**LAPORAN PRAKTIKUM**

**MATA KULIAH ALGORITMA DAN STRUKTUR DATA**

**PERTEMUAN 7 : SEARCHING**



**KAYLA RACHMAUDINA SATITI PUTRI**

**2341760103**

**D-IV SISTEM INFORMASI BISNIS**

**JURUSAN TEKNOLOGI INFORMASI POLITEKNIK**

**NEGERI MALANG**

**2024**

**JOBSHEET VI** **SEARCHING**

# 6.1. Tujuan Praktikum

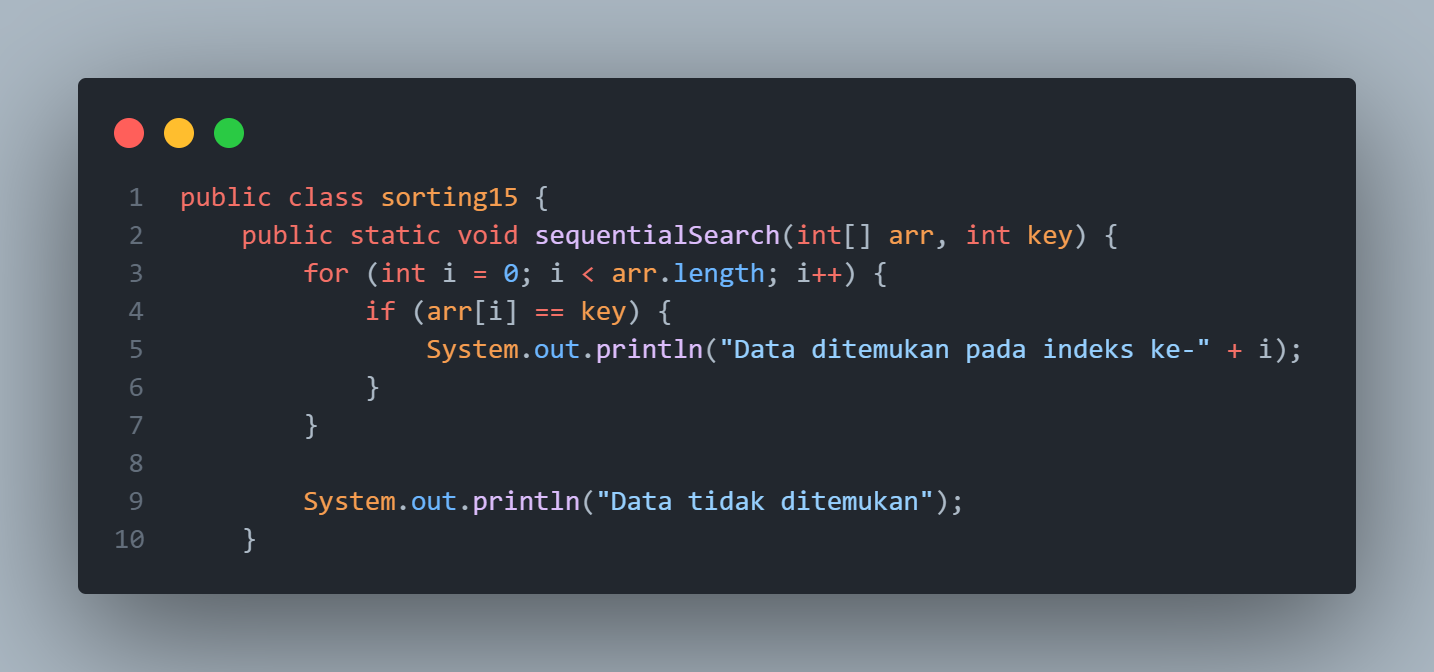
Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Menjelaskan mengenai algoritma Searching.
2. Membuat dan mendeklarasikan struktur algoritma Searching.
3. Menerapkan dan mengimplementasikan algoritma Searching.

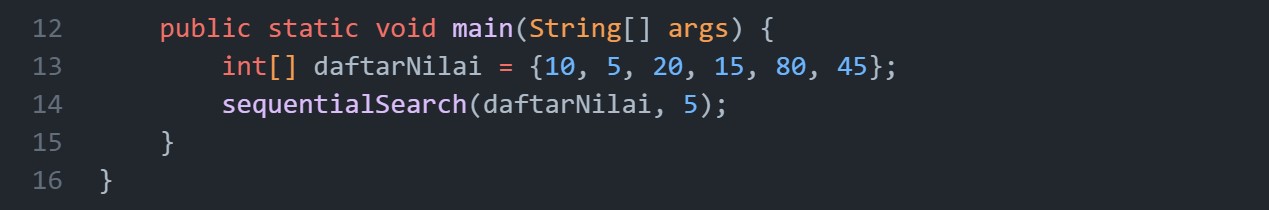
# 6.2. Searching / Pencarian Menggunakan Algoritma Sequential Search

## 6.2.1 Sequential Search Menggunakan Array

1. Buat folder baru dengan nama Praktikum06. Buat file dengan nama Sorting.java
2. Tambahkan method sequentialSearch() yang melakukan pencarian data bertipe integer di dalam array of integer



1. Tambahkan fungsi main sebagai berikut

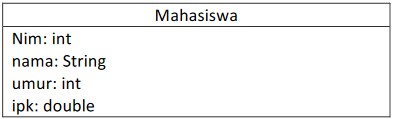


1. Compile dan run program



## 6.2.2 Sequential Search Menggunakan Array of Object

Perhatikan diagram class Mahasiswa di bawah ini! Diagram class ini yang selanjutnya akan dibuat sebagai acuan dalam membuat kode program class Mahasiswa.



