# LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

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

# Jobsheet-7 SEARCHING



Nama : Surya Rahmat Fatahillah

NIM : 2341760020

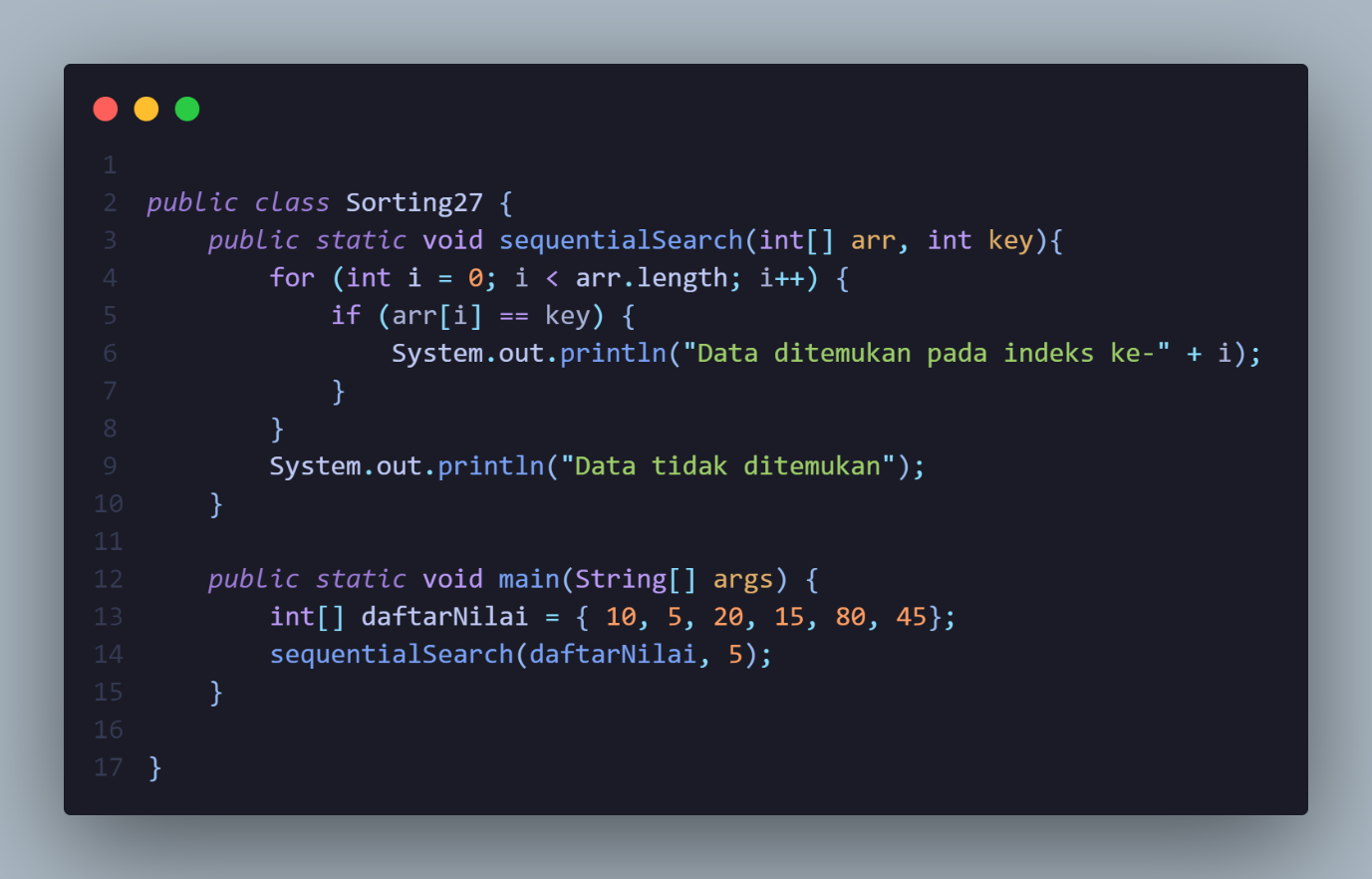
Prodi : Sistem Informasi Bisnis

# JURUSAN TEKNOLOGI INFORMASI POLITEKNIK NEGERI MALANG 2023/2024

# 6.2. Searching / Pencarian Menggunakan Algoritma Sequential Search

**6.2.1 Sequential Search Menggunakan Array**

Buat folder baru dengan nama Praktikum06. Buat file dengan nama Sorting.java. Tambahkan method sequentialSearch() yang melakukan pencarian data bertipe integer di dalam array of integer dan fungsi main sebagai berikut

