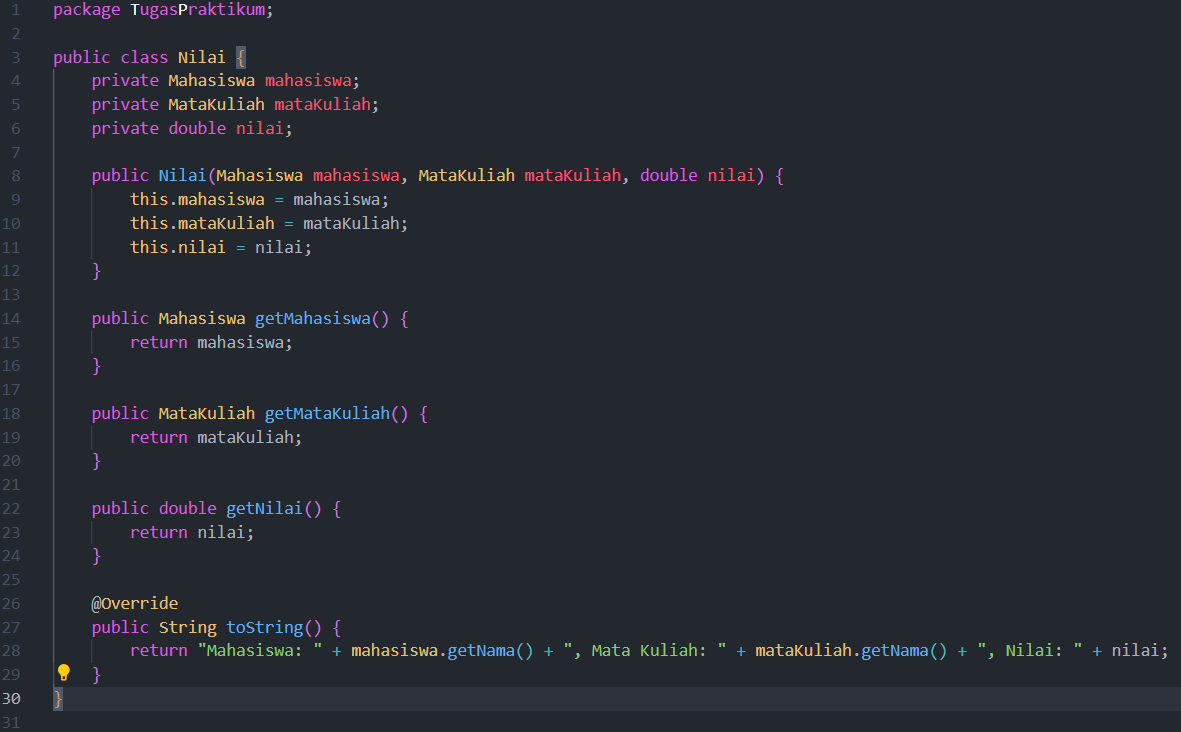
*Tampil Nilai*



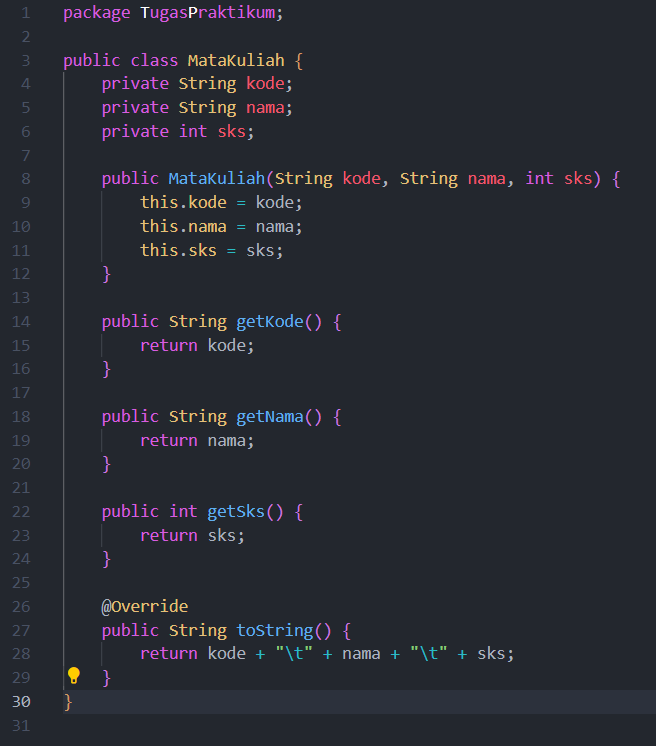
*Pencarian Data Mahasiswa*

*Pengurutan Data Nilai*

