**LAPORAN PRAKTIKUM**

**MATA KULIAH ALGORITMA DAN STRUKTUR DATA**

Dosen Pengampu : Triana Fatmawati, S.T, M.T

**PERTEMUAN - 15 - Collection**

****

**Nama : M. Zidna Billah Faza**

**NIM : 2341760030**

**Prodi : D-IV Sistem Informasi Bisnis**

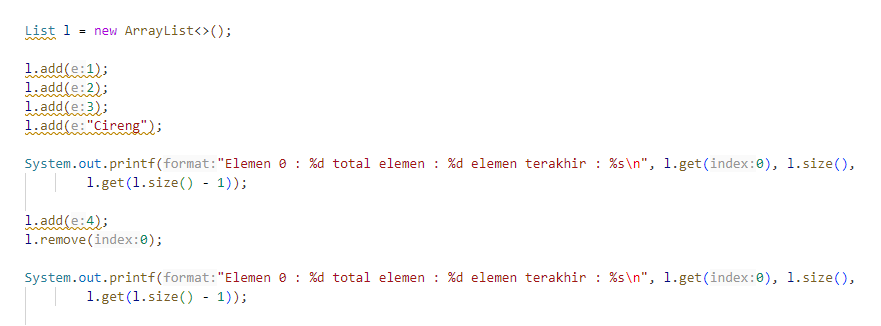
**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

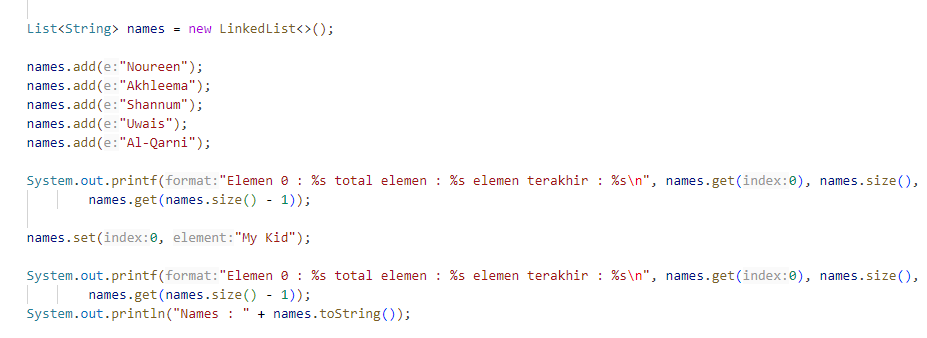
**2024**

**Percobaan 1**

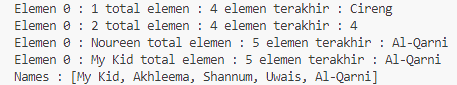
1. Buatlah sebuah class ContohList yang main methode berisi kode program seperti di bawah ini



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



1. Verifikasi Hasil Percobaan

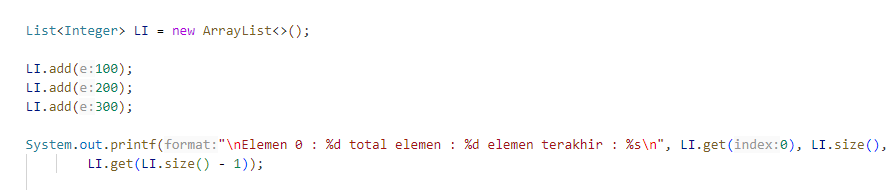


**Pertanyaan Percobaan 1**

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

Karena pada deklarasi ArrayList() tidak disebutkan parameter yang spesifik maka ketika mengisi ArrayList() dapat berupa banyak tipe data

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

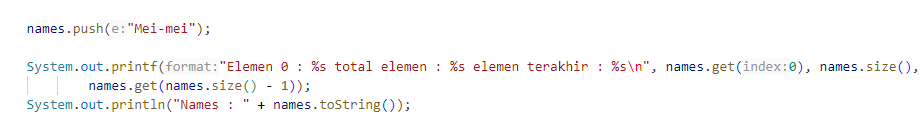




1. Ubah kode pada baris kode 38 menjadi seperti ini



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



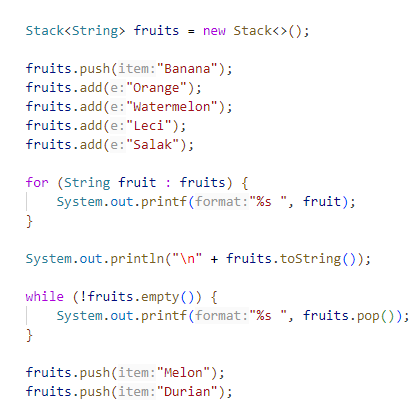
1. Dari penambahan kode tersebut, silakan dijalankan dan apakah yang dapat Anda jelaskan!