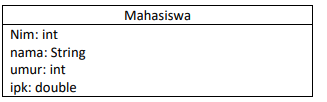
****

Compile dan run program

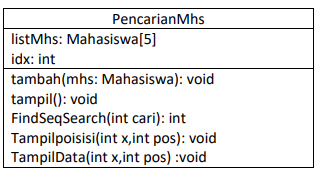


**6.2.2 Sequential Search Menggunakan Array of Object**

Berdasarkan class diagram di bawah, 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.

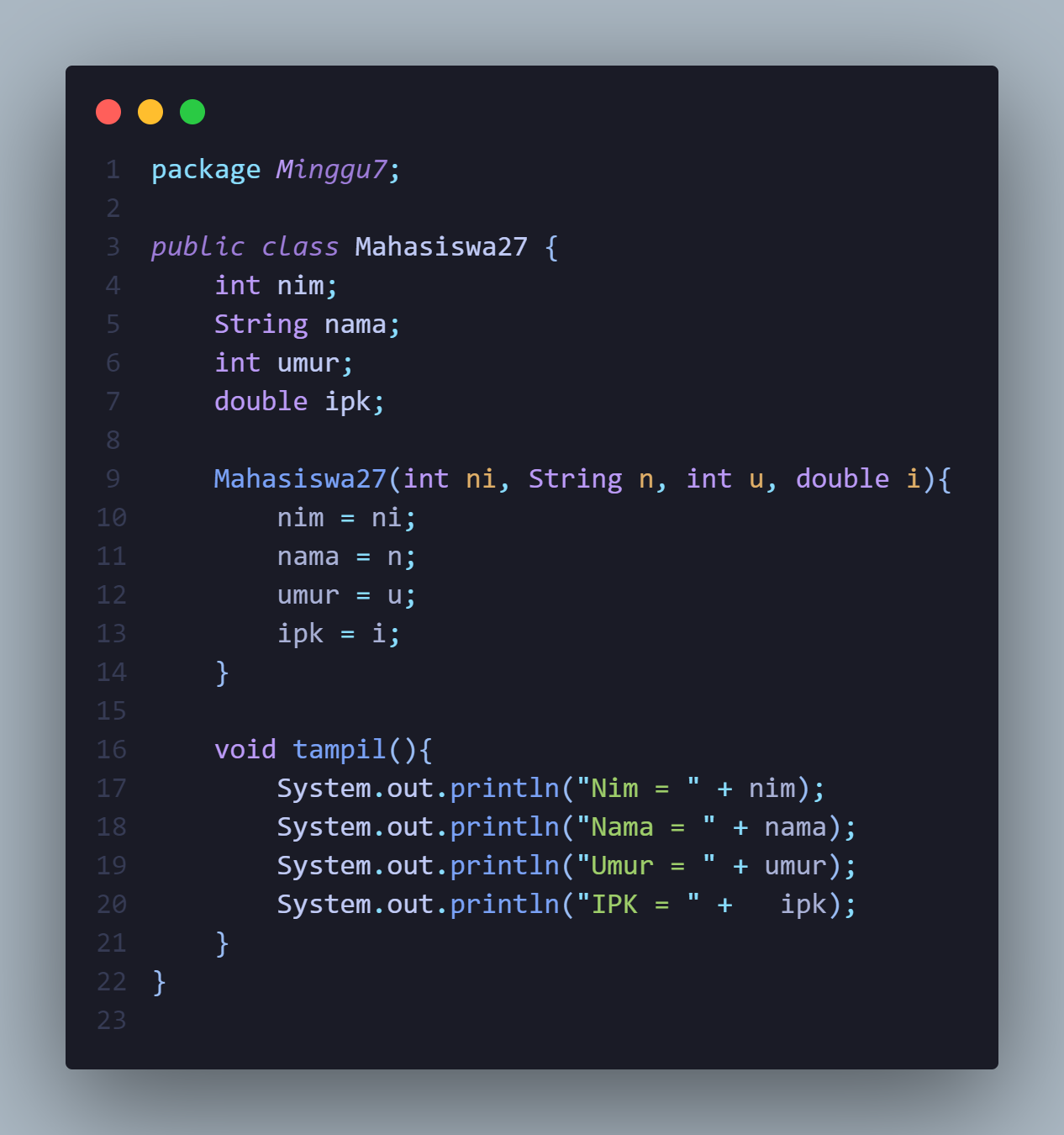
****

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.