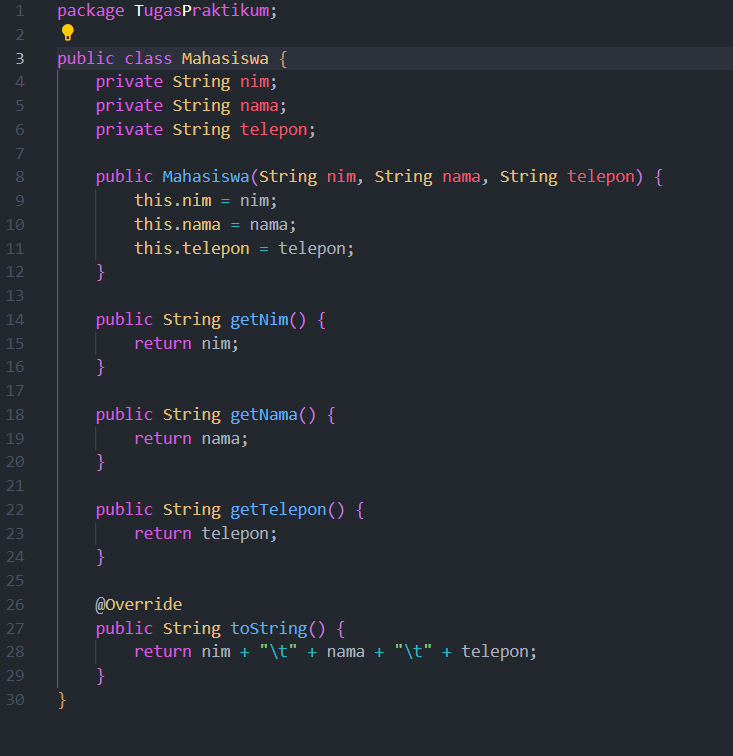
* + 1. Tambahkan prosedur hapus data mahasiswa melalui implementasi Queue pada collections Tugas nomor 1!

Nilai: 

