*Ramanda Nur Ikram*

*6706213065*

*45-05*

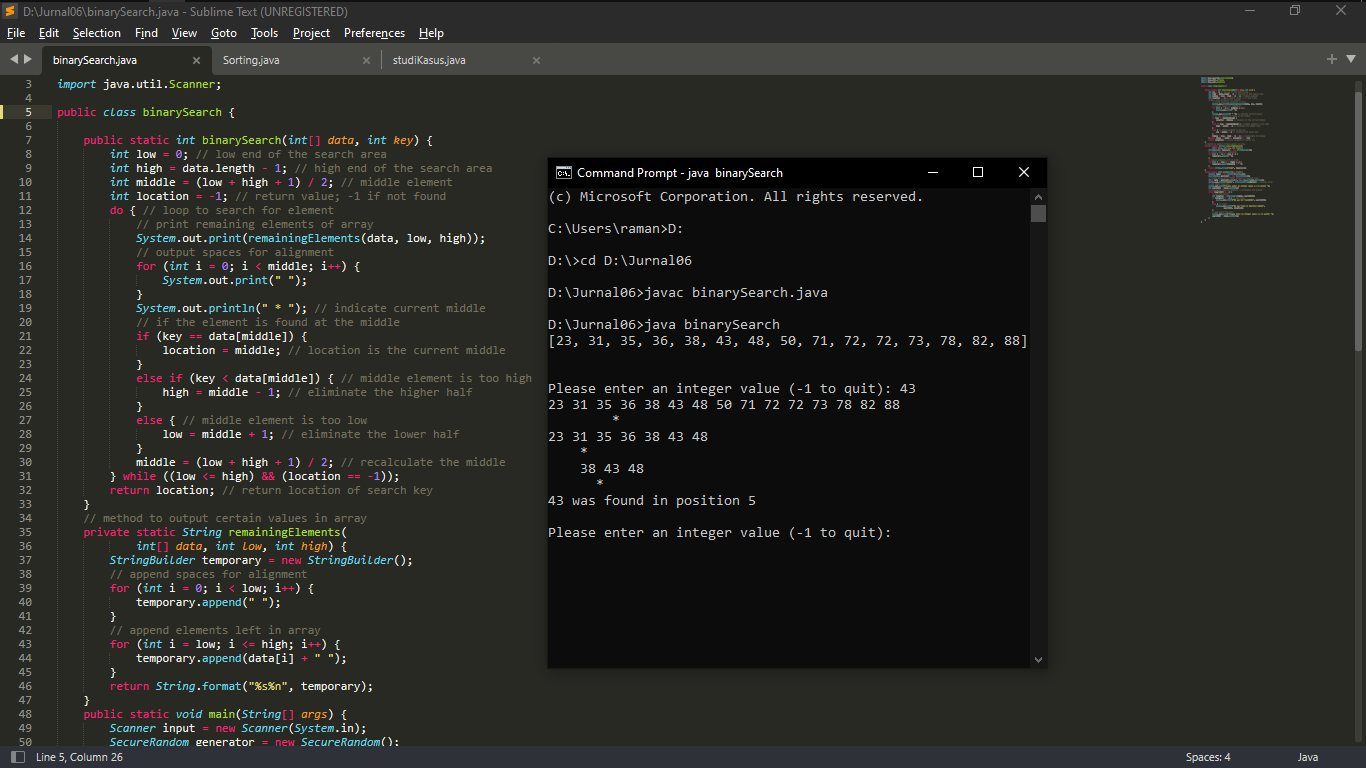
**Gdrive Link**

<https://drive.google.com/drive/folders/1nKZWgumMwp4f5Ujw9WHfB55z-1XE6HRO?usp=sharing>

**Github Link**

[**https://github.com/ramansssss/Ramanda-N.I.git**](https://github.com/ramansssss/Ramanda-N.I.git)

1. **Penjelasan koding**

**Binary Search

**Binary search** merupakan algortima yang lebih efisien daripada linear search, tetapi algoritma ini hanya dapat digunakan pada data yang telah terurut. Dalam hal ini, data yang disimpan pada array telah terurut. Pada iterasi pertama, dilakukan pengetesan pada elemen tengah array. Jika elemen ini ternyata sama dengan nilai yang dicari, algoritma akan berhenti.