**Hasil dari Langkah-langkah Percobaan Sequential Search**

* class Mahasiswa

****

* Class PencarianMhs

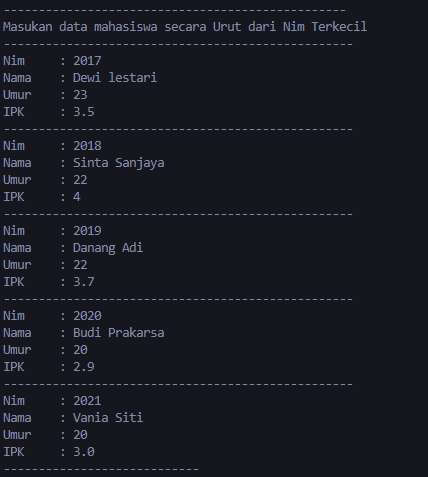


* Class MahasiswaMain



**6.2.2. Verifikasi Hasil Percobaan**

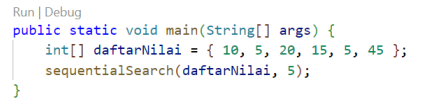
Cocokkan hasil kode program anda dengan gambar berikut ini

****

****

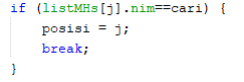
**6.2.3. Pertanyaan**

1. Lakukan perubahan array daftarNilai pada fungsi main().



2. Jelaskan perbedaan metod TampilData dan Tampilposisi pada class PencarianMhs

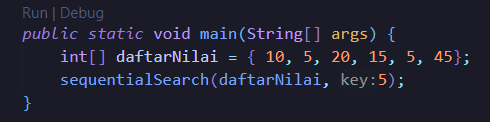
3. Jelaskan fungsi break pada kode program dibawah ini!



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

**Jawaban!**

1. Berikut perubahan array daftar nilai pada fungsi main()



1. Metode **TampilData** dan **TampilPosisi** pada class **PencarianMhs27** memiliki perbedaan dalam tugas atau output yang mereka lakukan.

Metode **TampilData** bertanggung jawab untuk menampilkan seluruh data mahasiswa yang sesuai dengan indeks yang ditemukan melalui pencarian.

Jika data ditemukan pada indeks tertentu, metode ini akan menampilkan semua atribut (NIM, Nama, Umur, IPK) dari mahasiswa yang sesuai dengan indeks tersebut.

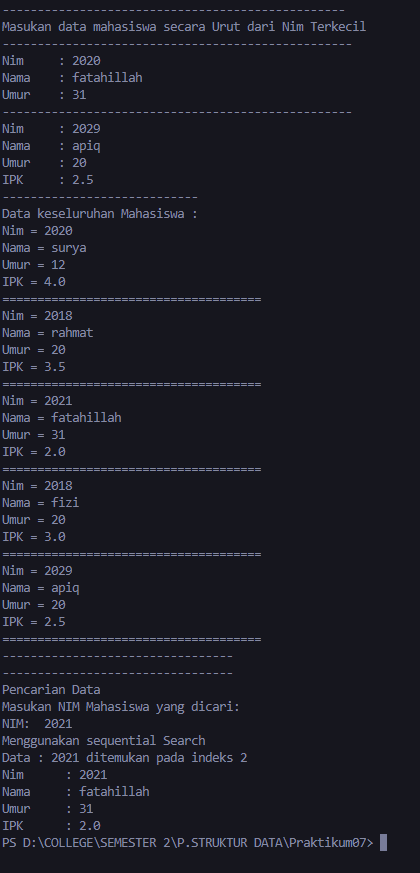
Jika data tidak ditemukan (posisi = -1), maka akan menampilkan pesan bahwa data tidak ditemukan.