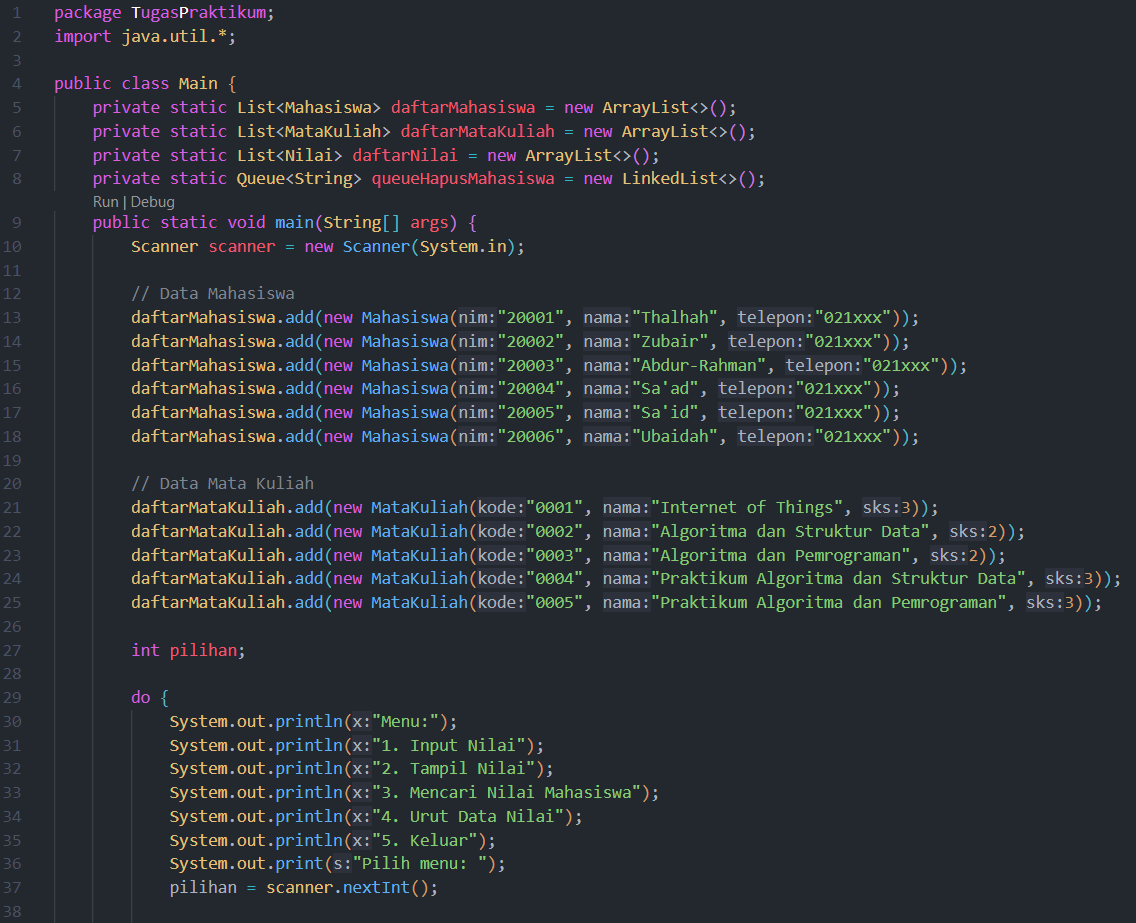
Mata kuliah:

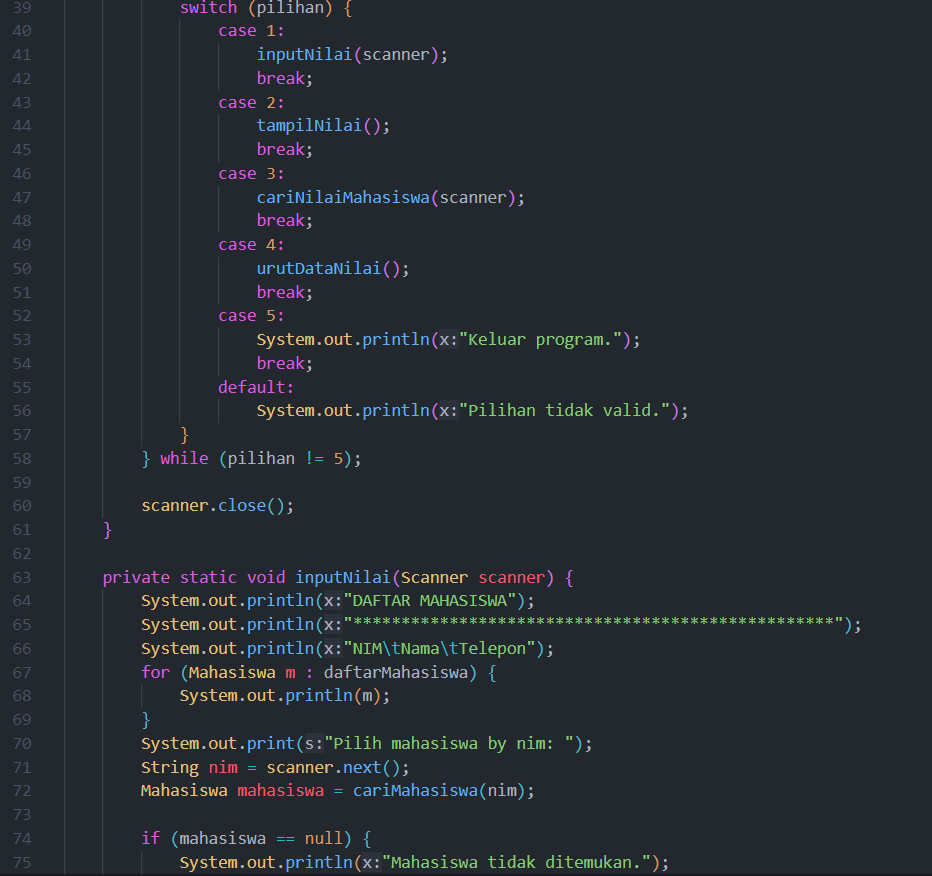


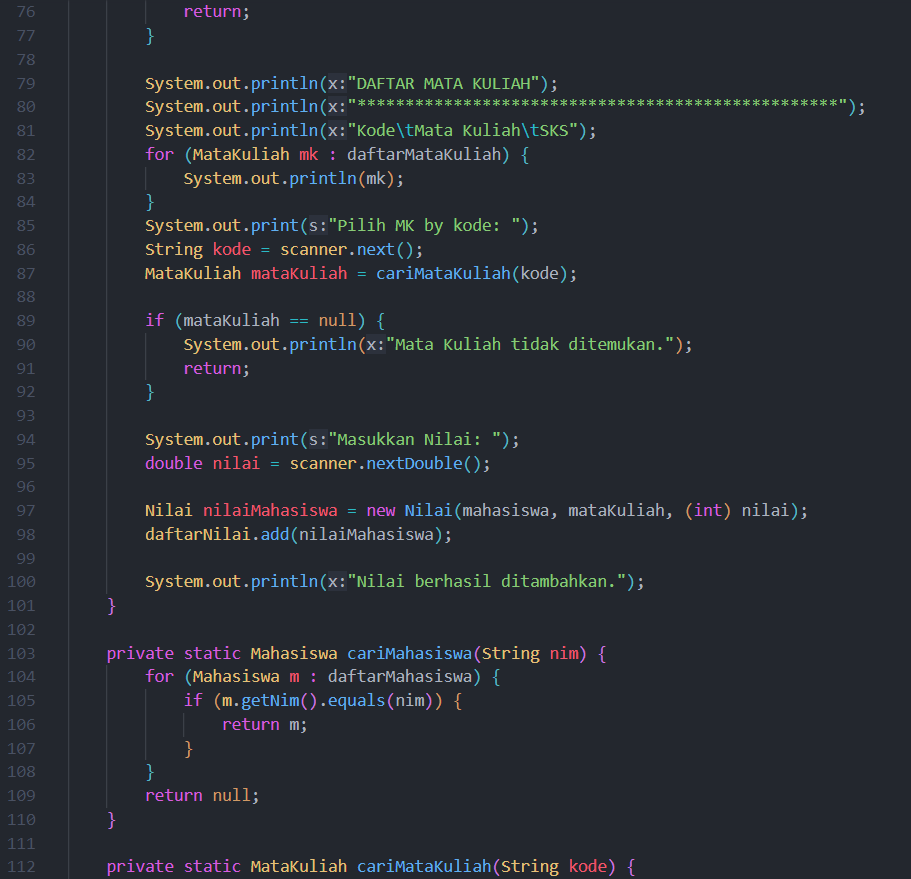
Mahasiswa:

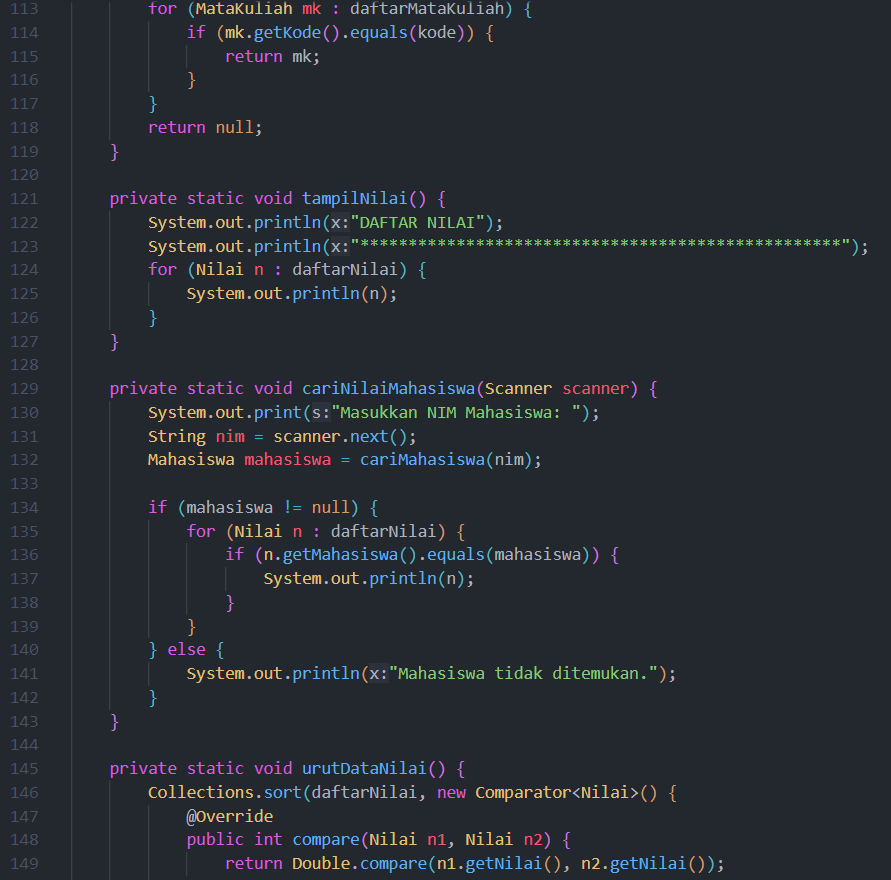


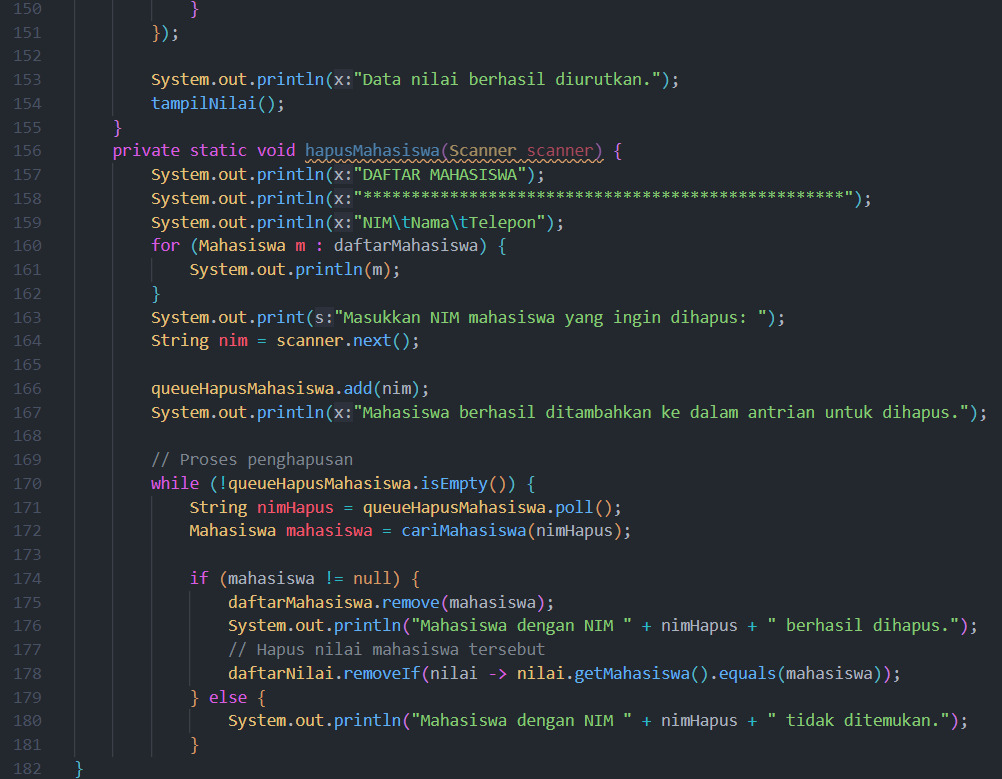
Main:













--- \*\*\* ---