Jika tidak, dengan asumsi bahwa array terurut dalam susunan ascending, jika nilai yang dicari (search key) lebih kecil daripada nilai elemen tengah, maka tidak mungkin nilai yang dicari tersebut ada di setengah bagian atas dari array. Karenanya, pencarian akan dilanjutkan pada setengah bagian awal (bawah) dari array. Hal sebaliknya terjadi jika search key lebih besar daripada nilai tengah array.

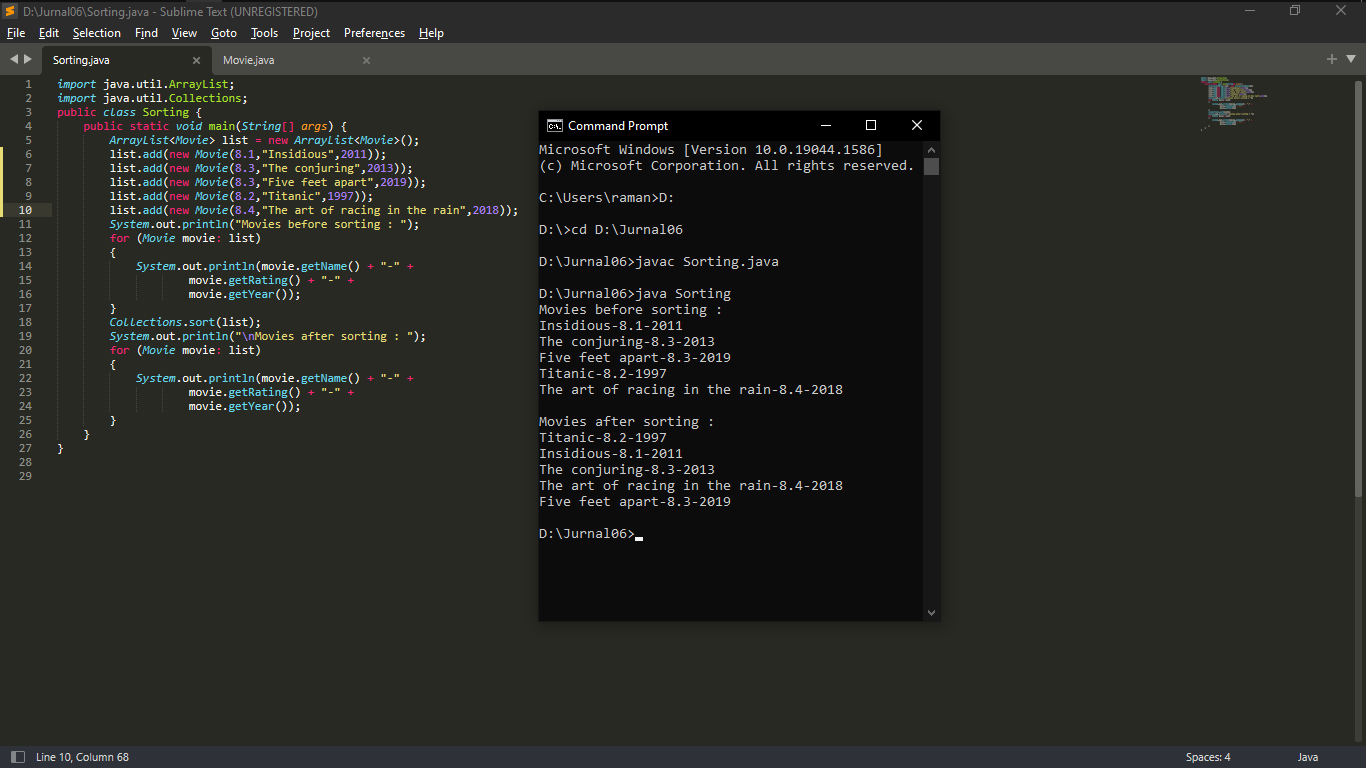
Pada screenshot diatas saya memasukan angka bernilai 43 yang pada array list di screenshot di awal angka 43 berada di posisi ke 6 atau di dalam array list adalah posisi 5. Tanda \* yang di tampilkan merupakan hasil eliminasi yang ada di codingan yang berujung pada nilai akhir 19.