Sedangkan metode **TampilPosisi** bertanggung jawab untuk menampilkan hanya posisi indeks dari data yang dicari.

Jika data ditemukan pada indeks tertentu, metode ini hanya akan menampilkan pesan yang menyatakan bahwa data tersebut ditemukan pada indeks tersebut.

Jika data tidak ditemukan (posisi = -1), maka akan menampilkan pesan bahwa data tidak ditemukan.

1. Pernyataan **break** pada kode program tersebut memiliki fungsi untuk menghentikan eksekusi perulangan **for** ketika kondisi yang ditentukan terpenuhi. Dalam konteks kode yang diberikan. Kode tersebut akan mengecek apakah terdapat mahasiswa dengan NIM yang sama dengan nilai yang ingin dicari (**cari**). Jika nilai NIM dari suatu mahasiswa sama dengan nilai yang dicari, maka pernyataan **break** akan menghentikan eksekusi perulangan **for**, sehingga tidak ada iterasi tambahan yang dilakukan. Ini berarti setelah menemukan data yang dicari, pencarian dihentikan dan tidak perlu melanjutkan iterasi ke elemen array selanjutnya.
2. Masih bisa, kemudian berikut untuk hasil yang dikeluarkan oleh program:

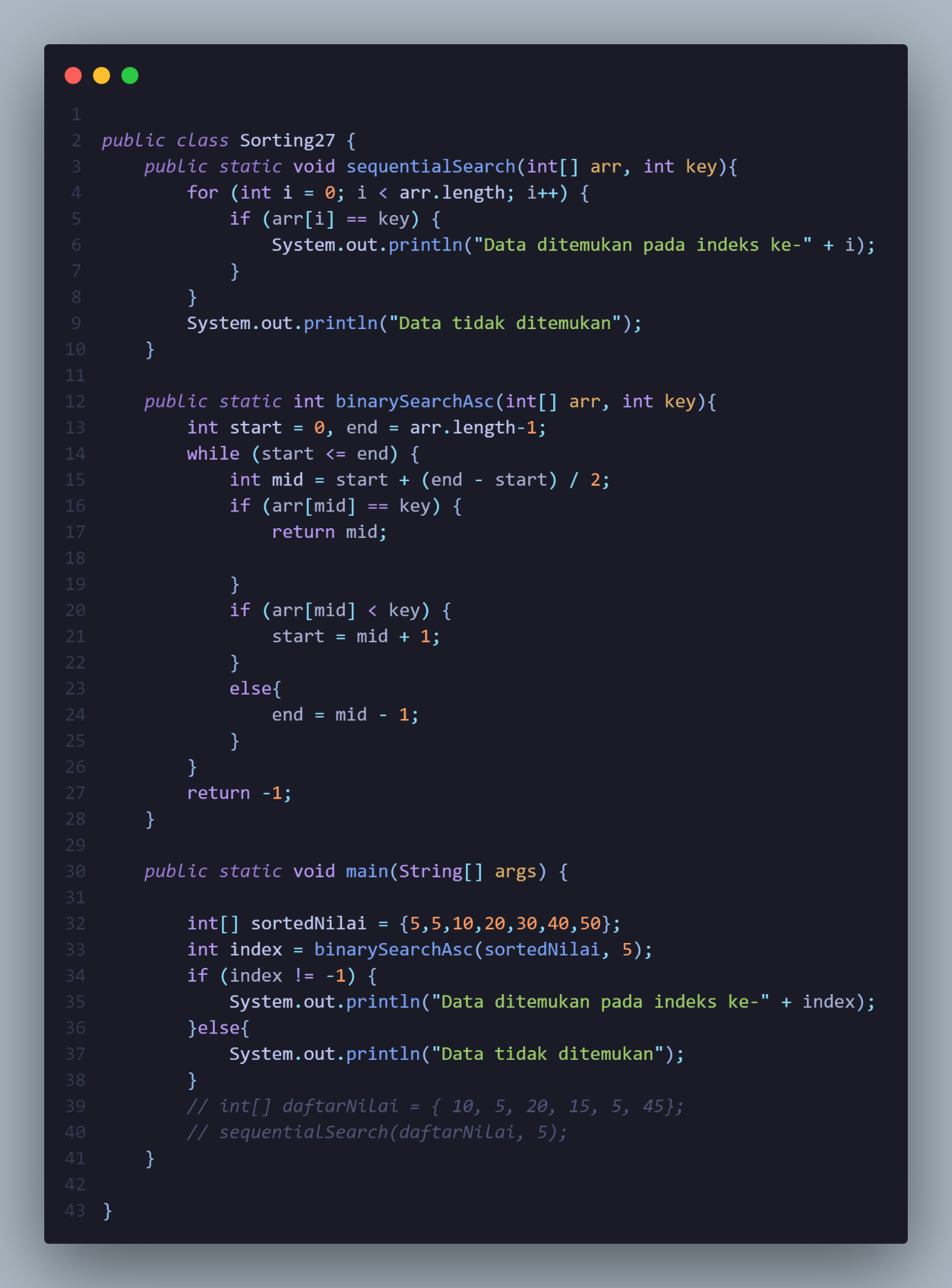


Hal tersebut bisa terjadi, karena kita menggunakan sequential search yang mana algoritma ini akan mengecek satu persatu nilai dari indeks secara bergantian tanpa ada kewajiban untuk melakukan pengurutan terlebih dahulu.

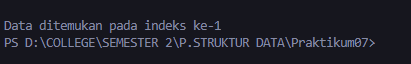
**6.3. Searching / Pencarian Menggunakan Binary Search**

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

Tambahkan method binarySearchAsc() pada file Sorting.java dan Tambahkan baris program untuk menguji method binarySearchAsc() pada fungsi main()

