  
 Name : Azaria Cindy Sahasika

Number Id : 2341760169 / 06

Class : 1G – Business Information System

Lesson : Algorithm and Data Structure

Material : Material 13 – Collections

Github Link : <https://github.com/azariacindy/algorithm-ds>

JOBSHEET 16

Collection

1. 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.
4. Kegiatan Praktikum 1
5. 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

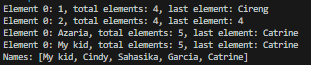
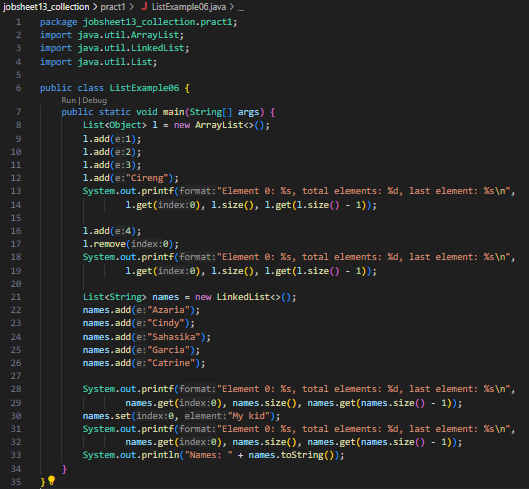
Graphical user interface, text, application, email

Description automatically generated

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

Graphical user interface, text

Description automatically generated



1. Verifikasi Hasil Percobaan

Verifikasi hasil kompilasi kode program Anda dengan gambar berikut ini.

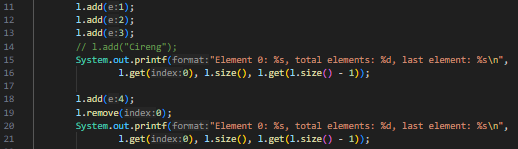
Text

Description automatically generated

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

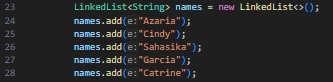
* Karena ArrayList tidak memmiliki kurung sudut yang menentukan data apa yang akan disimpan, ArrayList pada baris code 25-36 diatas digunakan debagai sintak umum yang dapat menyimpan data dari semua jenis data.

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

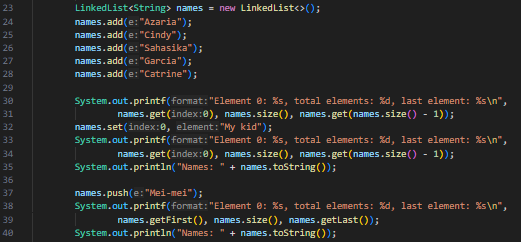




1. Tambahkan juga baris berikut ini, untuk memberikan perbedaan dari tampilan yang sebelumnya

Text

Description automatically generated with medium confidence



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

* Dengan menggunakan class LinkedList, kita dapat menggunakan metode daftar terkait seperti push(), getFirst(), dan getLast(), tetapi ketika menggunakan class List tidak dapat melakukannya.

1. Kegiatan Praktikum 2
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**.**

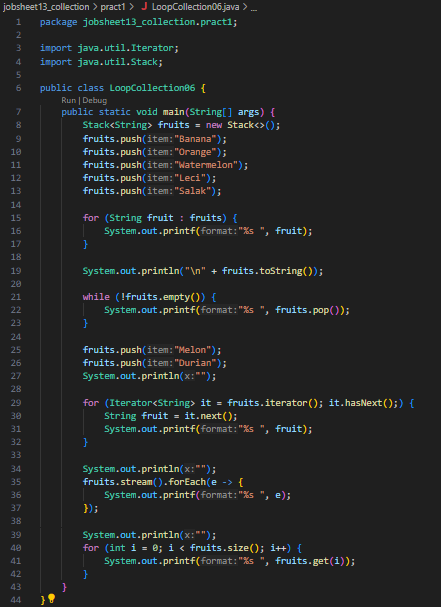
Text

Description automatically generated

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

Text

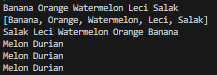
Description automatically generated



1. Verifikasi Hasil Percobaan

Verifikasi hasil kompilasi kode program Anda dengan gambar berikut ini.