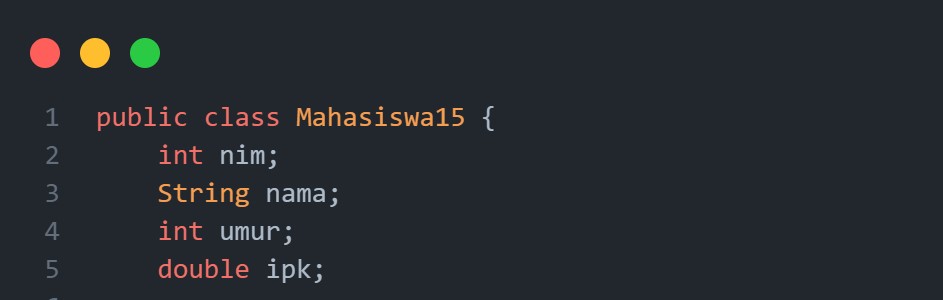
Berdasarkan class diagram di atas, akan dibuat class Mahasiswa yang berfunsi untuk membuat objek mahasiswa yang akan dimasukan ke dalam sebuah array. Terdapat sebuah konstruktor berparameter dan juga fungsi tampil() untuk menampilkan semua attribute yang ada.

|  |
| --- |
| PencarianMhs |
| listMhs: Mahasiswa[5]  idx: int |
| tambah(mhs: Mahasiswa): void  tampil(): void  FindSeqSearch(int cari): int  Tampilpoisisi(int x,int pos): void  TampilData(int x,int pos) :void |

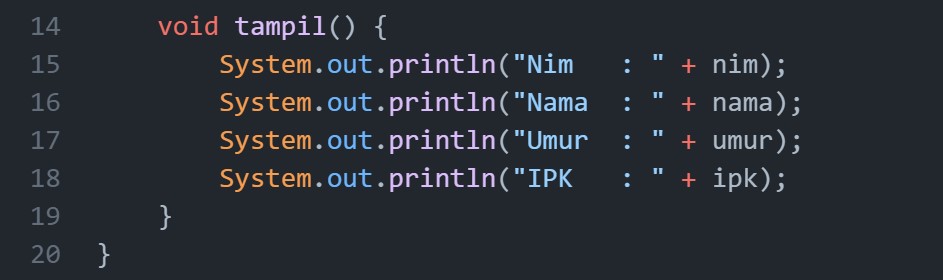
Selanjutnya class diagram di atas merupakan representasi dari sebuah class yang berfungsi untuk melakukan operasi-operasi dari objek array mahasiswa, misalkan untuk menambahkan objek mahasiswa, menampilkan semua data mahasiswa, untuk melakukan pencarian berdasarkan NIM menggunakan algoritma Sequential Search, menampilkan posisi dari data yang dicari, serta menampilkan data mahasiswa yang dicari.

#  Langkah-langkah Percobaan Sequential Search

1. Buatlah Project baru pada Netbeans dengan nama **TestSearching**
2. Kemuadian buat packages baru dengan nama **minggu7**.
3. Buat class **Mahasiswa**, kemudian deklarasikan atribut berikut ini:

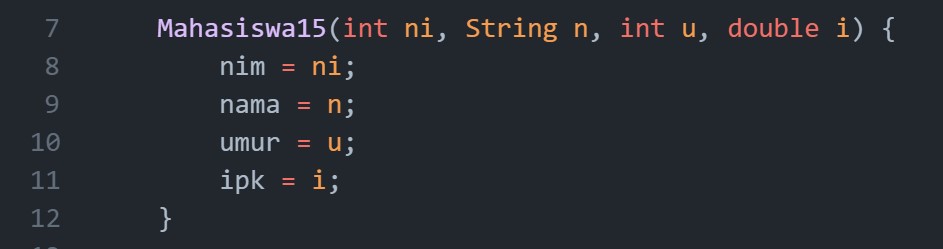


1. Buatlah konstruktor dengan nama **Mahasiswa** dengan parameter (**int ni, String n, int**



**u, double i**

)



5.



Buatlah method



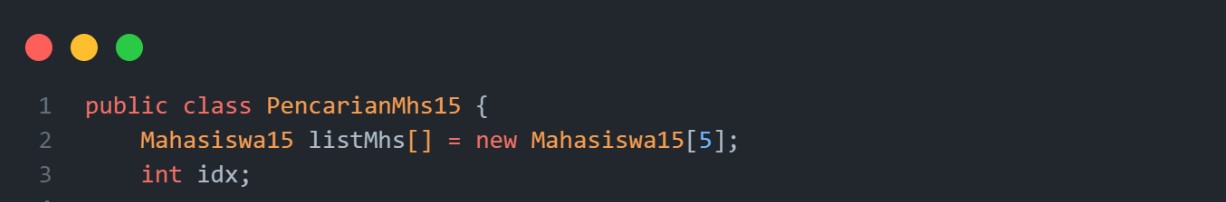
**tampil**



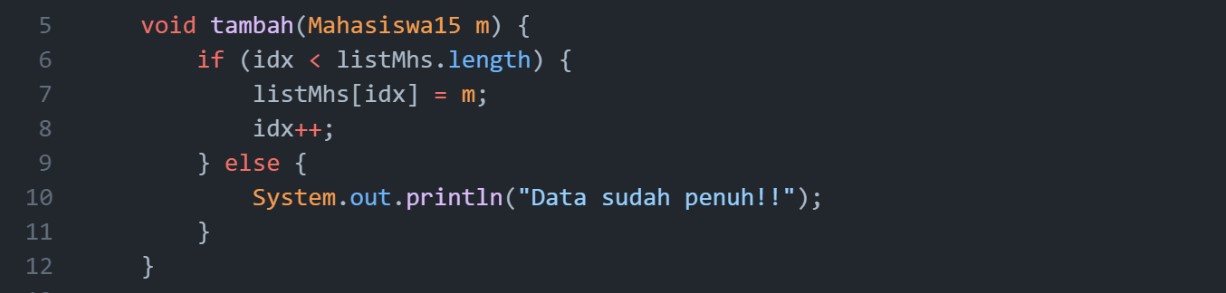
bertipe void.