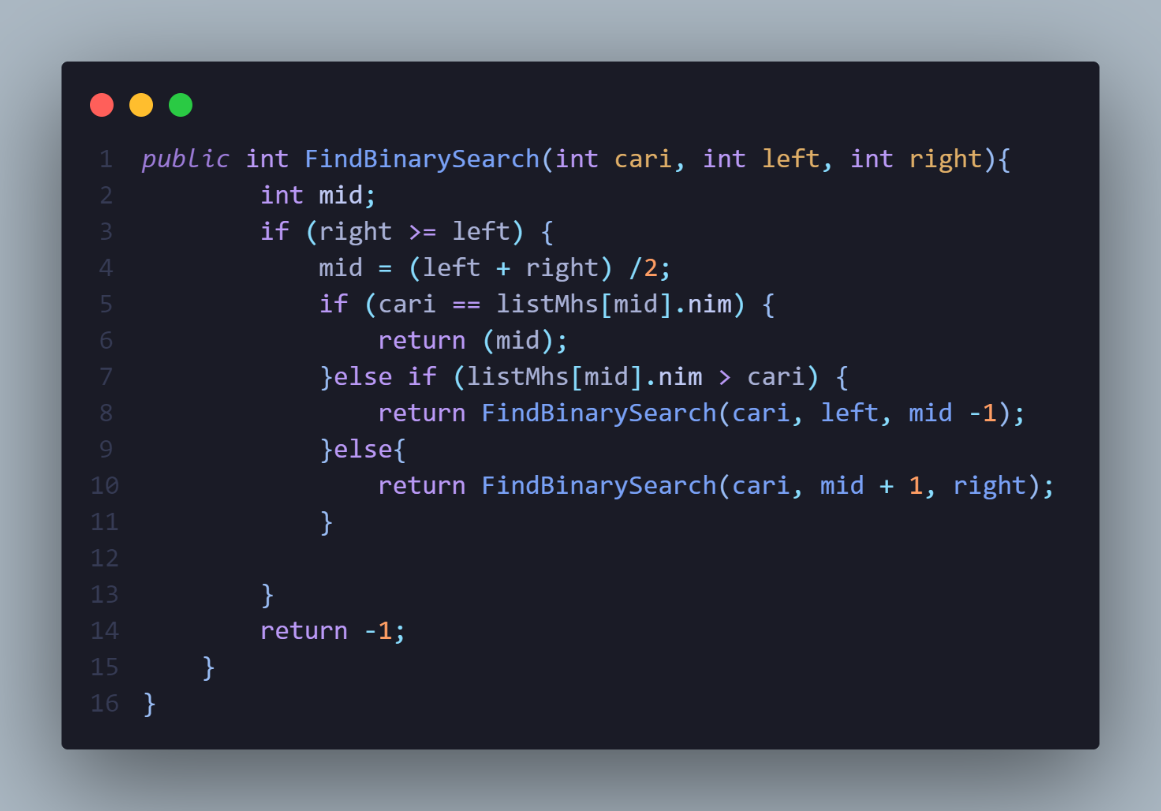
****

**Run dan compile program**

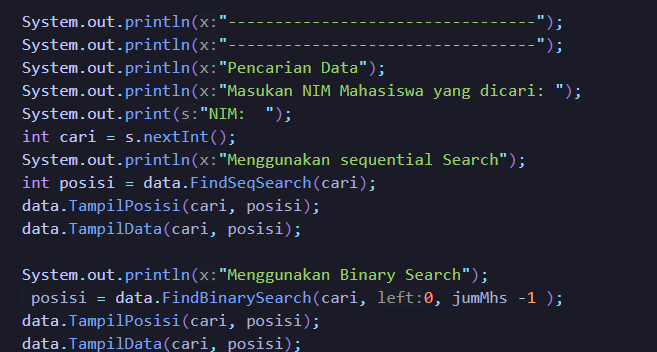
****

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

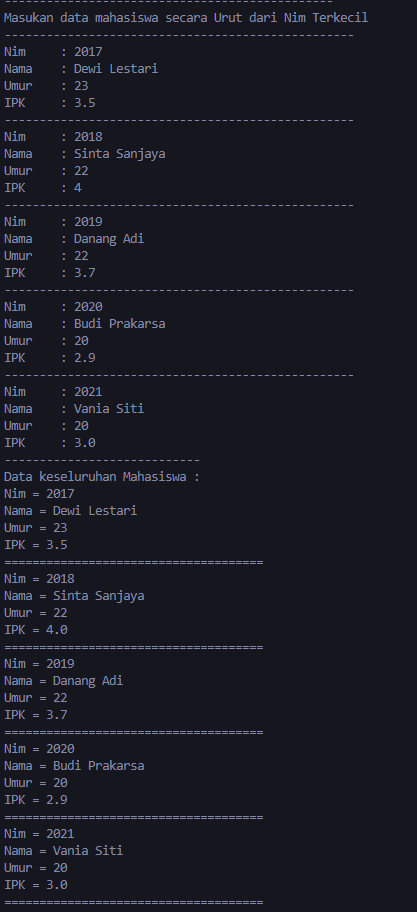
Pada percobaan 6.2.2 (sequential search) tambahkan method FindBinarySearch bertipe integer pada class PencarianMhs. Kemudian Deklarasikan isi method FindBinarySearch dengan algoritma pencarian data menggunakan teknik binary searching

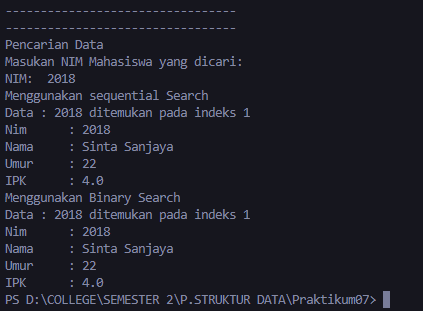


Panggil method FindBinarySearch terdapat pada class PencarianMhs di kelas Mahasiswamain. Kemudia panggil method tampilposisi dan tampilData



**6.3.2. Verifikasi Hasil Percobaan Cocokkan hasil kode program anda dengan gambar berikut ini**

****

****

**6.3.3. Pertanyaan**

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

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

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

4. 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

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

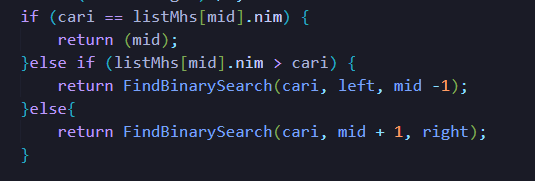
**Jawaban!**

1. Proses divide dalam algoritma pencarian binary search dijalankan pada baris berikut:

mid = (left + right) / 2;

Pada baris ini, nilai dari **left** dan **right** (yang mewakili batas-batas pencarian saat ini) dijumlahkan, dan hasilnya dibagi 2 untuk menentukan titik tengah (midpoint) dari rentang pencarian.