A picture containing text, indoor, screenshot

Description automatically generated

1. Pertanyaan Percobaan
2. Apakah perbedaan fungsi push() dan add() pada objek *fruits*?

* Push(): fungsi dari interface stack
* Add(): bukan fungsi dari interface stack

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



* Jika tidak push "Melon" dan "Durian" kembali ke dalam stack, stack akan tetap kosong setelah perulangan while, sehingga tidak ada keluaran lebih lanjut dari perulangan berikutnya dan pemrosesan stream. Hal ini terjadi karena bagian kode tersebut mengulang di atas tumpukan, yang pada saat itu, semua elemennya telah dihapus.

1. Jelaskan fungsi dari baris 46-49?



* For-each loop: untuk menginisialisasi dan mengatur perulangan untuk iterasi melalui setiap elemen dalam stack fruits. ‘fruit’ untuk mencetak setiap elemen dalam stack.



* While loop: untuk menginisialisadi while loop yang akan terus berjalan selama stack fruits tidak kosong. ‘fruits.pop()’ untuk mengeluarkan dan mencetak setiap elemen dari stack hingga stack tidak kosong.

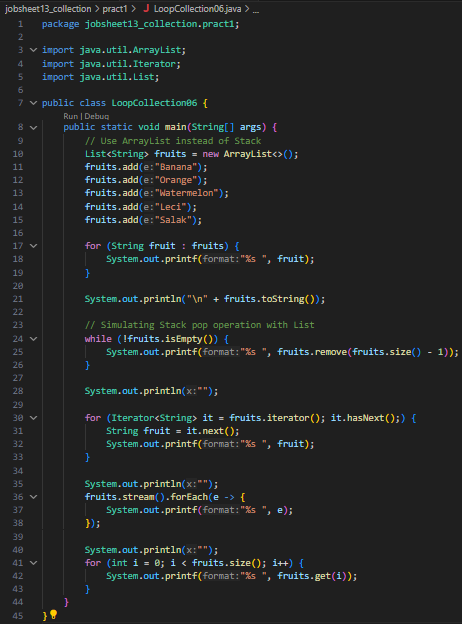


* Iterator loop: ‘Iterator<String> it’ untuk mengatur perulangan menggunakan iterator pada stack fruits. ‘it.next()’ untuk mengatur variabel ‘fruit’ dengan elemen berikutnya dalam iterasi.



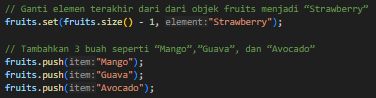
* Stream forEach: menginisialisasi perulangan menggunakan stream dan method ‘forEach’ untek setiap elemen ‘e’ dalam stack ‘fruits’.

1. Silakan ganti baris kode 25, *Stack<String>* menjadi *List<String>* dan apakah yang terjadi? Mengapa bisa demikian?

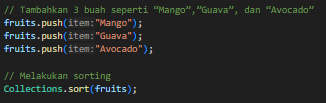
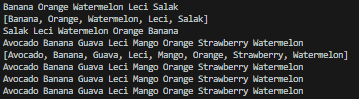


* push untuk Stack, sedangkan metode add digunakan untuk List
* pop untuk Stack, sedangkan metode hapus dapat digunakan untuk List

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



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

1. Kegiatan Praktikum 3
2. 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.

**Graphical user interface, text, application

Description automatically generated**

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.