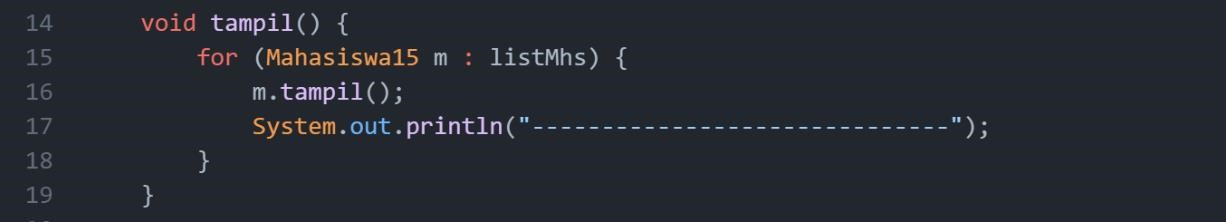
* 1. Buat class baru dengan nama **PencarianMhs**



* 1. Tambahkan method **tambah()** di dalam class tersebut! Method **tambah()** digunakan untuk menambahkan objek dari class Mahasiswa ke dalam atribut listMhs.



* 1. Tambahkan method **tampil()** di dalam class **PencarianMhs**! Method **tampil()** digunakan untuk menampilkan semua data mahasiswa-mahasiswa yang ada di dalam class tersebut! Perhatikan penggunaan sintaks for yang agak berbeda dengan for yang telah dipelajari sebelumnya, meskipun secara konsep sebenarnya mirip.



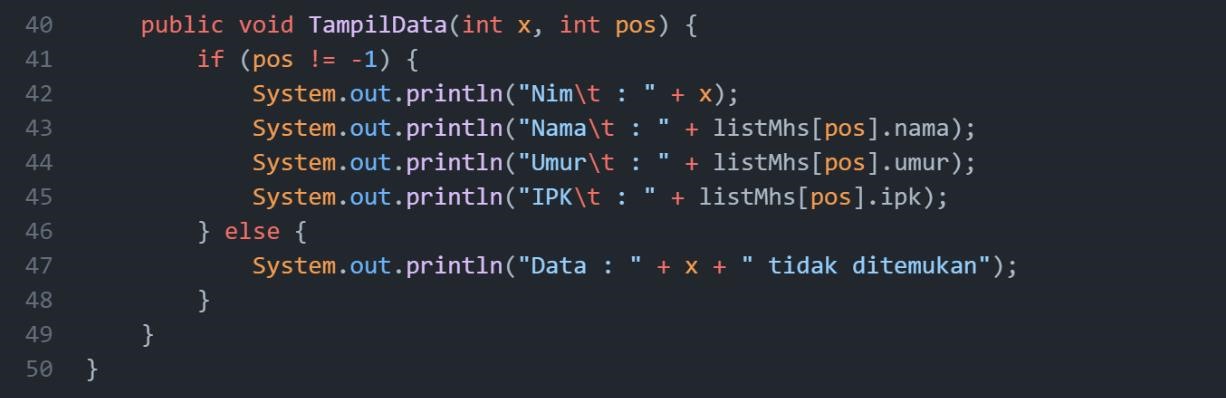
* 1. Tambahkan method **FindSeqSearch** bertipe integer dengan parameter **cari** bertipe integer. Kemudian Deklarasikan isi method **FindSeqSearch** dengan algoritma pencarian data menggunakan teknik sequential searching.



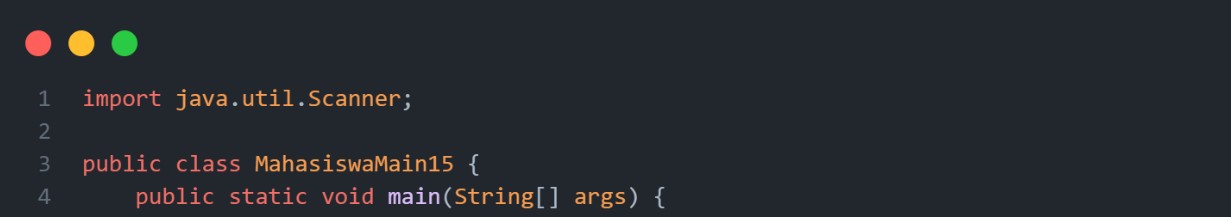
* 1. Buatlah method **Tampilpoisisi** bertipe void dan Deklarasikan isi dari method