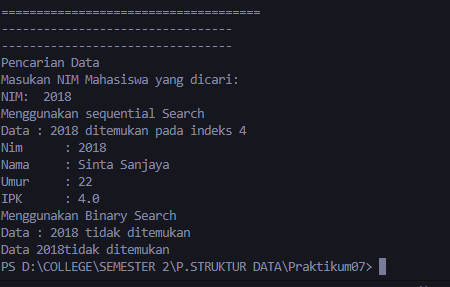
1. Proses Conquer dalam algoritma pencarian binary search dijalankan pada bagian berikut:



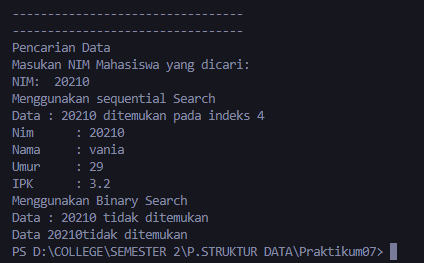
Pada bagian ini, setelah titik tengah (**mid**) ditentukan, program akan memeriksa apakah nilai yang dicari (**cari**) sama dengan nilai NIM pada indeks **mid**. Jika iya, proses conquer dilakukan dengan mengembalikan indeks **mid** tersebut sebagai hasil pencarian.

Jika tidak, maka akan dilakukan pemilihan sisi mana untuk melanjutkan pencarian. Jika nilai yang dicari lebih kecil dari nilai NIM di tengah (**listMhs[mid].nim**), maka proses conquer dilakukan pada sisi kiri dari titik tengah dengan melakukan pemanggilan rekursif **FindBinarySearch(cari, left, mid - 1)**. Begitu pula jika nilai yang dicari lebih besar dari nilai NIM di tengah, maka proses conquer dilakukan pada sisi kanan dengan pemanggilan rekursif **FindBinarySearch(cari, mid + 1, right)**.

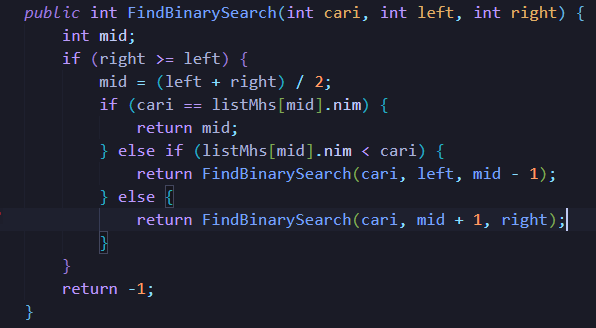
1. Jika data NIM yang dimasukkan tidak terurut, maka program masih dapat berjalan karena algoritma pencarian biner (binary search) dapat diimplementasikan untuk mencari nilai tertentu dalam kumpulan data yang tidak terurut. Namun, karena algoritma binary search mensyaratkan bahwa data harus dalam kondisi terurut untuk memberikan hasil yang benar, maka kemungkinan hasilnya data tidak ditemukan.



1. Jika daftar NIM dimasukkan dari yang terbesar ke yang terkecil, dan elemen yang dicari adalah 20210, hasil dari binary search tidak akan sesuai. Ini karena algoritma binary search mengasumsikan bahwa daftar data telah diurutkan dari yang terkecil ke yang terbesar atau sebaliknya, dan mencari nilai di tengah-tengah rentang pencarian. Dalam kasus ini, karena data telah diurutkan dari yang terbesar ke yang terkecil, algoritma tidak akan dapat bekerja secara efektif.