Graphical user interface, text, application

Description automatically generated

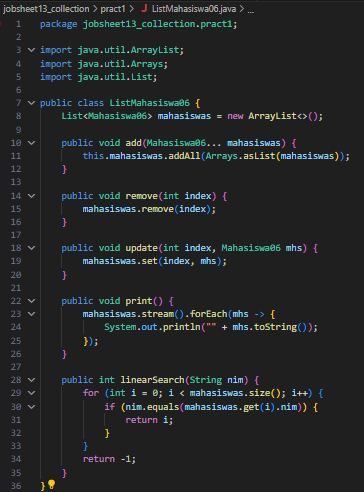
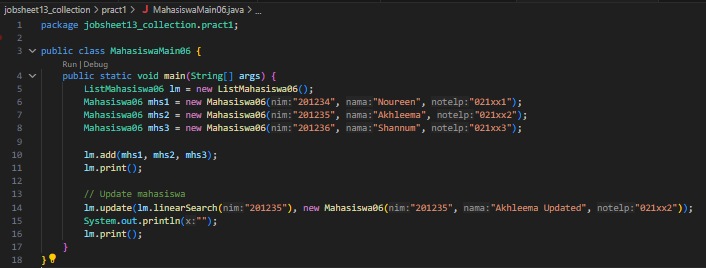
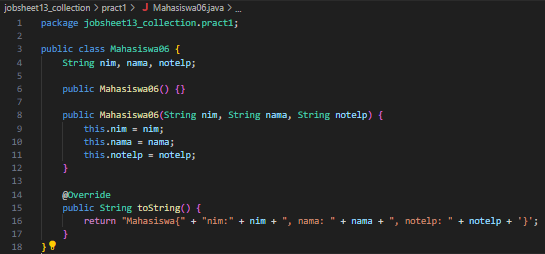
1. Untuk proses hapus, update membutuhkan fungsi pencarian terlebih dahulu yang potongan kode programnya adalah sebagai berikut

Graphical user interface, text, application

Description automatically generated

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

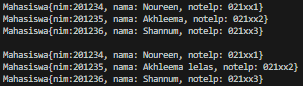
Graphical user interface, text

Description automatically generated with medium confidence

1. Verifikasi Hasil Percobaan

Verifikasi hasil kompilasi kode program Anda dengan gambar berikut ini.

A picture containing text, indoor

Description automatically generated 

1. Pertanyaan Percobaan
2. Pada fungsi tambah() yang menggunakan unlimited argument itu menggunakan konsep apa? Dan kelebihannya apa?

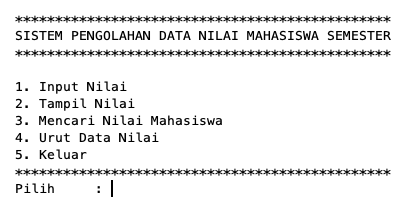
* **Fungsi** add() **dengan menggunakan varargs (**Mahasiswa06... mahasiswas**)** menggunakan konsep **varargs** (variable-length arguments). Kelebihannya adalah memungkinkan kita untuk memasukkan jumlah argumen yang variatif saat memanggil metode add(), tanpa harus membuat array terlebih dahulu.

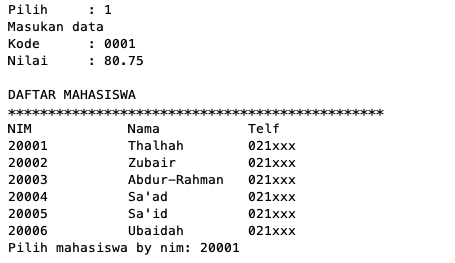
1. Pada fungsi linearSearch() di atas, silakan diganti dengan fungsi binarySearch() dari collection!
2. Tambahkan fungsi sorting baik secara ascending ataupun descending pada class tersebut!

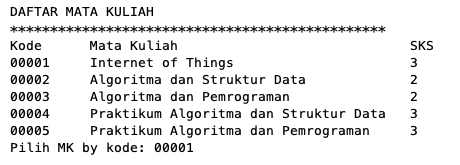
1. Tugas Praktikum
2. Buatlah implementasi program daftar nilai mahasiswa semester, minimal memiliki 3 class yaitu Mahasiswa, Nilai, dan Mata Kuliah. Data Mahasiswa dan Mata Kuliah perlu melalui penginputan data terlebih dahulu.