Baris kode 38 telah diubah untuk menggunakan LinkedList<String> untuk objek names. Kemudian, baris kode 39 ditambahkan untuk memasukkan string "Mei-mei" ke dalam LinkedList menggunakan metode push().

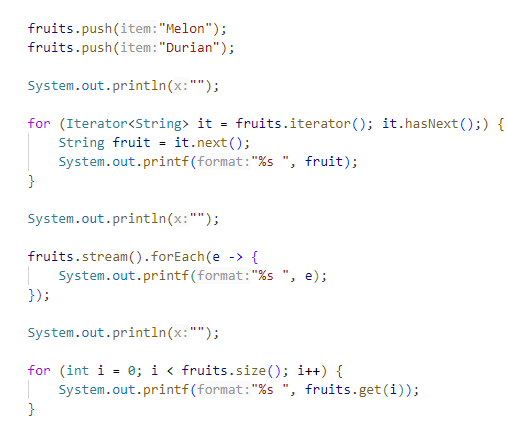
Selanjutnya, pada baris kode 40, printf() dipanggil untuk mencetak elemen pertama (getFirst()), ukuran total (size()), dan elemen terakhir (getLast()) dari LinkedList names. Dan akhirnya, pada baris kode 41, toString() dipanggil untuk mencetak LinkedList names.

**Percobaan 2**

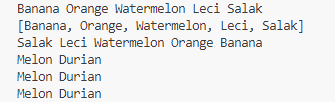
1. Buatlah class dengan nama LoopCollection serta tambahkan method main yang isinya adalah sebagai berikut



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



1. Verifikasi Hasil Percobaan



**Pertanyaan Percobaan 2**

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

add() lebih cocok digunakan untuk menambahkan elemen kedalam *stack* dengan konsep LIFO, sedangkan push() lebih cocok digunakan untuk menambahkan elemen kedalam *list* dengan tidak memiliki konsep LIFO

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

Program tidak mencetak apapun karena tidak ada elemen yang ada di dalam Stack karena tidak ada elemen di dalam Stack

1. Jelaskan fungsi dari baris 46-49?

Kode tersebut berfungsi untuk mencetak semua elemen didalam stack dengan mengggunakan iterator, maka ketika iterator menemukan elemen didalma stack maka akan di print

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

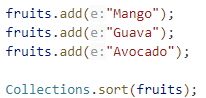
Maka akan terjadi error pada push() karena pada List tidak memiliki push() maka dapat diganti dengan add()

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





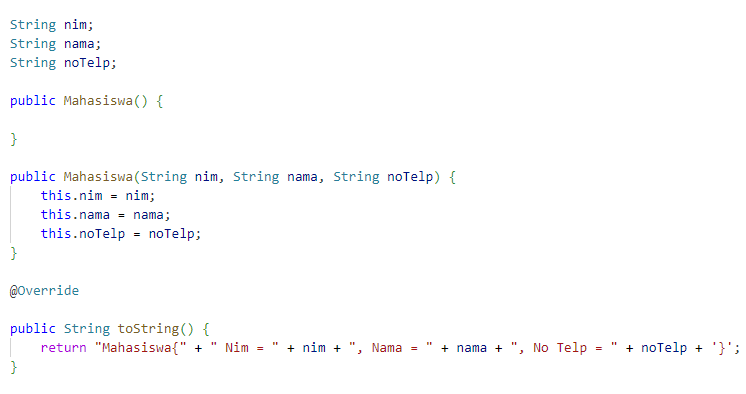
1. Tambahkan 3 buah seperti “Mango”,”guava”, dan “avocado” kemudian dilakukan sorting!





**Percobaan 3**

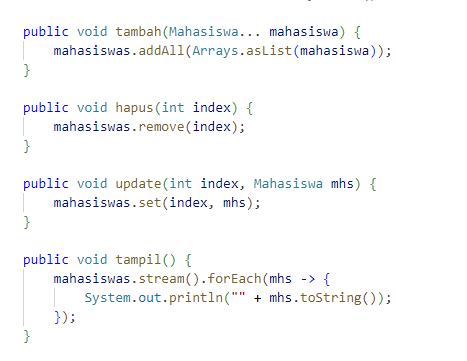
1. Buatlah sebuah class Mahasiswa dengan attribute, kontruktor, dan fungsi sebagai berikut.