Untuk mengatasi masalah ini, kita dapat memodifikasi algoritma binary search agar sesuai dengan kondisi data yang telah diurutkan dari yang terbesar ke yang terkecil. Maka modifikasi program seperti berikut:



Hasil program nya sebagai berikut:



1. Berikut modifikasi dari program:





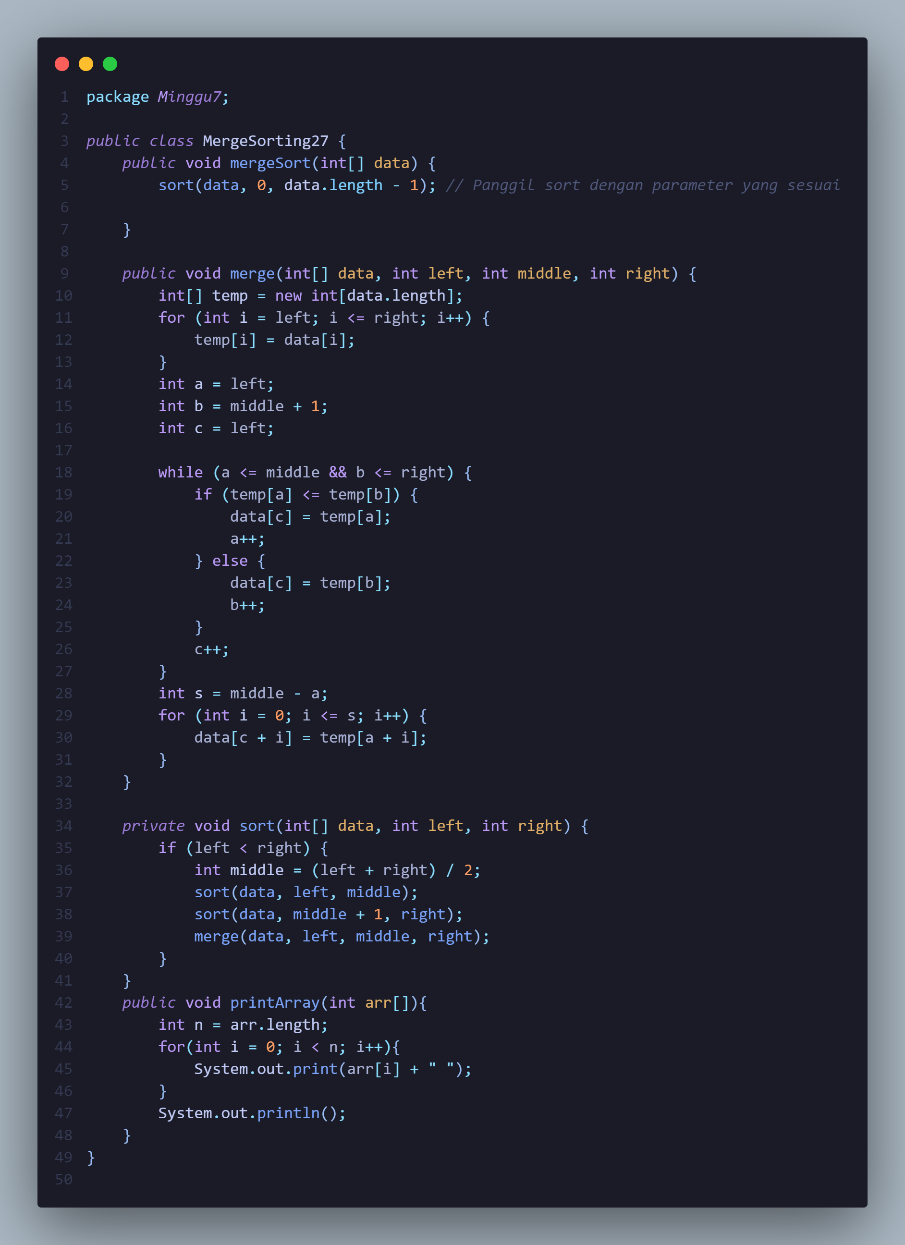
Output program:



**6.4. Percobaan Pengayaan Divide and Conquer**

**6.4.1. Langkah-langkah Percobaan Merge Sort**

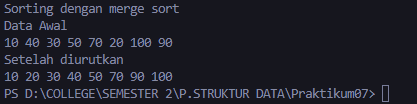
Buatlah class MergeSorting buatlah method mergeSort yang menerima parameter data array yang akan diurutkan. Kemudian buatlah method merge untuk melakukan proses penggabungan data dari bagian kiri dan kanan.Buat Method sort. Pada method mergeSort, panggil method sort dengan parameter data yang ingin diurutkan serta range data awal sampai dengan akhir. Tambahkan method printArray

****

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