**Ilustrasi Program**

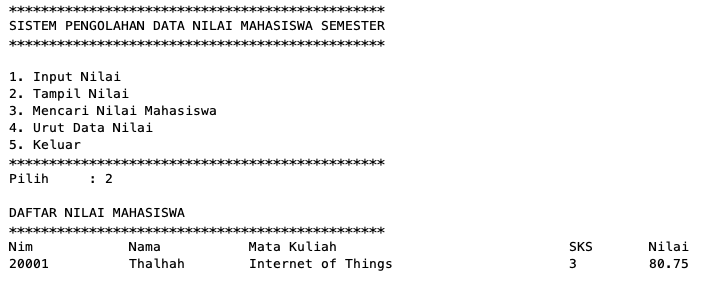
*Menu Awal dan Penambahan Data*



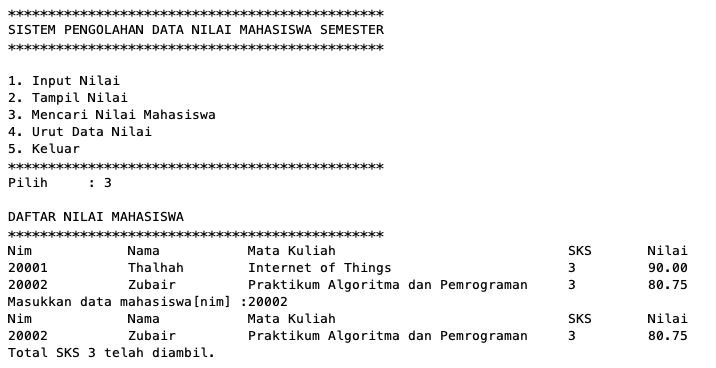




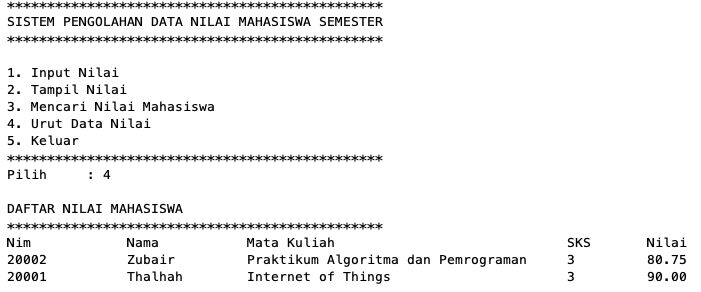
*Tampil Nilai*



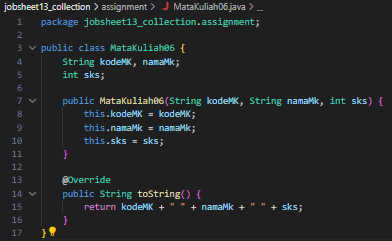
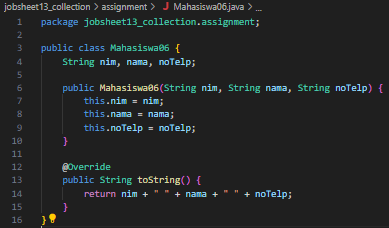
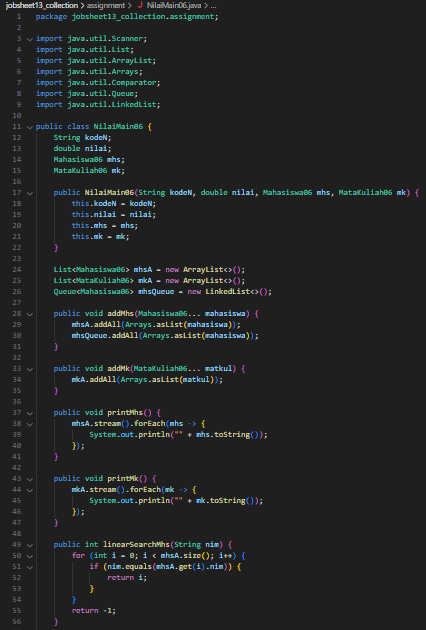