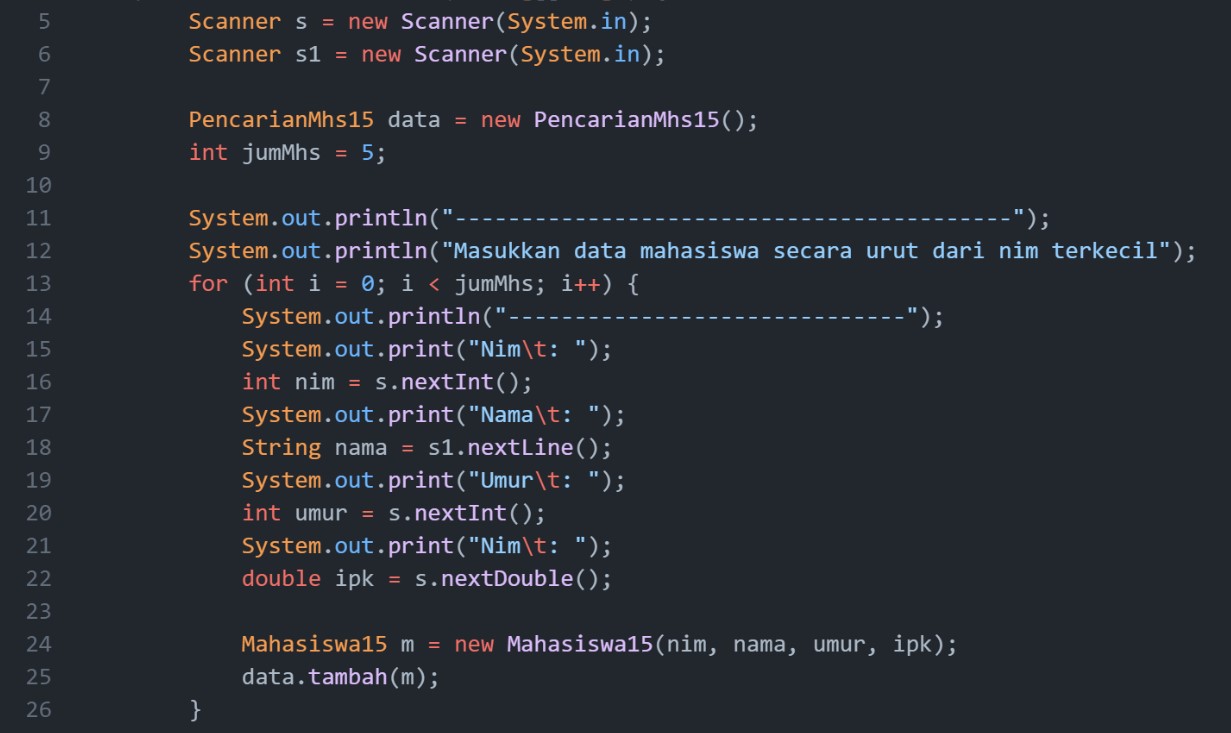
* 1. Buatlah method **TampilData** bertipe void dan Deklarasikan isi dari method **TampilData.**



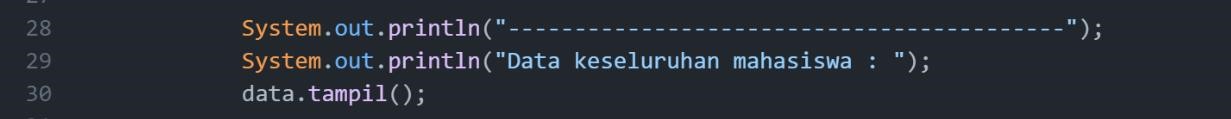
* 1. Buatlah class baru dengan nama **MahasiswaMain** tambahkan method **main** seperti pada gambar berikut!



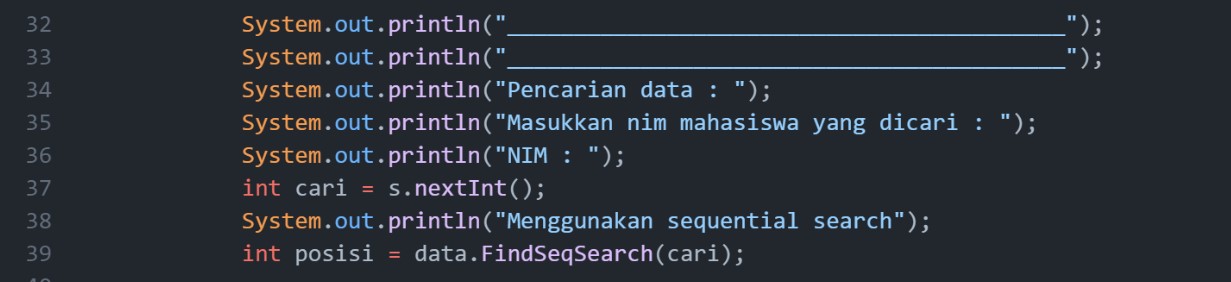
* 1. Di dalam method **main(),** buatlah sebuah objek PencarianMhs dan buatlah 5 objek mahasiswa kemudian tambahkan semua objek mahasiswa tersebut dengan memanggil fungsi tambah pada objek PencarianMhs.



* 1. Panggil method **tampil()** untuk melihat semua data yang telah dimasukan.



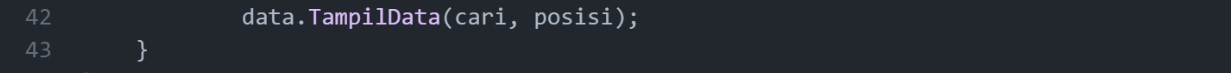
* 1. Untuk melakukan pencarian berdasarkan NIM mahasiswa. Buatlah variable **cari** yang dapat menampung masukan dari keyboard lalu panggil method **FindSeqSearch** dengan isi parameternya adalah variable cari.



* 1. Lakukan pemanggilan method **Tampilposisi** dari class **PencarianMhs.**



* 1. Lakukan pemanggilan method **TampilData** dari class **PencarianMhs.**



* 1. Jalankan dan amati hasilnya.

## 6.2.2. Verifikasi Hasil Percobaan





1.

Lakukan perubahan array daftarNilai pada fungsi main().