Pada setiap iterasi, pengetesan dilakukan pada nilai tengah dari bagian array yang tersisa. Jika search key tidak sesuai dengan nilai elemen, algoritma ini akan menghilangkan setengah bagian lagi dari elemen yang tersisa.

Algoritma ini akan berhenti pada dua keadaan; elemen yang dicari ditemukan, atau subarray yang tersisa telah menjadi nol. Algoritma binary search merupakan algoritma yang digunakan pada Java Collections framework. Method searching tersebut pun menggunakan nama yang sama dengan algoritmanya, yaitu method binary search. Method ini dapat digunakan baik untuk LinkedList maupun ArrayList.

Dengan rumus :

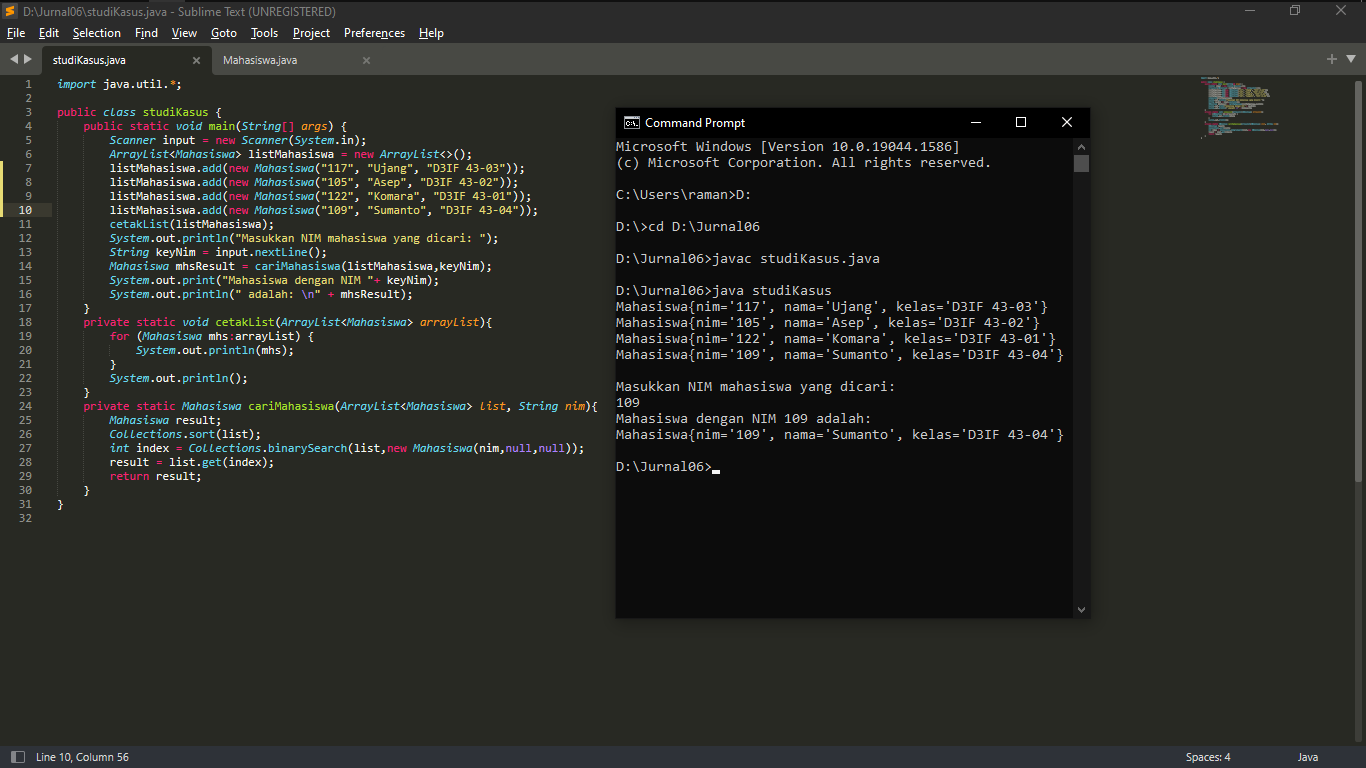
*-*Jika value < cari maka, awal = tengah + 1- Jika value > cari maka, akhir = tengah – 1 - Jika value == cari maka, pencarian berakhir