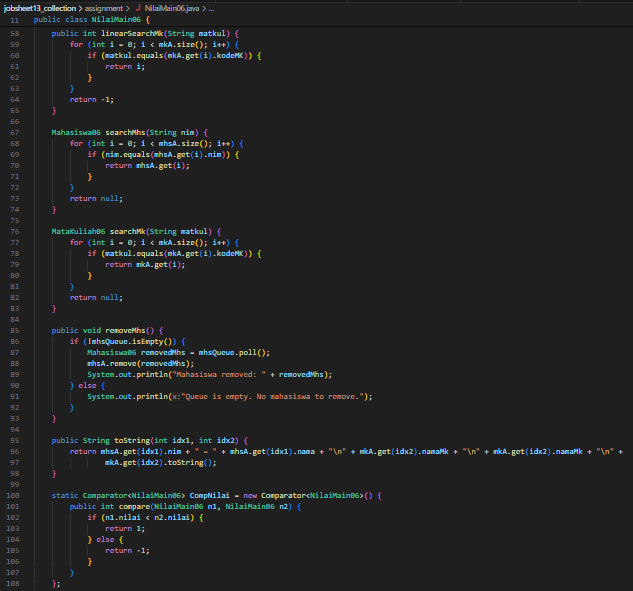
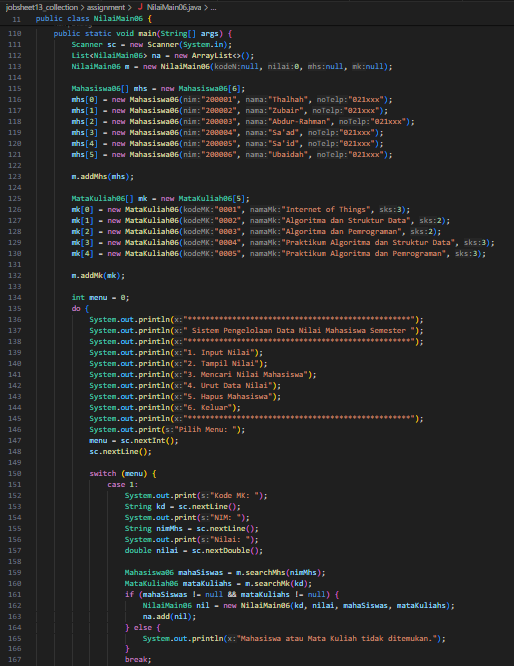
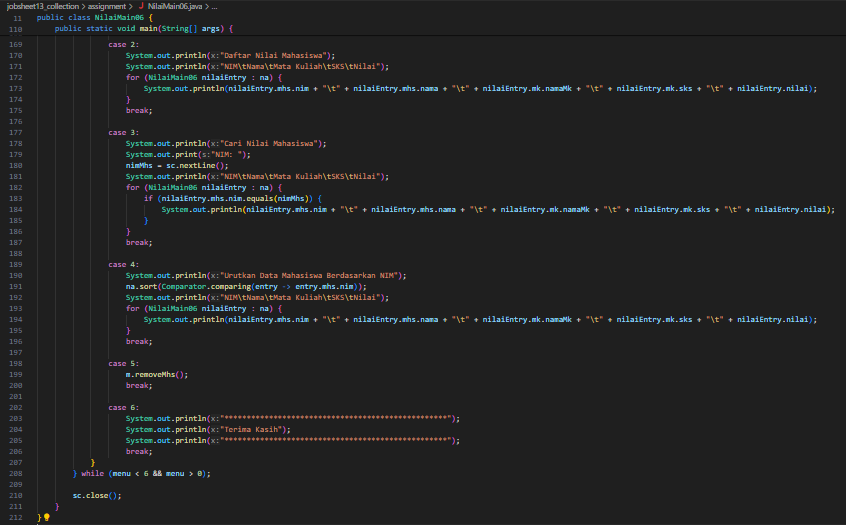
*Pencarian Data Mahasiswa*



*Pengurutan Data Nilai*

**

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

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