**6.2.3.**



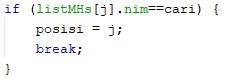
**Pertanyaan**



1. Jelaskan perbedaan method **TampilData** dan **Tampilposisi** pada class PencarianMhs

TampilData menampilkan informasi lengkap data, sedangkan TampilPosisi hanya menunjukkan posisinya di array.

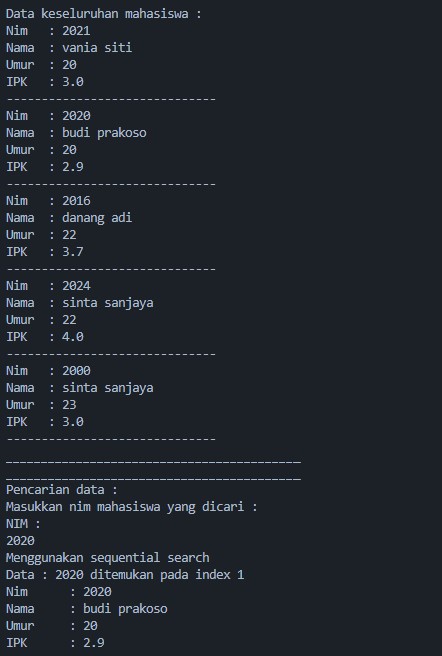
1. Jelaskan fungsi **break** pada kode program dibawah ini!



Fungsi break pada potongan kode tersebut digunakan untuk menghentikan loop for setelah data yang dicari dengan NIM cari ditemukan.

1. Jika Data Nim yang dimasukkan tidak terurut dari kecil ke besar. Apakah program masih dapat berjalan? Apakah hasil yang dikeluarkan benar? Mengapa demikian!

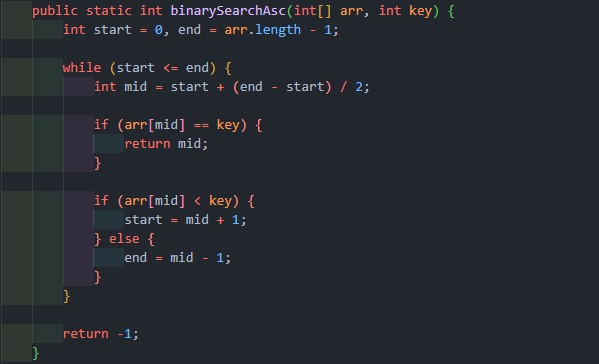
Program masih dapat berjalan dengan data yang tidak terurut, namun hasil yang dikeluarkan bisa jadi tidak akurat untuk method TampilData. Hal ini terjadi karena method TampilData bergantung pada urutan data yang ditemukan oleh FindSeqSearch, sedangkan data tidak terurut.



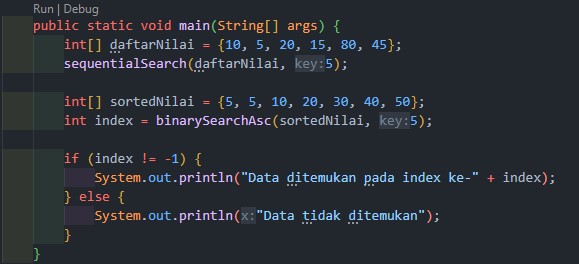
# 6.3. Searching / Pencarian Menggunakan Binary Search

## 6.3.1. Langkah-langkah Percobaan Binary Search menggunakan Array

1. Tambahkan method binarySearchAsc() pada file Sorting.java



1. Tambahkan baris program untuk menguji method binarySearchAsc() pada fungsi main()

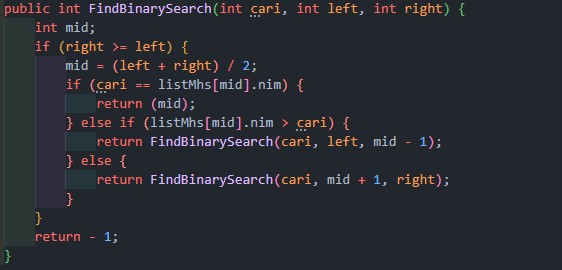


1. Run dan compile program



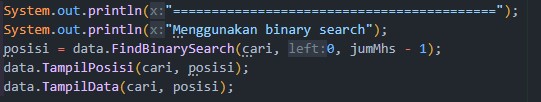
## 6.3.2. Langkah-langkah Percobaan Binary Search menggunakan Array of Object

1. Pada percobaan 6.2.2 (sequential search) tambahkan method **FindBinarySearch** bertipe integer pada class **PencarianMhs**. Kemudian Deklarasikan isi method **FindBinarySearch** dengan algoritma pencarian data menggunakan teknik binary searching.

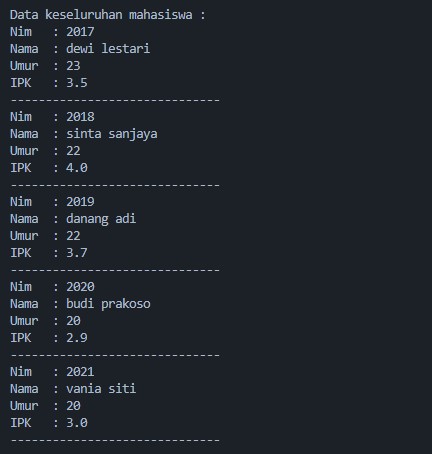
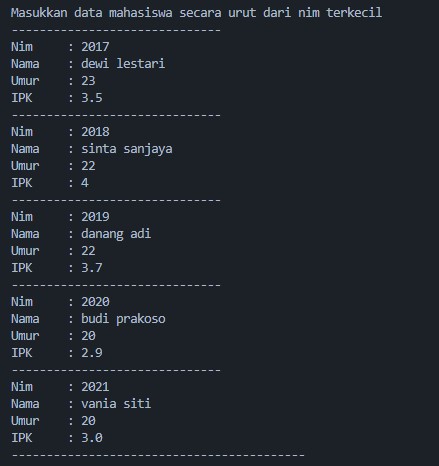