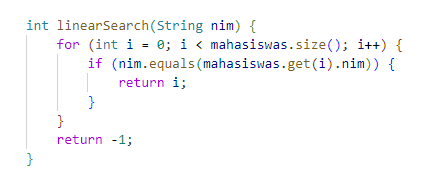
1. Selanjutnya, buatlah sebuah class ListMahasiswa yang memiliki attribute seperti di bawah ini



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



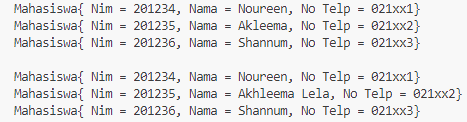
1. Untuk proses hapus, update membutuhkan fungsi pencarian terlebih dahulu yang potongan kode programnya adalah sebagai berikut



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



1. Verifikasi Hasil Percobaan



**Pertanyaan Percobaan 3**

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

Pada fungsi tambah() menggunakan konsep varargs yang mempermudah penambahan objek Mahasiswa ke dalam list mahasiswas tanpa harus membatasi jumlah argumen yang diberikan saat pemanggilan metode.

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

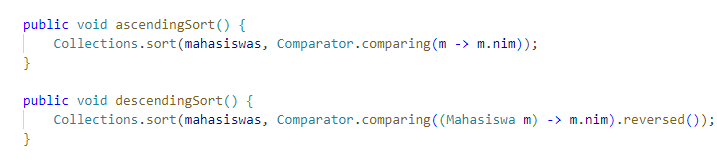


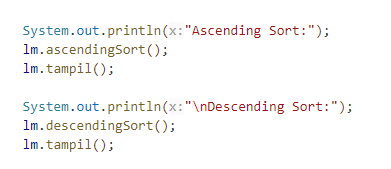