Sorting

**Algoritma sorting** akan menyusun ulang elemen dari suatu kumpulan (koleksi) sehingga elemen-elemen tersebut akan tersimpan dalam susunan yang sudah terurut.

Hal penting yang harus diingat dalam melakukan sorting adalah bahwa hasil akhir dari sorting (array yang sudah terurut) akan sama, apapun algoritma sorting yang digunakan untuk melakukan pengurutan.

Data yang akan diurutkan adalah data Movie dengan atribut rating, name dan year. Pengurutan akan dilakukan berdasarkan tahun (year). Pada kelas ini kita tidak menggunakan method toString() karena pencetakan objek dilakukan dengan menggunakan method get (baik untuk atribut name, rating, maupun year). Perhatikan, method compareTo() mengembalikan nilai dari year dengan menggunakan method getYear (tidak langung mengakses atribut year). Dengan demikian, kita akan melakukan pengurutan berdasarkan year. Namun, kekurangan penggunaan Comparable ini adalah, kita hanya dapat melakukan perbandingan pada satu atribut saja (tidak bisa digunakan untuk atribut year, name, dan rating sekaligus). Jika ingin untuk melakukan sorting berdasarkan lebih dari satu atribut, maka harus digunakan Comparator.

Studi Kasus

Mahasiswa memiliki atribut NIM, nama dan kelas. Karena binary search hanya bisa diimplementasikan pada data yang telah terurut, maka objek Mahasiswa harus dapat diurutkan, dengan demikian kelas Mahasiswa harus mengimplementasikan interface Comparable. Mahasiswa akan diurutkan berdasarkan NIM, yang terlihat pada implementasi method compareTo di kelas Mahasiswa.

Saat pembentukan objek kelas Mahasiswa dan pemanggilan method binary search, kata kunci (keyword) yang dimasukkan adalah NIM. Keyword untuk binary search dan metode pengurutan pada method compareTo haruslah berasal dari atribut yang sama (dalam kasus ini, NIM). Kemudian, karena binary search hanya dapat mencari berdasarkan tipe data elemen yang disimpan dalam array, maka key untuk binary search seharusnya adalah elemen dengan tipe data Mahasiswa. Karena itu, ketika method binary search dari Java Collections dipanggil, yang dimasukkan adalah suatu objek Mahasiswa baru, namun dengan isi dari konstruktor hanya untuk NIM saja.

1. **Insight Anda mengenai materi pada modul 06. Apa yang Anda pahami dari materi Sorting dan Searching? Apakah ada kesulitan dalam memahami materi? Silahkan juga dituliskan hal-hal lain yang Anda inginkan.**

Untuk materi yang saya pahami adalah pengertian dan cara kerja system nya dan untuk kesulitan mungkin dari algoritma codingnya karena sangat sulit bagi saya saat meng implementasikanya.

Hal yang saya ingin capai yaitu dari sekarang saya ingin lebih mengenal tentang apa itu java dari segi penerapanya dan hal lainya. Karena sampai detik ini pun saya masih memerlukan bantuan dari orang lain. Maaf pa atas keterlambatan pengumpulan tugas.