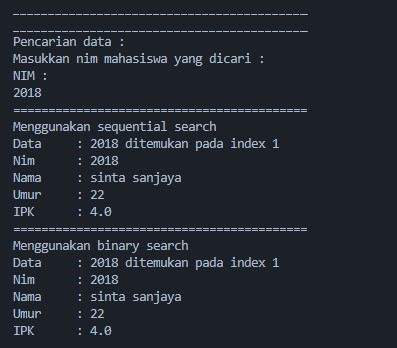
1. Panggil method **FindBinarySearch** terdapat pada class **PencarianMhs** di kelas

**Mahasiswamain.** Kemudia panggil **method tampilposisi** dan **tampilData**



## 6.3.2. Verifikasi Hasil Percobaan





1. Tunjukkan pada kode program yang mana proses divide dijalankan!



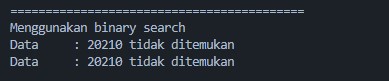
1. Tunjukkan pada kode program yang mana proses conquer dijalankan!

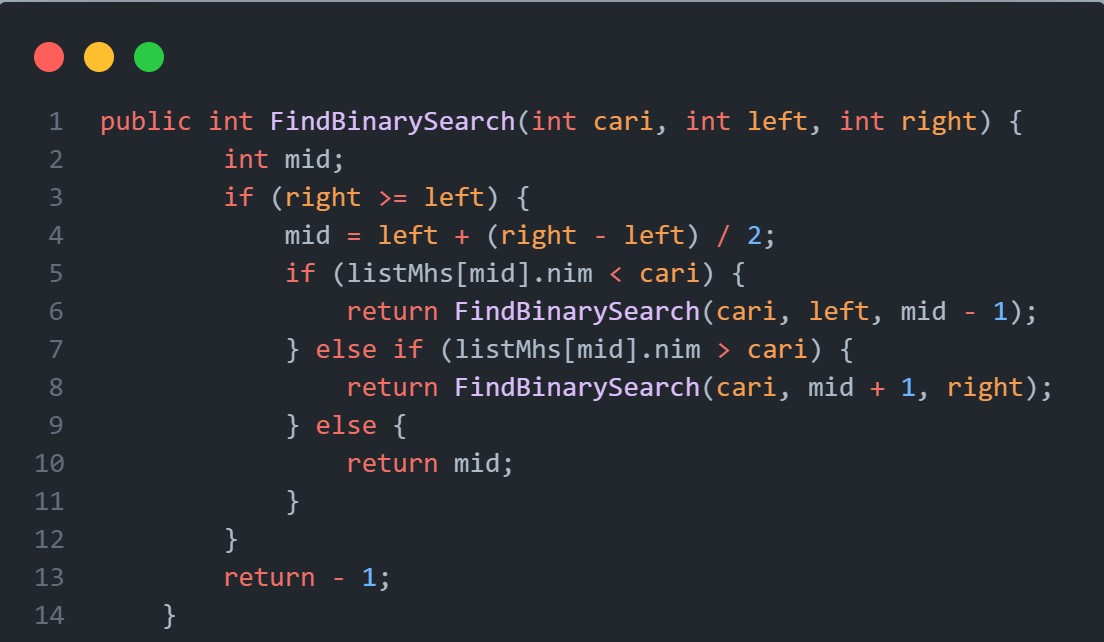
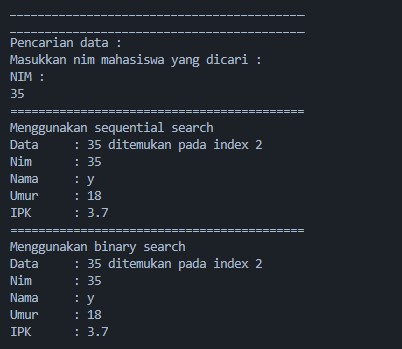


1. Jika data Nim yang dimasukkan tidak urut. Apakah program masih dapat berjalan? Mengapa demikian!

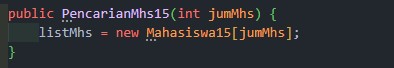
Program masih dapat berjalan dengan data yang tidak terurut, namun hasil yang dikeluarkan bisa jadi tidak akurat. Hal ini dikarenakan metode pencarian binary membutuhkan data yang telah berurutan sebelumnya agar dapat bekerja dengan benar.

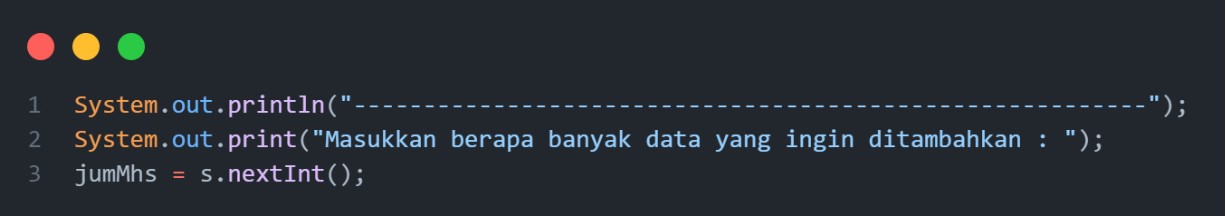
1. Jika Nim yang dimasukkan dari NIM terbesar ke terkecil (missal : 20215, 20214, 20212, 20211, 20210) dan elemen yang dicari adalah 20210. Bagaimana hasil dari binary search? Apakah sesuai? Jika tidak sesuai maka ubahlah kode program binary seach agar hasilnya sesuai