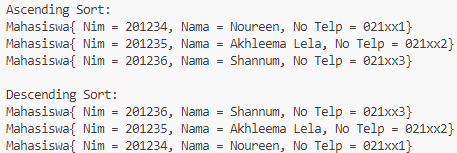


1. Tambahkan fungsi sorting baik secara ascending ataupun descending pada class tersebut!

Sorting berdasarkan NIM



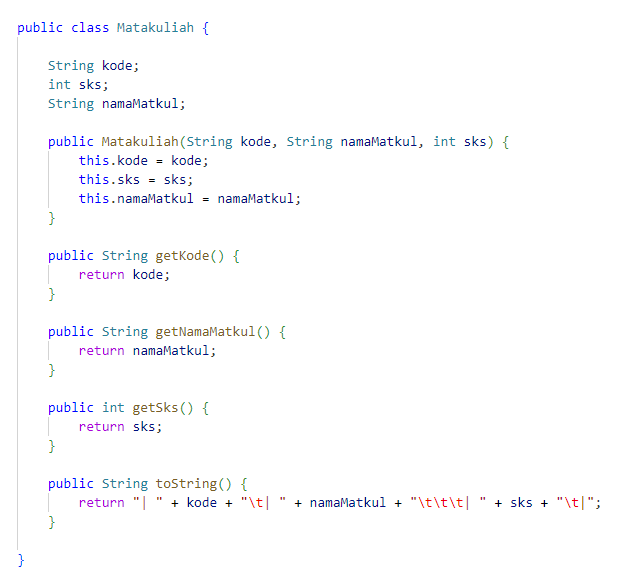




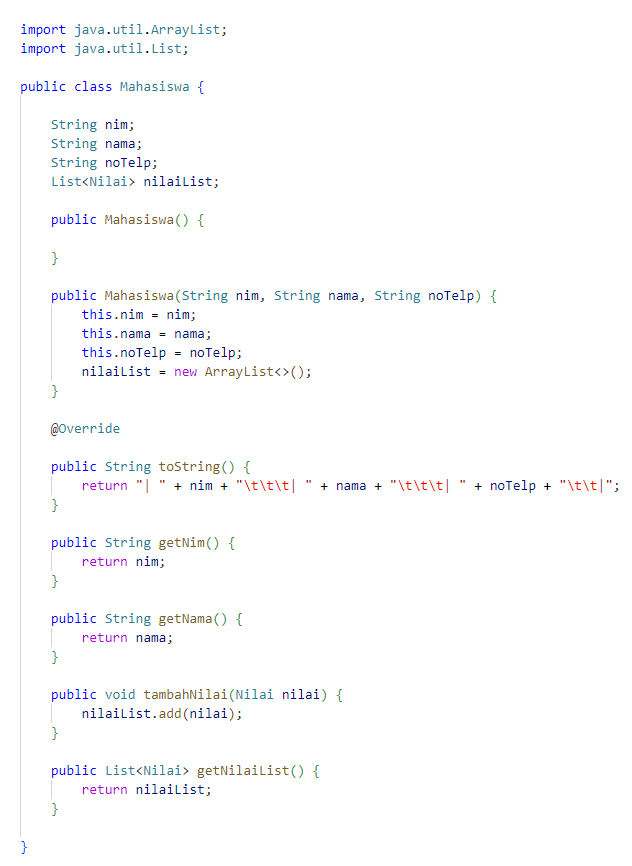
**Tugas Praktikum 1**

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

* Class Matakuliah



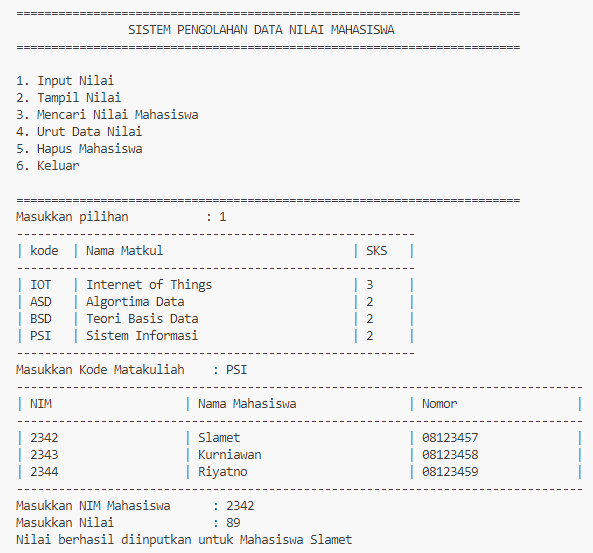
* Class Mahasiswa



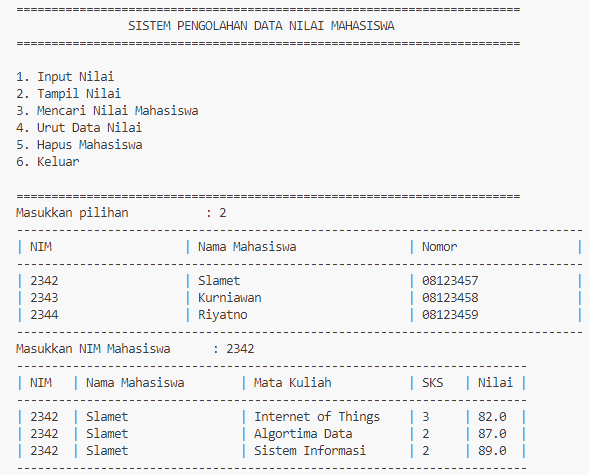
* Class Nilai



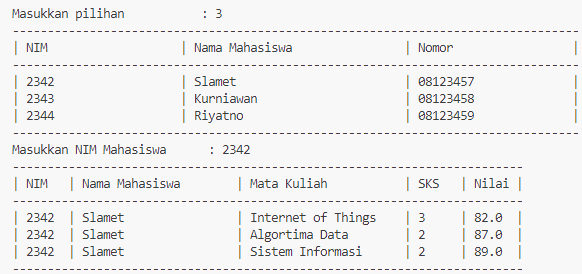
* Output Input Nilai



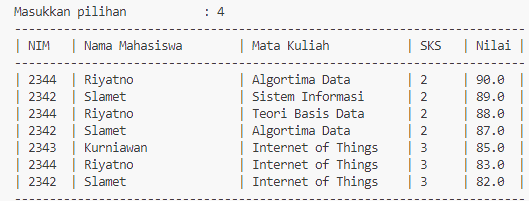
* Output Tampil Nilai



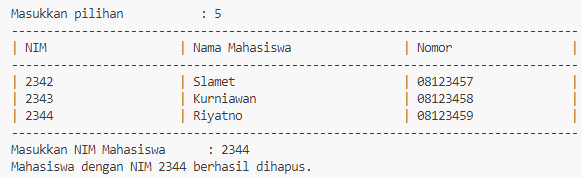
* Output Mencari Nilai Mahasiswa

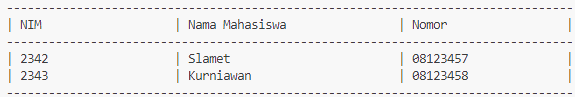


* Output Urut Data Nilai



* Output Hapus Mahasiswa





*\*full code ada di github*

Repository : <https://github.com/zidnafaz/Praktikum-Algoritma-Struktur-Data>