1. Modifikasilah program diatas yang mana jumlah mahasiswa yang di inputkan sesuai dengan masukan dari keyboard.



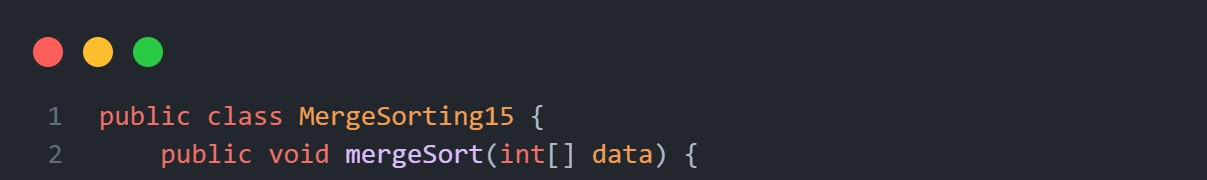


# 6.4. Percobaan Pengayaan Divide and Conquer

## 6.4.1. Langkah-langkah Percobaan Merge Sort

1 Buatlah Package baru pada NetBeans dengan nama **MergeSortTest**

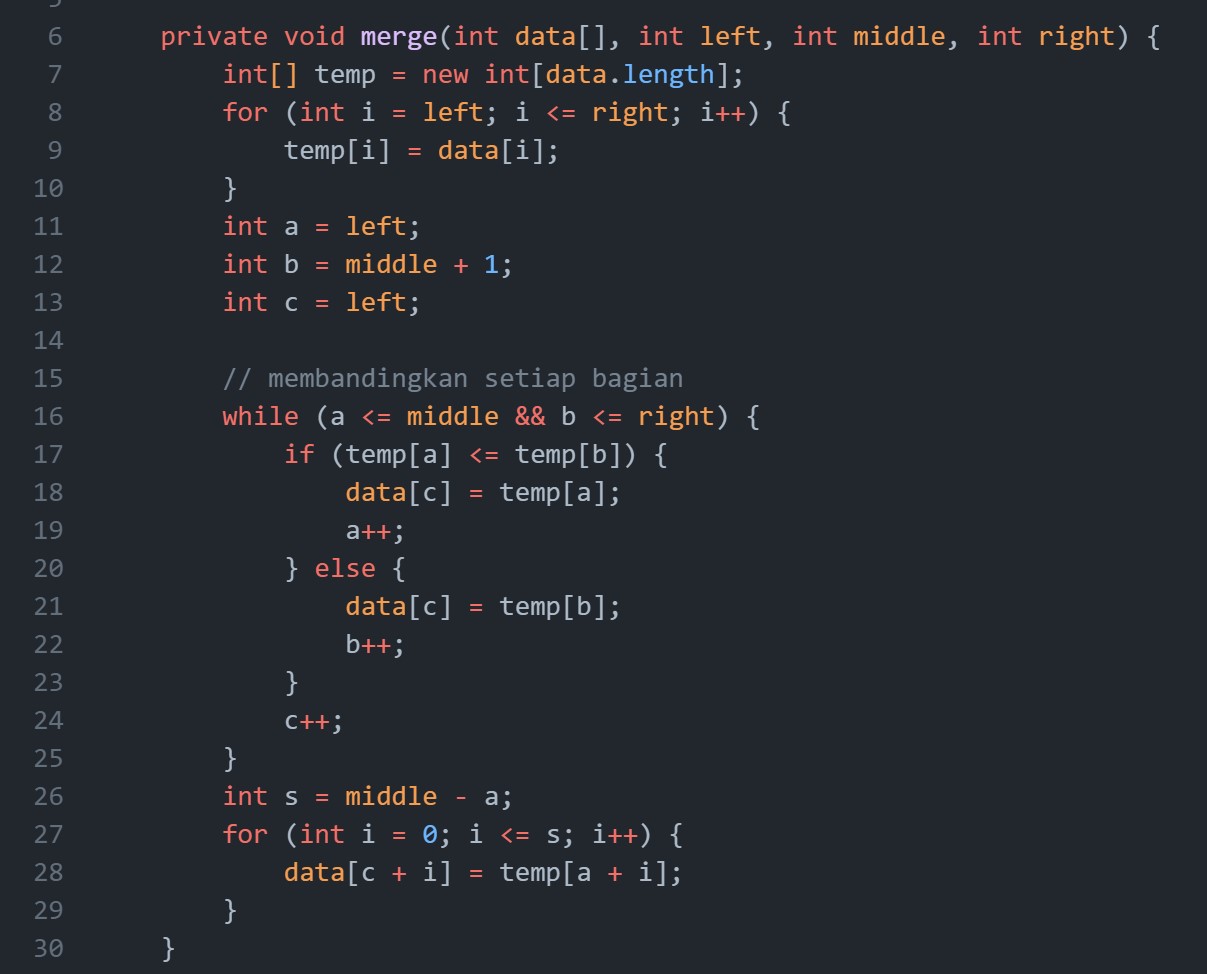
1. Tambahkan class **MergeSorting** pada package tersebut
2. Pada class **MergeSorting** buatlah method **mergeSort** yang menerima parameter data array yang akan diurutkan



1. Buatlah method **merge** untuk melakukan proses penggabungan data dari bagian kiri dan kanan.



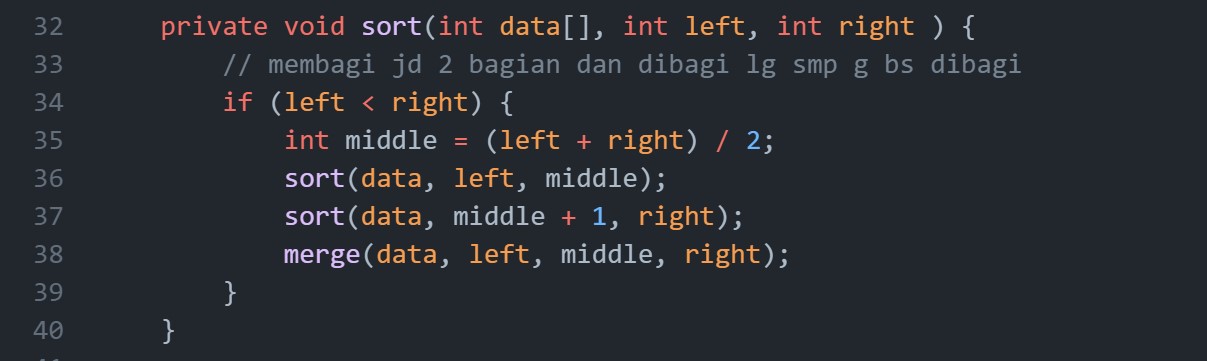
1. Implementasikan proses merge sebagai berikut.



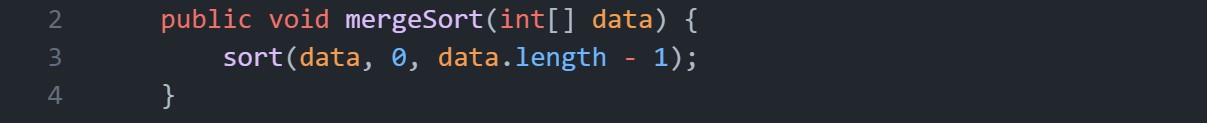
1. Buatlah method sort



1. Implementasikan kode berikut pada method sort



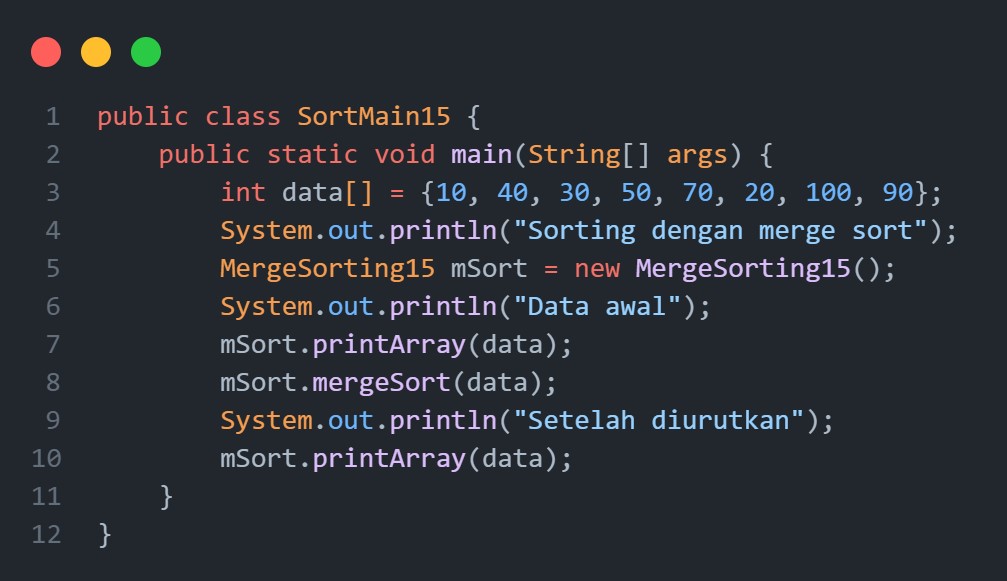
1. Pada method **mergeSort**, panggil method **sort** dengan parameter data yang ingin diurutkan serta range data awal sampai dengan akhir.



1. Tambahkan method **printArray**

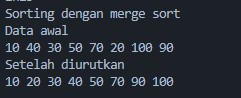


1. Sebagai langkah terakhir, deklarasikan data yang akan diurutkan kemudian panggil proses sorting pada class SortMain



### 6.4.2. Verifikasi Hasil Percobaan

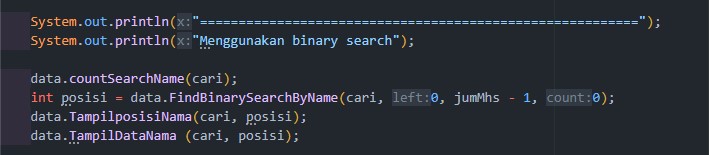
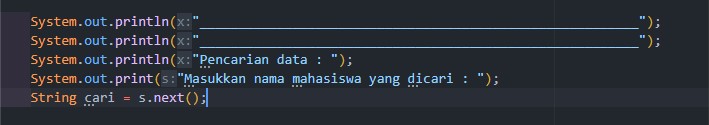
Cocokkan hasil compile kode program anda dengan gambar berikut ini.



# 6.5. Latihan Praktikum

1. Modifikasi percobaan searching diatas yang menggunakan Searching array of object dengan ketentuan berikut ini

* Pencarian dilakukan berdasarkan Nama Mahasiswa (gunakan Algoritma binary Search)
* Buat aturan untuk mendeteksi hasil pencarian lebih dari 1 hasil dalam bentuk kalimat peringatan!





## OUTPUT

